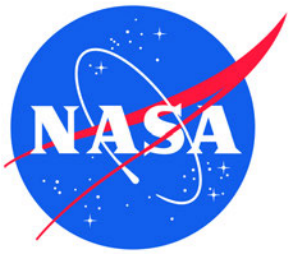
NASA/TP—2016–219422

JANNAF/GL—2016–0001
JDOC Abstract Number: 2016-0002



Simulation Credibility

Advances in Verification, Validation, and Uncertainty Quantification

Unmeel B. Mehta, Editor
NASA Ames Research Center, Moffett Field, California

Dean R. Eklund
United States Air Force Research Laboratory
Wright-Patterson Air Force Base, Ohio

Vicente J. Romero
Sandia National Laboratories, Albuquerque, New Mexico

Jeffrey A. Pearce
Universal Technology Corporation, Dayton, Ohio

Nicholas S. Keim
The Johns Hopkins University, Columbia, Maryland

November 2016

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

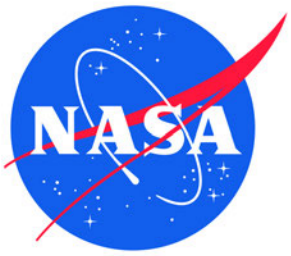
- Access the NASA STI program home page at http://www.sti.nasa.gov

- E-mail your question to help@sti.nasa.gov

- Phone the NASA STI Information Desk at 757-864-9658

- Write to:
  NASA STI Information Desk
  Mail Stop 148
  NASA Langley Research Center
  Hampton, VA 23681-219

NASA/TP—2016–219422 **JANNAF/GL—2016–0001**



Simulation Credibility

Advances in Verification, Validation, and Uncertainty Quantification

Unmeel B. Mehta, Editor
NASA Ames Research Center, Moffett Field, California

Dean R. Eklund
United States Air Force Research Laboratory
Wright-Patterson Air Force Base, Ohio

Vicente J. Romero
Sandia National Laboratories, Albuquerque, New Mexico

Jeffrey A. Pearce
Universal Technology Corporation, Dayton, Ohio

Nicholas S. Keim
The Johns Hopkins University, Columbia, Maryland

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035-1000

November 2016

Available
from

NASA STI Program
Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681-2199

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161

This report is also available in electronic form at http://www.sti.nasa.gov/ and http://ntrs.nasa.gov/

Table of Contents

Preface

Quantified simulation credibility—establishing the accuracy in terms of uncertainty—has been the principal challenge in the field of numerical simulation technology since its inception a century ago. Beginning in 1980, the significance of simulation credibility has continuously increased as numerical simulations have become ubiquitous in an increasing number of fields such as science and engineering, technology development, product acquisition, warfare, risk assessment, and "what if" studies. Simulation verification, simulation validation, uncertainty quantification, uncertainty propagation, and uncertainty aggregation are the key processes used to quantify numerical, model, referent & input, impact of input uncertainty, and output uncertainty, respectively. In this joint National Aeronautics and Space Administration (NASA) and Joint Army-Navy-NASA-Air Force (JANNAF) Interagency Propulsion Committee publication, we present a compilation of some key advances in these processes for establishing the credibility of simulation over the last 10 years, specifically in the discipline of continuum physics.

Development of formalized approaches for simulation credibility principally started in the 1990s. At that time, the United States Department of Defense (DoD) and Department of Energy (DoE) began to formulate systematic, uniform processes. In 1991 the DoD established the Defense Modeling and Simulation Office, which became the Modeling and Simulation Coordination Office in 2012, and that now serves as the Executive Secretariat for DoD M&S Management. In 1995, the DoE started the Advanced Strategic Computing Initiative, which evolved into the Advanced Simulation and Computing (ASC) Program in 2000. Organizations such as the American Institute of Aeronautics and Astronautics (AIAA) and American Society of Mechanical Engineers (ASME) elucidated the conduct of verification and validation: (1) *Guide for the Verification and Validation of Computational Fluid Dynamics Simulations*, AIAA, 1998; (2) *Guide for the Verification and Validation of Computational Solid Mechanics*, ASME, 2006; and (3) *Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer*, ASME, 2009.

Among the many examples driving motivation for such activities, two stand out: (1) The National Aero-Space Plane (NASP), the X-30, was to be largely designed with computational fluid dynamics (CFD) because aero-propulsion issues could not be duplicated in ground-test facilities. (2) The performance, reliability, and safety of the U.S. nuclear weapons stockpile were to be and are being mainly ensured through simulation after testing of nuclear devices was banned in 1996.

From 1991 to 1999, the Airbreathing Propulsion Subcommittee of the JANNAF Interagency Propulsion Committee held many workshops related to simulation credibility processes, for example, workshops on Computational Fluid Dynamics Code Validation/Calibration. In 1999, Unmeel Mehta was asked to advocate to the JANNAF Executive Committee the creation of the Modeling and Simulation Subcommittee (MSS), which was established under his leadership with a charter that included simulation credibility, virtual engineering, and integrated system health management. One of the panels he established was the Uncertainty Assessment and Management (UA&M) Panel, which was renamed the Simulation Credibility Panel. In 2005, Dean Eklund joined MSS and the UA&M Panel.

In 2007, we began a voluntary endeavor to develop a simulation credibility guide to describe and demonstrate state-of-the-art approaches for simulation verification, simulation validation, and uncertainty quantification. Mehta took the lead to accomplish this endeavor. Eklund and a team of three colleagues—Vicente Romero, Jeffrey Pearce, and Nicholas Keim—assisted and advised. We highly appreciate the keen interest and dedication of these colleagues.

Numerous Simulation Credibility workshops were held to identify specialists in different aspects of simulation credibility, discern and discuss advances in the aforementioned processes, and highlight key modeling and simulation issues related to air-breathing hypersonic propulsion and liquid and solid rocket propulsion. This publication is the result of the Panel's efforts.

We were restricted in the number of simulation credibility advances that could be presented in one publication. The absence of different demonstrated points of view on the conduct of model validation in this volume remains our responsibility. Nevertheless, we envision that creators of continuum physics simulations will find the detailed processes and illustrated examples in this volume very useful to guide and assist them in conveying the credibility of their simulations to users of simulations, specifically decision makers.

Since the 1980s, many science, technology, engineering, policy, and modeling and simulation experts have driven improvements in simulation credibility. The authors involved in this publication have been dedicated to advancing simulation credibility. It is a great pleasure to acknowledge their willingness to participate in this endeavor beyond their normal professional tasks and to recognize both their commitment to this endeavor and their work in preparing and finalizing their chapters. We are indebted to them for their outstanding service to the simulation community at large.

Special thanks to: Vicente Romero for bringing to this endeavor his expertise in the conduct of uncertainty analysis and his experience in the use of simulations in support of the U. S. nuclear stockpile and for recruiting other authors from Sandia National Laboratories; Jeffrey Pearce for meticulously recording Simulation Credibility Panel minutes and for sharing his experience in a similar effort that developed *Scramjet Propulsion Testing Standards, Recommended Practices, and Guidelines* (CPIAC Publication 710); and Nicholas Keim for greatly facilitating the conduct of this endeavor within the framework of the JANNAF Interagency Propulsion Committee.

We diligently reviewed drafts of all chapters and contributing authors were provided constructive comments for improvements. External technical reviewers largely assisted us in this review process. Meticulous efforts of these external reviewers are very much appreciated. Jill Dunbar, Michelle Moyer, and Katharine Pitta expertly edited draft versions of chapters and offered many suggestions to significantly enhance their readability. Katharine Pitta also primarily assembled this publication. We thank them.

Unmeel Mehta, Editor
Dean Eklund

Simulation Credibility—Uncertainty, Verification, Validation, and Risk—Panel
Modeling and Simulation Subcommittee
Joint Army-Navy-NASA-Air Force Interagency Propulsion Committee

"The reasonable man adapts himself to the world;
the unreasonable one persists in trying to adapt the world to himself.
Therefore all progress depends on the unreasonable man."

— George Bernard Shaw

Introduction[1]

Unmeel B. Mehta[*] and Dean R. Eklund[†]

[*]*NASA Ames Research Center, Moffett Field, CA 94035*
[†]*United States Air Force Research Laboratory, Wright-Patterson Air Force Base, OH 45433*

1. Prolegomena

1.1 Overview

Decision makers and other users of simulations need to know quantified simulation credibility to make simulation-based critical decisions and effectively use simulations, respectively. The credibility of a simulation is quantified by its accuracy in terms of uncertainty, and the responsibility of establishing credibility lies with the creator of the simulation. In this volume, we present some state-of-the-art philosophies, principles, and frameworks. The contributing authors involved in this publication have been dedicated to advancing simulation credibility. They detail and provide examples of key advances over the last 10 years in the processes used to quantify simulation credibility—verification, validation, and uncertainty quantification. The philosophies and assessment methods presented here are anticipated to be useful to all technical communities conducting continuum physics-based simulations; for example, issues related to the establishment of simulation credibility in the discipline of propulsion are discussed. We envision that simulation creators will find this volume very useful to guide and assist them in quantitatively conveying the credibility of their simulations.

1.2 Background

In 1910, Richardson presented the approximate arithmetic solution by finite differences, computing stresses in a masonry dam [1]. Subsequently, he published *Weather Prediction by Numerical Process* [2]. In 1928, Thom solved the Navier-Stokes equations by replacing them with finite difference equations to simulate the flow past a circular cylinder at a Reynolds number of 10 [3]. In 1953, Kawaguti obtained a numerical solution of the Navier-Stokes equations for the flow around a circular cylinder at Reynolds number 40 using a mechanical desk calculator, working 20 hours a week for 18 months [4]. Over the last 60 years, tremendous advances in many key facets of digital computing, from hardware to algorithms, have taken place.

These advances are already making a significant impact on aspects of design, engineering, manufacturing, and safe operations of aerospace components and subsystems. In other endeavors,

[1] This introduction presents updated relevant aspects of the paper titled, "The JANNAF Simulation Credibility Guide On Verification, Uncertainty Propagation and Quantification, and Validation" (Invited), U. Mehta, V. Romero, D. Eklund, J. Pearce, and N. Keim, *53rd AIAA Aerospace Sciences Meeting*, AIAA Paper 2015-0502, Kissimmee, Florida, 5−9 January 2015.

such as nuclear power generation, nuclear weapon safeguards, civil engineering, healthcare, and science, such computational advances are also making a tremendous impact. These advances have the potential to revolutionize the way critical decisions are made and to contribute significantly to the advancement of science, technology, and engineering—if the uncertainty of a simulation is quantified and its value is acceptable for making a simulation-based decision. Simulation credibility has been the principal issue from the inception of numerical simulation technology [1].

During the 1980s the focus was on code validation; see for example Reference 5. Since circa 1995, the emphasis has shifted to model validation [6, 7]. But, we believe the focus ought to be on simulation validation [8]. How accurate are simulations? What are the quantified uncertainties in the simulations [5, 9]? What level of confidence can one assign to simulations? Is the available evidence enough to make correct decisions? Decision-makers and other users of simulations need answers to such vital questions; for example, see References 5, 8, and 10. Simulation credibility is essential for risk-based decision-making and risk-informed decision analysis. If simulations are provided to users, then a statement regarding simulation credibility must accompany them. The burden of proof for simulation credibility lies with the presenter of the simulations.

2. Simulation Credibility

"Accuracy builds credibility."

— Emanuel James "Jim" Rohn

Simulation credibility is assessed both qualitatively and quantitatively. The former assesses the quality of simulation with circumstantial evidence; the latter quantifies the quality of the simulation. The overall credibility is strongly impacted by the quantitative assessment of credibility. In science, technology, and engineering, what ultimately matters is the quantified credibility. That is the focus of this volume, and the phrase "quantified simulation credibility" is abbreviated to "simulation credibility."

The estimated simulation error includes numerical error, model error, and input parameter error. This estimate of error is essential for establishing simulation credibility. The acceptability of this estimate for the intended use leads to the acceptability of the simulation model, which could include a physics model, calibrated physics model, or a surrogate (metamodel). As defined in Reference 7, model calibration is the act of tuning a simulation model, somewhat without scientific justification. A metamodel is a model of a model. Irrespective of the kind of model used, to determine simulation credibility, a simulation is compared with a referent.

Simulation validation is the process of quantifying uncertainty of the simulation. This quantification is achieved with simulation verification, propagation of input uncertainty, referent uncertainty, quantification of input and physics model uncertainties, and aggregation of all uncertainties. A favorable outcome of simulation validation establishes the validity of the simulation model, may lead to its use for predictions, and validates the corresponding mathematical model. However, a valid mathematical model does not necessarily assure an acceptable simulation for the same validation conditions, if the conversion from the mathematical model leads to a different simulation model that is not comparable in effect to the simulation model used for validation.

A mathematical model (original model) and its corresponding simulated mathematical model (modified model) are not the same; see Reference 11. When the mathematical model is difficult

or impossible to solve analytically, a modified model is solved non-analytically, for example, using computational fluid dynamics (CFD) techniques. If the mathematical model is f(u) = 0, then the modified model is f(u) + g(u,h) = 0, where h represents some measure of the discretized domain (such as characteristic cell size). While a great deal of effort is spent on reducing and localizing the effect of g(u,h), this effect cannot be eliminated. Furthermore, if physics is not fully resolved, the effect of unresolved physics is often captured; for example, when effects of shock layers or small-scale turbulence are simulated. Thus, a simulation model is neither identical to the mathematical model nor a perfect analog of reality when physics is not fully resolved.

Although the Navier-Stokes (NS) equations model is valid for a class of real-world physics, a direct NS (DNS) simulation requires a referent to establish the credibility of the simulation. Strictly speaking, these simulations are performed with modified NS equations. Moreover, the road to laminar-to-turbulent transition is dependent on boundary conditions that need to be known precisely. A confirmation through simulation verification requires that numerical errors, both spatial and temporal, are negligible and through the referent that all quantities of interest are simulated correctly and all relevant flow scales are fully resolved. Physics with multiple time and space scales are difficult to simulate given the limitation of computing technologies and tools.

As shown in Figure 1, simulation model validation is conducted at critical and sensitive locations in the domain identified with the design of experiments technique (green area). This technique assists in the conduct of simulation validation in the domain of interest to establish the domain of validity of the simulation model [12–18]. If the outcome of simulation validation is satisfactory, predictions are done at locations other than these validation locations when appropriate. Simulation credibility of predicted simulations inside the simulation model validation domain and in the neighboring domain (blue area) could be higher than outside these domains.

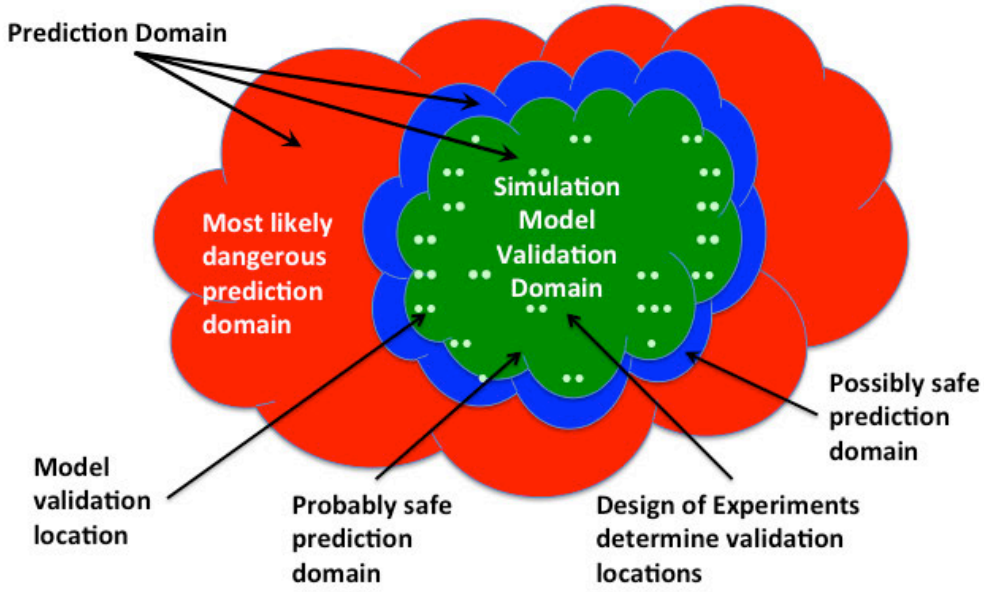


Figure 1. Simulation validation domain.

Figure 2 presents a high-level diagram of the important role of uncertainty in various aspects of simulation validation. The top row and the center of the diagram show a linkage between the requirements for simulation, resource allocation, and simulation validation for both simulation model validation and prediction credibility. These aspects play major roles in the use of predictions for design, analysis, and operations of a system and its components.

The International Organization for Standardization (ISO) has codified the concepts and definitions of error and uncertainty for the experimental community in the Guide to the Expression of Uncertainty in Measurement [19]. Coleman and Stern adapted most of these concepts and definitions for simulations [20], establishing the use of a common language between the experimental community and the simulation community. Subsequently, the ASME Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer (ASME V&V 20-2009) [21] did the same. This standard quantifies model uncertainty starting with the comparison error between a simulation and a referent.[2]

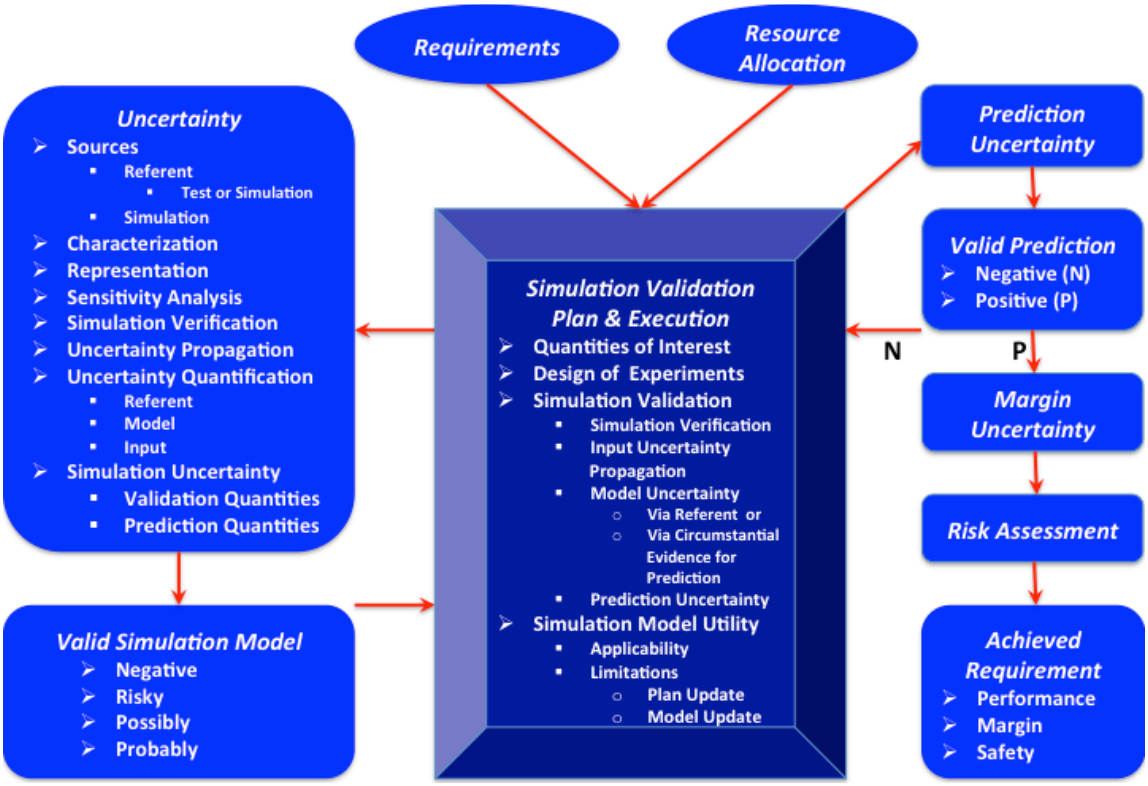


Figure 2. The role of uncertainty in simulation validation for a simulation-based design.[3]

During simulation validation, a comparison of simulation is done with a referent. The comparison error equation adapted[4] from ASME V&V 20 is as follows [21]:

$$E = S - R = (T + \delta_{Sval}) - (T + \delta_R) = \delta_{Sval} - \delta_R \quad (1)$$

Here, E, S, R, and T are the comparison error, simulation, referent, and true value, respectively. The latter value is assumed to be the same for the simulation and the referent. The quantities δ_{Sval} and δ_R are simulation error and referent error during validation, respectively. The simulation error, δ_S, is:

[2] In Chapter 10, Romero presents a more general comparison scheme to handle added complications of traveling parametric and discrete uncertainties that are related to the model and of replicate experiments with randomly varying stochastic systems and models for predicting the stochastic phenomena.

[3] In Fig.1 of Chapter 5, Romero presents some important aspects of uncertainty and sensitivity analysis.

[4] The nomenclature used here is somewhat different from that in ASME V&V 20.

$$\delta_S = \delta_{mod} + \delta_{num} + \delta_{input} \tag{2}$$

where quantities δ_{mod}, δ_{num}, and δ_{input} are, simulation model error, numerical error, and input parameter error, respectively; and $\delta_{num} = \delta_{nunm} + \delta_{numd}$ where δ_{nunm} and δ_{numd}, are numerical method error and discretization error, respectively. The model error is:

$$(\delta_{mod})_{val} = [E - (\delta_{num} + \delta_{input} - \delta_R)]_{val} \tag{3}$$

A measure of uncertainty is defined as an estimate of an interval $\pm u$ that should contain error, δ. The model error falls in the interval given by:

$$(\delta_{mod})_{val} \in [E - u_{mod}, E + u_{mod}]_{val} \tag{4a}$$

where E_{val} is an estimate of $(\delta_{mod})_{val}$ and the model validation uncertainty, $(u_{mod})_{val}$, is:

$$(u_{mod})_{val} = \sqrt{\left(u_{num}^2 + u_{input}^2 + u_R^2\right)_{val}} \tag{4b}$$

Equation 4 is used only when the underlying statistical assumptions regarding independent uncertainties are valid. When they are *not* independent, the approach for computing $(u_{mod})_{val}$ is presented in Reference 21.

The model uncertainty, u_{mod}, in Equation 4 is a systematic standard uncertainty and is defined as an estimate of the standard deviation of the set of possibilities within which δ_{mod}—the combination of errors $(\delta_{num} + \delta_{input} - \delta_R)$—lies. No assumption about the form of the parent distribution is associated with the definition of u_{mod}. It is determined from a combination of numeric, input, and referent uncertainties. Simulation verification provides u_{num} and uncertainty propagation of input uncertainties provides u_{input}. Thus, simulation verification and input uncertainty quantification and propagation are necessary first steps for establishing simulation validity. The referent uncertainty, u_R, must also be estimated. Note that $(\delta_{mod})_{val}$ is generally not the same for different uses of the simulation model.

Simulation validation has other issues, namely: multi-dimensional problems are often difficult to solve; simulations could be non-deterministic; the possibility of achieving either an erred simulation or a non-converging simulation during mesh refinement for simulation verification (for example, see Reference 22); available computing capability may not be sufficient; the mathematical model may include multiple physics models; or numerical methods may also impact simulated physics and modify the effect of or interact with physics model(s) used in simulation. Examples of the last two issues are: multiple physics models in a single simulation problem (see Figure 3); and methods of uncertainty propagation may impact the physics simulated with the simulation model. Moreover, it is often hard to disaggregate modeling errors from different physics models and from nonphysical (numerical) models that impact physics.

Scenarios that further complicate the validation effort include: not knowing all necessary input conditions, having only partial relevant real-world data, or conducting validations at just a few locations in the simulation model validation domain. Different test facilities and measurement techniques may provide somewhat different data. The assumed true value of measurement determines the estimate for measurement error, as shown in Figure 4. The operational environment of a system may differ from the environment in a test facility. Finally, limited resources could affect whether or not the simulation validation is conducted properly.

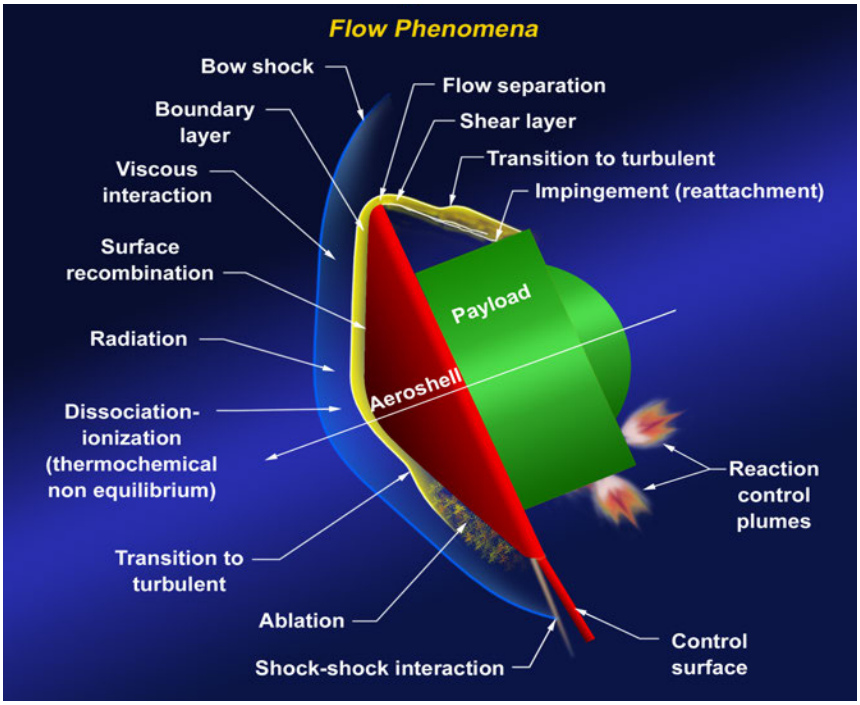


Figure 3. Kaleidoscope of physics surrounding a craft during atmospheric entry (Credit: NASA).

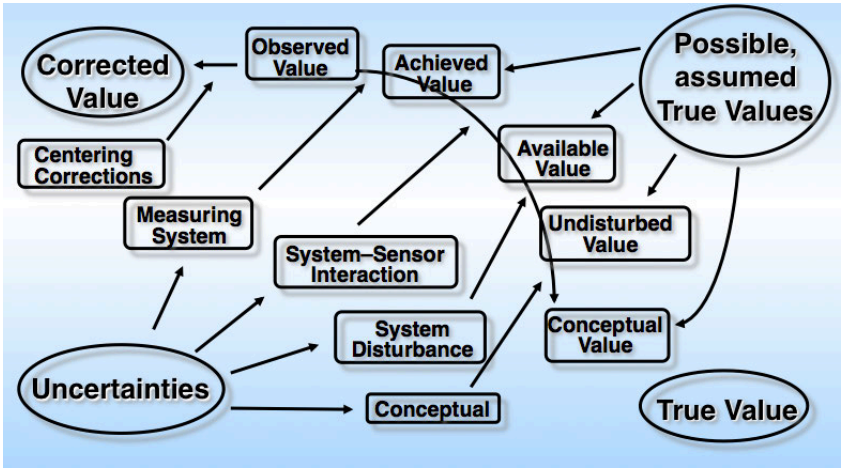


Figure 4. Observed, true values, and uncertainties in tests (adapted from Moffat [23]).

Simulation users are interested in the estimated simulation error during simulation validation. This topic is not discussed in ASME V&V 20 [21]. We determine this error and suggest a way to estimate the prediction error both within and in the immediate vicinity of the validation domain.

When simulation validation is conducted, the simulation error (δ_{Sval}) is obtained from Equations 2 and 3:

$$\delta_{Sval} = \delta_{mod} + \delta_{num} + \delta_{input} = [E - (\delta_{num} + \delta_{input} - \delta_R)]_{val} + [\delta_{num} + \delta_{input}]_{val} \quad (5)$$

A measure of uncertainty is defined as an estimate of an interval $\pm u$ that should contain error δ. The validation simulation error, δ_{Sval}, falls in the following interval:

$$\delta_{Sval} \in [E_{val} - u_{val}, E_{val} + u_{val}] \tag{6a}$$

where:

$$u_{val} = \sqrt{\left(u_{mod}^2 + u_{num}^2 + u_{input}^2\right)_{val}} \tag{6b}$$

with Equation 4b defining u_{mod}.

If the simulation error (Eq. 6a) is acceptable, then the simulation is valid. With the outcomes of simulation verification and uncertainty propagation and with the referent uncertainty acceptable, the simulation model error (Eq. 4) is acceptable. Thus, the simulation model is valid. If the simulation model is valid, then the modified mathematical model is valid for the numerical method used and the input parameters considered. Consequently, the original mathematical model is valid with the same constraints. To remove this qualification, a justification is required.

Model developers are principally interested in reducing the error in the simulation model, δ_{mod}. The user is principally interested in simulations, and needs to know simulation prediction error, δ_{Spred}. Prediction credibility is established with direct and circumstantial evidence [5, 8].

Conceptually, prediction error is within the interval given in Equation 7. This equation is used only when the underlying statistical assumptions regarding independent uncertainties are valid.

$$\delta_{Spred} \in [E_{pred} - u_{pred}, E_{pred} + u_{pred}] \tag{7a}$$

with:

$$u_{pred} = \sqrt{\left(u_{mod}^2 + u_{input}^2 + u_{num}^2\right)_{pred}} \tag{7b}$$

In Equation 7b, u_{pred} is the root sum square of the standard model, numeric, and input uncertainties.

The prediction uncertainty may be determined in four ways with varying degrees of direct and circumstantial evidence. If the uncertainty is acceptable for the intended use of the prediction, then the prediction is acceptable. However, the simulation code and its use should be the same and consistent, respectively, for predictions and simulation validation. The following are the alternatives for quantities in Equation 7:

1. E_{pred} and u_{pred} (at a prediction location) are assumed to be the same as E_{val} and u_{val} (at a nearby validation location), respectively.
2. u_{num} and u_{input} are determined and E_{pred} and u_{mod} are:
    a. Assumed to be the same as those at the nearby validation location.
    b. Interpolated to the prediction location within the model validation domain.
    c. Extrapolated to the prediction location outside the model validation domain.
3. E_{pred}, u_{mod}, u_{num}, and u_{input} are:
    a. Interpolated to the prediction location within the model validation domain.
    b. Extrapolated to the prediction location outside the model validation domain.

Whichever alternative is selected, it must be justified. Interpolation or extrapolation is done from neighboring validation locations. In the second approach, numerical and input errors, δ_{num} and δ_{input}, are computed. Generally, approach 3.a is much less risky than approach 3.b. The extrapolation error can be dominant, and this error is very difficult to quantify.

3. Some Simulation Credibility Frameworks

"The mere formulation of a problem is often far more essential than its solution . . ."
— Albert Einstein

The significance of processes for establishing simulation credibility is manifested in the creation of guides, standards, archival publication journals, and professional society conferences, leading to a number of definitions and frameworks. Principally, how the word "validation" is defined and how the process it represents is conducted differ within ASME and across U.S. government organizations such as the Department of Defense (DoD), Department of Energy (DoE), and NASA. Additionally, there are personal differing points of view among the simulation community. Briefly, some of these differences are highlighted, and the M&S Results Credibility Assessment presented in the NASA Standard of Models and Simulations (NASA-STD-7009A) [24] is discussed below.

The definition of verification used in NASA-STD-7009A deals with "conceptual models, mathematical models, or other constructs"; mandatory requirements for verification are to verify all models and document the domain of verification of all models (see Table 1). Simulation verification providing numerical uncertainty is not a mandatory requirement—the document does not mandate simulation verification, whereas ASME V&V 20-2009 [21], Guide for the Verification and Validation of Computational Fluid Dynamics Simulations (AIAA-G-077-1998) [25], and Guide for Verification and Validation in Computational Solid Mechanics (ASME V&V 10-2006) [26] *do* require simulation verification.

The AIAA and ASME documents include both code and simulation verification in their conduct of "verification." But code verification and simulation verification are addressed separately. Both the NASA standard and Department of Defense Instruction (DoDI) [27] assign the same role to "verification" (Table 1), namely to verify whether the code is built correctly. Mehta [8] focuses only on simulation verification, as code verification is a vital but separate task.

The AIAA and ASME documents define "verification" without the following qualification: "for the intended use"; see References 25, and 26. However, this phrase is necessary because the required level of numerical accuracy depends on the intended use. For example, the level of numerical accuracy and associated uncertainty required for the drag coefficient of a commercial transport aircraft is quite stringent compared to that required for its lift coefficient for the same modeling accuracy. Mehta uses this phrase in the definition of verification [7].

The concepts of verification and validation used in ASME V&V 20 [21] are consistent with the definitions used in AIAA-G-077-1998 and ASME V&V 10 [25, 26]. The definitions used for verification and validation in ASME V&V 10 are consistent with those used in AIAA-G-077-1998. In the latter publication the definition of validation was taken verbatim from Reference 6 and the 1996 version of DoDI. None of these publications use the word "simulation" in their definitions of validation. Mehta [8] *does* include—*introduces*—simulation in his definition of validation: a model is valid only if the simulation resulting from the model is valid. In other

words, the first step necessary to validate a model is simulation validation, as explained previously. Subsequently, the definition of validation used in NASA-STD-7009A / NASA-STD-7009 [24], and DoDI [27] is consistent with Mehta's [8] with the phrase "real world" equivalent to the word "reality." In the DoDI, validation assesses both numerical and modeling accuracy.

Table 1. Existing Definitions of Verification and Validation.

Source	Verification	Validation
AIAA-G-077-1998 [25] (Adapted by ASME V&V 10 [26])	The process of determining that a model implementation accurately represents the developer's conceptual description of the model and the solution to the model.	The process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model.
Mehta [8]	Verification assesses the credibility of a simulation model by estimating the degree to which this model is an accurate representation of the conceptual model from the perspective of the intended uses of simulations.	Validation assesses the credibility of a reality model or a simulation by estimating the degree to which it is an accurate representation of reality from the perspective of intended uses of the model or simulation.
NASA-STD-7009A [24]	The process of determining the extent to which an M&S is compliant with its requirements and specifications as detailed in its conceptual models, mathematical models, or other constructs.	The process of determining the degree to which a model or a simulation is an accurate representation of the real world from the perspective of the intended uses of the M&S.
DoDI 5000.61 [27]	The process of determining that a model or simulation implementation and its associated data accurately represent the developer's conceptual description and specifications.	The process of determining the degree to which a model or simulation and its associated data are an accurate representation of the real world from the perspective of the intended uses of the model.

The execution of model validation in ASME V&V 10 differs from that in AIAA-G-077-1998 and ASME V&V 20. The former addresses fully the issue of *the intended uses of the model*, while the latter two do not by conducting validation only at locations where referent data are available. Without an acknowledgement of this shortcoming, AIAA-G-077-1998 was approved and subsequently reaffirmed. An explicit statement is made in ASME V&V 20 limiting its scope.

In Chapter 10, Romero defines Real Space (RS) model validation as follows: "Model Validation is the compilation of useful indicators regarding the accuracy and adequacy of a model's predictive capability for output quantities of interest, where meaningful comparisons of experiment and simulation results are conducted at points in the modeling space that present

significant prediction tests for anticipated uses of the model." He considers this definition to be an "operational definition" of model validation (see Reference 28). This definition includes the word "adequate," whereas the definitions of validation in Table 1 do not include this word.

Both of the words "accuracy" and "adequacy" characterizing model are in context of reality (see Chapter 10). *Accuracy* is used to decide "on the formulation or metric for characterizing the discrepancy between model and experiment results." *Adequacy* decides "on the threshold or criterion for model adequacy (acceptable agreement with reality)." An assertion is also made, "[T]he common paradigm … of proposing pre-specified accuracy requirements against which a model is to be validated for adequacy is … unworkable in practice." This assertion raises an obvious question that needs to be addressed: What should be the process to validate a model to make this requirement workable in practice?

Schlesinger et al., define model validation as the substantiation that, within its domain of applicability, a model possesses a satisfactory range of accuracy consistent with its intended application [29]. It is worthwhile comparing Schlesinger's definition with other discussed definitions. Here, the terms "intended use" and "use purpose" incorporate "domain" and "application"; and "satisfactory" means "adequate." The differences are: "accuracy" versus "the degree to which a model is an accurate representation of the real world"; and "a satisfactory range of accuracy" versus "the compilation of useful indicators regarding the accuracy." Among these phrases, the common word is "accuracy." The determination of uncertainty quantifies accuracy. *Accuracy*—a quantitative entity in science, technology, and engineering—for the intended use decides *adequacy*—a qualitative entity. In other words, these two entities are not independent and the phrase "for the intended use" addresses *adequacy*.

In AIAA-G-077-1998 [25] and ASME V&V 20 [21], the validation guidance states that there can be no validation without experimental (real-world) data with which to compare the simulation or the derived simulated result. However, microphysics, valid simulations with estimated error in macrophysics quantities of interest derived from these simulations, might be used as referents to validate macrophysics simulations. For example, subgrid-scale models for large eddy simulations (LES) can be validated with DNS that fully resolve physics of turbulence. Likewise, simulations generated with a physics model serve as a referent to validate simulations based on a metamodel.

ASME V&V 10-2006 defines the word "referent" as "data, theory, or information against which simulation results will be compared" and also defines the phrase "validation requirements" as "specifications and expectations that a computational model must meet to be acceptable for its intended use" [26]. The latter definition is elaborated with the statement: "These requirements are established in the V&V plan and often include descriptions of specific tests that need to be run or other referents that need to be gathered, as well as accuracy statements about the agreement between simulation outcomes and experimental outcomes or referents." Thus, ASME V&V 10-2006 allows for referents that are other than experimental outcomes. Note that the NASA standard adapts this definition of "referent." The DoD defines this word as "a codified body of knowledge about a thing being simulated" [30].

The ability to perform validation with test data is limited by the availability of these data: tests may be limited, difficult, costly, or unfeasible. The latter is illustrated by the following example: the safety of the U.S. nuclear stockpile has been ensured through simulation since testing of nuclear devices was banned in 1996. Likewise, the DoD cannot conduct an experimental war to collect real data. The DoD defines "real world" as "the set of real or hypothetical causes and effects that simulation technology attempts to replicate" [27]. In other words, "real world" is defined as either test data, another "valid" simulation, or subject-matter expert judgment.

NASA-STD-7009A presents an approach to M&S results credibility assessment and states: "The assessment of credibility discussed in Appendix E does not purport to determine credibility directly, but merely to assess key factors that contribute to a decision-maker's credibility assessment. The assessment of these factors is focused on the results produced by the M&S; however, the completeness and rigor of all the activities throughout the life cycle of the M&S contribute to it. This is why there are both developmental and operational factors for credibility. The decision makers can then make their own credibility assessment based upon this information in the context of the decision at hand" [24].

The NASA credibility assessment approach considers eight factors that are assumed "nearly orthogonal, i.e., largely independent" [24] and are evaluated in three categories: Development (data pedigree, verification, validation); M&S Use (Operations—input pedigree, uncertainty characterization,[5] results robustness); and Supporting Evidence (M&S history, M&S process/ product management). Each factor is evaluated with a five-level assessment scale. Note that M&S Use comes after verification (as defined in the NASA standard) and preferably after validation, making the assumption of independence among these factors questionable.

Under Operations, simulation verification is not required. Just as a publication in the journals of AIAA and ASME requires a statement on numerical accuracy, simulation users need to know numerical uncertainty. That is the responsibility of the person creating simulations.

Operations are questionable if verification (as defined in the NASA standard) is not conducted or is assessed at a low level with respect to model design. Also, the final outcome of validation is questionable without first having the final outcome of verification. The standard responds to this concern by stating, "While verification is usually meant to occur prior to validation, that precedence is not necessarily required and the two often occur in a cyclical or mixed manner depending on many factors in the overall model development process, and the success or failure in the testing process."

NASA-STD-7009A requires that reported uncertainty estimates include one of the following: (1) a quantitative estimate of the uncertainty in the M&S results; (2) a qualitative description of the uncertainty in the M&S results; or (3) a clear statement that no quantitative estimate or qualitative description of uncertainty is available" [24].

The NASA policy is to "prefer use of performance (outcome-based) standards in procurement activities over design or process (method-based) standards" [31]. But the focus of NASA-STD-7009A is on documentation, output, and processes rather than on outcome. The key shortcoming of the NASA standard is the credibility assessment approach, both in terms of its logic and the acceptability of non-uncertainty-based assessment for making decisions. This approach provides a holistic, qualitative assessment of M&S.

Likewise, the Predictive Capability Maturity Model (PCMM) is "an expert elicitation tool designed to characterize and communicate completeness of the approaches used for computational model definition, verification, validation, and uncertainty quantification associated for an intended application" [32]. PCMM and the Phenomena Identification Ranking Table (PIRT), first developed for the nuclear power plant industry [33], form a holistic framework to assess M&S quality and manage risk in the use of M&S, as discussed in Reference 34. Note that

[5] This factor deals with the identification of all known sources of uncertainty and propagation of input uncertainties.

this reference accepts the definition of validation as listed in ASME V&V 10 and follows the guidance of ASME V&V 20: "There can be no validation without experimental data with which to compare the results of the simulation." PCMM does not address two issues: regulatory requirements and the impact of numerical methods on simulated physics [32].

Difficulties in properly conducting verification and validation, the inability to properly quantify uncertainties, the lack of real-world data, the limitations of computing technologies, and the limitations of resources have led to the consideration of circumstantial evidence and, in part, to the development of non-deterministic perspectives and methods (first introduced by Floyd [35]). Ultimately, credible predictions are needed. Direct and circumstantial evidence plays a major role in establishing this credibility.

Non-deterministic approaches are used to manage simulation uncertainty and to provide decision-support information for addressing high-consequence failure events associated with the development and operation of systems. Both probabilistic and non-probabilistic methods are used to propagate, quantify, and manage uncertainty. Often, values of uncertainties are assumed because they are difficult to quantify. The credibility of a non-deterministic simulation for making a critical decision is a primary concern.

4. Verification

"The man of science has learned to believe in justification, not by faith, but by verification."

— Thomas Henry Huxley

There are two aspects of verification: code verification and simulation verification. Simulation verification and validation can be conducted only after the outcome of code verification is satisfactory. The outcome of simulation verification, in terms of numerical uncertainty, is required for both simulation validation and for assessing prediction credibility.

Knupp presents in Chapter 1 the complex processes involved in verifying software correctness for computer codes that solve partial differential equations (PDEs) in order to simulate physical processes. Three approaches for conducting simulation verification are presented. For simulation verification, the focus is on estimating discretization errors—in particular, spatial discretization errors. In Chapter 2, Rider et al., survey the state of the art of simulation verification, and illustrate the application of Richardson Extrapolation (RE) to quantify uncertainty in integral output quantities. They employ techniques of both error estimation and uncertainty quantification. In Chapter 3, Cavallo advances Zhang's Error Transport Equations (ETE) method [36]. This method is used not only for quantifying discretization error but also for adaptive mesh refinement. He summarizes the applicability and limitations of the Richardson extrapolation method. In Chapter 4, Nemec and Aftosmis demonstrate the power of adaptive mesh refinement with adjoint-based error estimates. They present results showing reliable estimates and automatic control of discretization error.

Code verification is the first step in establishing simulation credibility. It is performed to ensure that a code is capable of producing "correct" simulation. Whether simulations from a code are correct is not known, except when exact solutions are known. Two principal sources of error are algorithm deficiencies and programming mistakes. Knupp provides context for code verification and describes the main concepts, practices, and processes along with various issues that can arise. He discusses two types of specialized testing, Numerical Algorithm Property Verification and Numerical Algorithm Adequacy Verification, as well as verification test suites.

Rider et al., present detailed descriptions of many aspects of simulation verification and propose a workflow for guiding simulation verification for the executable software to be examined after an algorithm implementation has passed the appropriate level of software quality assurance and code verification. The code verification provides an analysis of the numerical method as implemented, including its accuracy and stability properties.

With an appropriate sequence of mesh discretization for each problem of interest, the code user executes and produces appropriate metrics to evaluate the difference between the computed solutions, based on numerical parameters of the numerical method chosen (order of approximation or scheme) and within the user's control. These differences determine the sequence of estimated errors corresponding to the various discretizations and tolerances. The mesh sequence can also be used to determine the rate of convergence for the method, which is then compared to the theoretical rate. For iterative solver errors, the error as a function of the stopping criteria and the discretization must also be determined. Using these results, an assessment of the accuracy (level of error estimated) is rendered for the simulation with a given set of numerical settings. Rider et al. also examine degree of coverage of features in an implementation by the verification testing.

They also identify four general types of simulation verification errors: (1) round-off errors, (2) sampling errors, (3) iterative solver errors, and (4) discretization errors. Round-off errors are an unavoidable consequence of representing numbers on computers with finite precision, and usually make only a small contribution to numerical error. Sampling errors are present for stochastic methods (e.g., Monte Carlo methods). Iterative solver errors are a consequence of solving any linear or non-linear system of equations during the course of the computational procedure. Discretization errors are associated with the numerical scheme used to obtain a simulation of the discrete equations approximating the mathematical equations. Techniques by which numerical error estimates can be converted to numerical uncertainty estimates are described.

Cavallo presents another approach for solution verification that is well suited to fluids problems and unstructured meshes. To estimate error, in his approach, error equations are derived and solved simultaneously with the original PDE. The ETE is based on the upwind construction of the computational scheme and the substitution of the discrete solution into the original PDE. The right-hand side of the resulting modified equation represents the truncation error—the difference between the original PDE and the discretized equation. These terms form the source terms of the ETE. Truncation errors generated in one area of the domain are convected and manifested elsewhere as interpolation errors.

For hypersonic systems, errors arising from numerical discretization may act as erroneous waves, which the equations propagate like physical waves. Mesh adaptation methods that are tied to sources of discretization errors promise to be more efficient than adaptation to local errors alone. For viscous fluxes, the decomposition of the error into two components leads to the development of approximations to the truncation error: (1) error in velocity (assuming that viscosity is correct) and (2) error in viscosity (assuming that velocity is correct). The viscous ETE often captures the difference between the coarse grid and extrapolated results.

Cavallo uses the Richardson extrapolation technique when the solution sequence exhibits monotonic behavior, and then compares these extrapolated results with those obtained with the ETE method. When RE cannot be applied, the ETE method offers a viable alternative approach to solution verification. Cavallo summarizes significant advances toward practical analysis tools for automating error quantification and reduction for CFD.

The objective of most engineering applications is the prediction of specific output quantities, for example, the coefficients of lift and drag of an aircraft. The motivation for automatic goal-oriented simulation verification is to reliably predict the level of discretization error in these target outputs. Moreover, once the error is known, this information can be used to guide adaptive cell refinement to reduce the error below a user-specified tolerance.

Nemec and Aftosmis demonstrate the utility of combining an adjoint-based error estimate with a robust mesh generation tool based on a Cartesian cut-cell method to yield automated simulation verification for inviscid, compressible flow fields. An output functional J(Q) determined from a flow solution Q that satisfies the flow equations forms the basis of their approach.

Nemec and Aftosmis find that their method reliably identifies critical regions of the mesh for adaptation and provides a sharp estimate of discretization error. The CPU cost per iteration for solving the adjoint equations is reported to be approximately equal to the cost of solving the flow equations. An important feature of this approach is that convergence histories of the outputs and error estimates are available automatically. Overall, the presented examples demonstrate the power, performance, and benefits of adaptive mesh refinement with adjoint-based error estimates for verification of engineering simulations for integral output quantities.

On one hand, the Richardson extrapolation technique works well for the integral parameters to categorize the behavior of computed simulations. Grid refinement needs to be conducted at least three times, preferably four times. It is difficult to identify if and when the asymptotic range of the grid convergence is reached; the technique is not used for oscillatory grid convergence. In addition, it does not provide information about how to adapt meshes to reduce the discretization error in the quantity of interest, and is problematic when applied to local variables. On the other hand, the fact that both the ETE approach and the adjoint-based approach when applicable are very useful for simulation verification provides considerable motivation for further research to improve these approaches and to extend their utility to other physics simulation areas.

5. Uncertainty Quantification

"[E]xperience and instinct are poor substitutes for careful analysis of uncertainty." [6]

— Return to Flight Task Group, NASA

The only certain aspect of simulation or measured reality is that it is uncertain. Uncertainty is omnipresent in simulations and tests. The quantified uncertainty is the best measure of accuracy and is required to establish the veracity or validity of a simulation, model, or referent quantity.

The sensitivity-uncertainty analysis is the key to establishing simulation credibility. The management and assessment of uncertainty are critical for fully achieving the promise of modeling and simulation technology—that is, to make simulation-based decisions. Uncertainty management and assessment is conducted with a troika comprising a simulator (the one who simulates), a tester (the one who conducts reality tests), and a user of simulations; see Reference 37.

In Chapter 5, Romero explains some basic concepts related to uncertainty analysis. Romero et al., apply some of these concepts in Chapter 11. In Chapter 6, Swiler and Romero provide a survey of

[6] Another take: *Best practices are poor substitutes for careful analysis of uncertainty.*

advanced probabilistic uncertainty propagation and sensitivity analysis methods. Their focus is on sampling design, sensitivity analysis, and response model approximations. In Chapter 7, Barth discusses the calculation of moment statistics with error bounds using non-intrusive input uncertainty propagation and output uncertainty quantification (UQ) methods, and illustrates these for CFD simulations containing random parameters and fields.

Romero discusses some fundamental and advanced concepts, terminology, definitions, and relationships pertaining to uncertainty quantification and sensitivity analysis. Some important types and sources of uncertainty often encountered in simulations, models, and tests are identified and discussed. With uncertainty propagation methods, uncertainties in inputs for conducting a simulation are transformed into corresponding output uncertainties. For model predictions, these uncertainties are combined with simulation verification uncertainties, extrapolation uncertainties, and any other modeling uncertainties to obtain the overall quantified uncertainty.

Swiler and Romero present some contemporary uncertainty propagation methods for probabilistic uncertainty in computational applications, including sampling methods, optimization-based reliability methods, and response-surface metamodel-based methods including Gaussian processes and polynomial chaos expansions. They also discuss some critical issues that arise when UQ is conducted. The choice of an approach depends on the goal for conducting UQ: an accurate assessment of the response level at a specific percentile or a mean and/or variance of the output distribution. Two approaches for global sensitivity analysis are discussed: correlation and variance-based indices. After providing an overview of these methods, Swiler and Romero also demonstrate and compare sampling, quadrature, sensitivity analysis, use of surrogates, and reliability analysis on a cantilever beam displacement problem.

These advanced propagation methods complement the modern design of experiments based on response surface modeling approaches that are presented by DeLoach in Chapter 8, and the linearized probabilistic UQ approaches that are employed for validation by Romero et al., in Chapter 11. Propagation of probabilistic discrete uncertainties and non-probabilistic interval uncertainties are also explained in Chapter 11.

In Chapter 7, Barth discusses the ongoing development of combined uncertainty and error bound estimates for CFD simulations subject to imposed random parameters and random fields. He illustrates advanced quadrature-based methods for propagating probabilistic uncertainty in CFD. Uncertainty statistics for fluid dynamic quantities of interest are often estimated from a sequence of simulations that account for sources of uncertainty in a simulation model. Unfortunately, the accuracy of these estimated statistics could be severely compromised by the presence of numerical errors of two types: (1) numerical errors occurring in simulations and (2) numerical errors occurring in the calculation of uncertainty statistics integrals using a numerical method.

A major challenge for simulation practitioners is to construct computable error bound formulas for systematically determining how accurately simulations should be computed and how accurately uncertainty statistics integrals should be approximated in order to attain efficient and accurate moment statistics for output quantities of interest for a given computing cost. Formal error bounds are presented for uncertainty statistics that are estimated using simulations containing numerical errors, together with numerically approximated statistics integrals with quadrature error. These error bound formulas provide a quantitative guide when performing simulations by answering the following questions:

- How accurate is a computed moment statistic?

- How does simulation numerical error affect the accuracy of a computed moment statistic?
- How does statistics integral error affect the accuracy of a computed moment statistic?
- To further improve the accuracy of computed output statistics, should additional resources be devoted to solving simulation problems more accurately or improving the accuracy of numerically approximated statistics integrals?

The ability to quantitatively answer these questions and adjust the accuracy of simulations as needed appears to be a powerful new capability in UQ. Barth presents several numerical examples to demonstrate this capability.

Much research has been conducted in advancing the state of the art in forward propagation of input uncertainty and quantifying the output uncertainty, where the input uncertainty is often not known. The assessment of model and input uncertainties, which requires the inverse assessment of uncertainties, is a much more difficult problem than forward uncertainty quantification.

6. Simulation Validation

"As far as the laws of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality."

— Albert Einstein

Simulation validation is conducted in a domain of interest and to establish the validity of both quantities of interest and simulation model. The Modern Design of Experiments (MDOE) approach, including sensitivity analysis, facilitates the design of experiments to collect data for validation and for design of simulation analysis.

In Chapter 8, DeLoach summarizes the MDOE basic tenets of experiment design that can be applied to computational experiments as readily as to the physical experiments for which they were originally developed. Norris illustrates in Chapter 9 the use of CFD in the design of an experiment to validate models. In Chapter 10, Romero explains his Real Space (RS) model validation approach. In Chapter 11, Romero et al., present a comprehensive RS model validation effort, applying processes of UQ to experiments, and V&V and UQ to simulations of heated steel pipes pressurized to failure. The validity of a temperature-dependent, elastic-plastic ductile steel constitutive model is assessed.

DeLoach focuses on response surface modeling experiments. The ultimate objective of such experiments is to obtain sufficient new knowledge about a system to adequately forecast its responses for any combination of independent variables within the range of variables that would change in the experiment. An adequate forecast is taken to mean one for which there is a specified, acceptably high probability that the forecast is within a specified tight tolerance.

The relationship between model adequacy and the volume of data used to construct the model is also illustrated in Chapter 8. The question of how many data points to acquire to minimize cost while achieving objectives, and the related question of which points to acquire, are addressed. Various point distribution criteria are reviewed from the perspective of minimizing response model prediction uncertainty. The importance of implementing various quality assurance tactics during data acquisition is emphasized, in order to ensure independence among response estimates.

Several aspects of model building are reviewed, including a brief summary of regression methods and discussions of dependent and independent variable transformations to improve model

prediction quality. The chapter ends with a discussion of response model quality assessment. Traditional methods for assessing response model adequacy are reviewed, including the plotting of residuals. Relatively new developments in quality assurance are treated, including Critical Binomial Number analysis and the application of Bayesian statistics to account for inference error risk when model response predictions are compared with unbiased response estimates.

Norris established a team of experimentalists and physics modelers to manage uncertainty in the design of an experiment for the validation of CFD models. This team assured that the data being measured was indeed what was required for the modeling effort. Norris details the experiences of the Hypersonic Air-Breathing Propulsion Branch at NASA Langley Research Center in setting up the Isolator Dynamics Research Laboratory experiment. A very close contact and collaboration between the experimentalists and modelers from the inception of the design process led to the determination of the diagnostic approaches to employ and the hierarchy of measurements required—an example of the uncertainty management and assessment troika in action.

As part of the design of the experiment, CFD was utilized to answer design questions for various components as well as to investigate the sensitivity of the facility to changes in inflow and boundary conditions and of manufacturing tolerances on the test article. This allowed the team to design an experiment that would provide the best quality data for the modeling effort and to ensure that the effect of boundary conditions and measurement techniques were understood.

In Chapter 10, Romero describes the RS model validation approach. By working backward from an end objective of "best estimate with uncertainty" modeling and prediction, he developed the RS model validation approach. This objective was realistically pursued within the context of integrated experimental-modeling-analysis programs with end-to-end scope involving experiment design and analysis, data conditioning, model conditioning/calibration, model validation, hierarchical modeling, and extrapolative prediction under uncertainty.

The RS validation approach is so named because it differs from other established frameworks that employ validation metrics in a discrepancy *transform space*: that is, the subtractive difference metric in Reference 21, and the cumulative density function mismatch "area" validation metric in References 38 and 39. In Chapter 10 and Appendix D of Chapter 11, Romero compares his model validation framework with other established model validation frameworks.

For Romero, model validation is still a developing field in engineering and science because a large variety of viewpoints and approaches exist in the literature. In Section 6, Chapter 10, he asserts:

> As in the case study, the assessment of model adequacy often comes in the form of whether the model yields conservative predictions—but not overly conservative—at the validation points in the modeling space. If so, the possibility that this circumstance will be preserved at interpolation or extrapolation points in the modeling space is referred to as 'zeroth-order' adequacy establishment. … Thus, validation assessments rarely yield yes/no statements about whether a model is valid for a proposed use. Rather, validation provides a crucial source of information in a larger overall evaluation of model suitability for the proposed use.

Based on this assertion, the RS model validation approach provides circumstantial evidence but rarely a definitive assessment of the model validity for the proposed use. A larger, overall evaluation is conducted, for example with PIRT and PCMM (Section 3), to assess model suitability for the proposed use. Open questions that need to be addressed include:

- How should higher than zeroth-order adequacy be achieved?
- What should be the process of validating to achieve definitive yes/no assessments?
- If the RS model validation outcome were to assert a non-committal statement about whether a model is valid for a proposed use and a larger overall evaluation assessment of model suitability were to yield a "yes" verdict, would this verdict be correct?

The case study presented in Chapter 11 by Romero et al. involves:

- Material strength characterization and associated constitutive model calibration.
- Model-assisted design and analysis of the validation experiment to characterize and reduce uncertainty.
- Mesh and solver discretization studies (simulation verification) to control and characterize solution error and uncertainty.
- Model validation comparison of experimental and simulation results with uncertainties and interpretation of the results.

The validation experiments and simulations, their uncertainties, and processing of results are described, along with uncertainties for comparison in the RS framework. Relatively simple spreadsheet calculation procedures are explained for the linearized UQ approach used. Methods for representing and propagating interval and probabilistic uncertainties from multiple correlated and uncorrelated sources in the experiments and simulations are demonstrated for: material variability characterized by non-parametric random functions (discrete temperature-dependent stress-strain curves); very limited (sparse) experimental data at the coupon testing level for material characterization and at the pipe-test validation level; boundary condition reconstruction uncertainties from spatially sparse sensor data; normalization of pipe experimental responses for measured input-condition differences among replicate tests and for random and systematic uncertainties in measurement/processing/inference of experimental inputs and outputs; and numerical solution uncertainty from model discretization and solver effects.

7. Simulation of Propulsive Flows

"What we observe is not nature itself, but nature exposed to our method of questioning."

— Werner Karl Heisenberg

In Chapter 12, Riggins provides a brief overview of key physics and modeling classifications associated with hypersonic air-breathing propulsion systems. The emphasis is on an airframe-integrated scramjet engine, including the dual-mode scramjet. Key physical processes and attendant critical technical issues and challenges for modeling, simulation, and analysis are reviewed. These physical processes affect both performance and operability of the scramjet and the dual-mode scramjet engine. This discussion of physical issues is followed by the broad categorization and ordering of various M&S approaches. In general terms, fidelity of physics considered for modeling and specific methodology of simulation are discussed. Example results from a low-fidelity modeling analysis of a scramjet engine/vehicle in flight are then provided in terms of typical flow characteristics and related performance metrics and parameters.

Riggins presents comments regarding the credibility of simulations and some general observations regarding modeling and simulation in high-speed propulsion systems. The objective of these simulations is to provide performance and/or operability information that will facilitate

analysis, design, and optimization of a system or system component. Simulation credibility, as related to performance parameters, depends on what level of physics is considered adequate to address a particular analysis task. Propulsion simulations need to include computations of quantities relevant for performance assessment. Many challenging performance and operability issues must be considered in the design of hypersonic air-breathing propulsion systems. As there are numerous sources of uncertainty, Riggins recommends increased attention to and effort on the development of robust uncertainty quantification methods.

In Chapter 13, Sankaran et al., discuss high-fidelity physics-based modeling of liquid propulsion systems, rockets, gas turbines, and scramjets. They provide an overview of liquid propulsion and focus on dominant physical phenomena such as injection and mixing, heat transfer, and combustion/ignition dynamics. Correspondingly, they categorize the simulation objectives into these three main areas. In each area, the simulations are subdivided into two levels: steady or unsteady Reynolds-averaged Navier-Stokes (RANS) models, and large eddy simulation (LES) or detached eddy simulation (DES) models. Simulation objectives for each area and for each level are stated to assist the analysis and design of propulsion components. A summary of physical modeling approaches to analyze the selected dominant phenomena is presented.

Sankaran et al., classify the uncertainties as arising from either physics modeling or the numerical simulation procedure. Modeling uncertainties are owing to the limitation of the models for turbulence, combustion, and multi-phase phenomena. A fundamental issue involves the definitive interpretation of the eddy viscosity in LES models in the presence of numerical dissipation errors. Both combustion kinetics and turbulent combustion closure introduce modeling uncertainties for combustion problems. The models for primary and secondary atomization also introduce uncertainties. The techniques to quantify numerical uncertainty in LES cases are still in their infancy. The grid resolution required in LES simulation can vary for different algorithms and codes. In the case of multi-phase flows, grid-converged solutions are extremely challenging. Validation of liquid propulsion simulations is challenging because of complex physics, complex geometries, and difficulties in acquiring detailed test data.

In Chapter 14, Eaton discusses the primary objectives for modeling and simulating solid rocket motor (SRM) chamber and nozzle environments. These objectives can be divided into two main areas: (1) designing the chamber propellant configuration and nozzle to meet performance requirements; and (2) provide boundary conditions (pressure, temperature, surface heat flux and shear forces, and droplet impingement magnitude) that can be used by motor component designers to optimize the design configuration within performance, physical (materials, size, and weight), and cost constraints and to meet reliability requirements. To achieve these objectives, modeling and simulation tools have been developed that characterize the major physical mechanisms related to solid propulsion rocket motor environments and performance.

Eaton presents a broad overview that includes solid propulsion physics, M&S objectives and approaches, an example of SRM modeling and simulation results, and a list of primary sources of M&S uncertainties. The dominant uncertainties are owing to uncertainties in burn-rate data, lack of flow field data at typical operating temperatures and pressures, lack of material heat flux data, two-phase flow uncertainties, material property uncertainties, and combustion/reactive flow modeling uncertainties. He recommends that the best design practices involve a close relationship between numerical experiments and physical experiments. This relationship is a part of the uncertainty management and assessment troika mentioned in Section 5, above.

8. Concluding Remarks

"What can be asserted without evidence can also be dismissed without evidence."

— Christopher Hitchens

The focus of this volume is on simulation credibility—a quantitative assessment of simulation credibility. Some state-of-the-art philosophies, principles, and frameworks are presented. The authors involved in this publication have been dedicated to advancing simulation credibility. They discuss and illustrate key advances in techniques, approaches, and processes for quantifying simulation credibility over last 10 years. Crucial modeling and simulation issues are presented to conduct simulations in the discipline of high-speed propulsion.

Simulation credibility is the principal omnipresent measure in the utilization of simulated quantities. Simulation validation is the most important process for establishing this credibility. Validation assesses the credibility of a simulation by estimating the accuracy of simulation from the perspective of intended uses of the simulation. It involves simulation verification, input uncertainty quantification and propagation, simulation model uncertainty, referent uncertainty, and uncertainty quantification and aggregation related to simulated quantities of interest. Simulation validation, both for simulation model validation and prediction credibility, is often a difficult task. If this task cannot be accomplished with direct quantitative evidence, then a combination of direct, quantitative, circumstantial, and qualitative evidence is used. The challenge remains: how to use this combination effectively.

An uncertainty management and assessment troika—simulator, tester, and user—is recommended for managing and assessing uncertainties. The decision maker and other such users of simulations need to know uncertainties of simulations in order to make critical decisions and to properly use simulations, respectively. The responsibility of assessing simulation credibility lies with the simulator.

The anticipation is that simulators—creators of simulations—in all continuum physics fields will find the presented advanced concepts and theories, detailed processes, and illustrated examples in this volume very useful to guide and assist them in establishing the credibility of their simulations and then in communicating this credibility to users of simulations.

References

1. Richardson, L. F., "The Approximate Arithmetical Solution by Finite Differences of Physical Problems Involving Differential Equations, with Application to the Stresses in a Masonry Dam," *Philosophical Transactions of the Royal Society of London*, Vol. 210A, pp. 307–357, 1910.
2. Richardson, L. F., *Weather Prediction by Numerical Process*, Cambridge University Press, 1922, reprint Dover, New York, 1965.
3. Thom, A., "An Investigation of Fluid Flow in Two-Dimensions," Aeronautical Research Committee, R&M 1194, Great Britain, Nov. 1928.
4. Kawaguti, M., "Numerical Solution of the NS Equations for the Flow Around a Circular Cylinder at Reynolds Number 40," *Journal of the Physical Society of Japan*, Vol. 8, pp. 747–757, 1953.

5. Mehta, U. B., “Computational Requirements for Hypersonic Flight Performance Estimates,” *Journal of Spacecraft*, Vol. 27, No. 2, Mar.-Apr. 1990, pp. 103–112.

6. Department of Defense, "DoD Modeling and Simulation (M&S) Management," DoD Directive 5000.59, Jan. 4, 1994.

7. Mehta, U. B., “Guide to Credible Computer Simulations of Fluid Flows,” *Journal of Propulsion and Power*, Vol. 12, No. 5, 1996, pp. 940–948; also AIAA Paper 1995-2225, June 1995.

8. Mehta, U. B., “Simulation Credibility Level,” *5th Joint Army-Navy-NASA-Air Force (JANNAF) Modeling and Simulation Subcommittee Meeting*, CDJSC 49, CPIAC, Johns Hopkins University, May 2007.

9. Mehta, U. B., “Some Aspects of Uncertainty in Computational Fluid Dynamics Results,” *Journal of Fluids Engineering*, Vol. 113, Dec. 1991, pp. 538–563.

10. Mehta, U. B., “Credible Computational Fluid Dynamics Simulations,” *AIAA Journal*, Vol. 36, No. 5, May 1998, pp. 665–667.

11. Warming, R. F., and Hyett, B. J., “The modified equation approach to the stability and accuracy analysis of finite difference methods,” *Journal of Computational Physics*, Vol. 14, pp. 159–179, 1974.

12. DeLoach, R., “Applications of Modern Experiment Design to Wind Tunnel Testing at NASA Langley Research Center,” *36th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 1998-0713, Jan. 1998.

13. DeLoach, R., and Micol, J. R., “Comparison of Resource Requirements for a Wind Tunnel Test Designed with Conventional vs. Modern Design of Experiments Methods,” *49th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2011-1260, Jan. 2011.

14. Jiang, X., and Mahadevan, S. “Bayesian Optimal Design of Validation Experiments,” *Proceedings of the 25th International Modal Analysis Conference*, Feb. 2007.

15. Schultze, J. F., Hemez, F. M., Doebling, S. W., and Sohn, H., “Statistical Based Non-linear Model Updating Using Feature Extraction,” *Proceedings of the 19th International Modal Analysis Conference*, Feb. 2001.

16. Hemez, F., Wilson, A, and Doebling, S., “Design of Computer Experiments for Improving an Impact Test Simulation,” *Proceedings of the 19th International Modal Analysis Conference*, Feb. 2001.

17. Doebling, S., Hemez, F., Schultze, J., and Cundy, A., “A Metamodel-Based Approach to Model Validation for Nonlinear Finite Element Simulations,” LA-UR-01-5870, *Proceedings of the 20th International Modal Analysis Conference*, Feb. 2002.

18. Chen, W., Baghdasaryan, L., Buranathiti, T., and Cao, J., “Model Validation via Uncertainty Propagation and Data Transformations,” *AIAA Journal*, Vol. 42, No. 7, 2004, pp. 1406–1415.

19. International Organization for Standardization (ISO), *Guide to the Expression of Uncertainty in Measurement* (corrected and reprinted, 1995), ISO, Geneva, Switzerland, 1995.

20. Coleman, H. W., and Stern, F., “Uncertainties in CFD Code Validation,” *Journal of Fluids Engineering*, Vol. 119, pp. 795–803, Dec. 1997.

21. American Society of Mechanical Engineers (ASME), *Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer*, ASME V&V 20-2009; reaffirmed 2016.

22. Shi, J., Zhang, Y.-T., and Shu, C.-W., "Resolution of High Order WENO Schemes for Complicated Flow Structure," *Journal of Computational Physics*, Vol. 186, 2003.

23. Moffat, R. J., "Describing the Uncertainties in Experimental Results," *Experimental Thermal and Fluid Sciences*, Vol. 1, pp. 3–17, Jan. 1988.

24. National Aeronautics and Space Administration (NASA), *Standard for Models and Simulations*, NASA-STD-7009A, 2016; superseding NASA-STD-7009, 2008.

25. American Institute of Aeronautics and Astronautics (AIAA), *Guide for the Verification and Validation of Computational Fluid Dynamics Simulations*, AIAA-G-077-1998; reaffirmed 2002.

26. American Society of Mechanical Engineers (ASME), *Guide for Verification and Validation in Computational Solid Mechanics*, ASME V&V 10-2006; reaffirmed 2016.

27. Department of Defense, "DoD Modeling and Simulation (M&S) Verification, Validation, and Accreditation (VV&A)," DoD Instruction (DoDI) 5000.61, Dec. 9, 2009; superseding the 2003 version which superseded the 1996 version.

28. Romero, V. J., "Elements of a Pragmatic Approach for dealing with Bias and Uncertainty in Experiments through Predictions," Sandia Report, SAND2011-7342, Nov. 2011.

29. Schlesinger S., Crosbie, R., Gagne, R., Innis, G., Lalwani, C., Loch J., et al., "Terminology for Model Credibility," *Simulation,* Vol. 32, No. 3, pp. 103–104, Mar. 1979.

30. Department of Defense, Modeling and Simulation Coordination Office, "VV&A & Glossary," 2011.

31. National Aeronautics and Space Administration, *NASA Policy Directive on Technical Standards*, NPD 8070.6B, May 2003.

32. Hills, R. G., Witkowski, W. R., Trucano, T. G., and Urbina, A., "Development of a Fourth Generation Predictive Capability Maturity Model," Sandia Report SAND2013-8051, 2013.

33. Shaw, R. A., Larson, T. K., and. Dimenna, R. K., "Development of a Phenomena Identification and Ranking Table (PIRT) for Thermal-Hydraulic Phenomena During a PWR Large Break LOCA," INEL, NUREG/CR-5047, Nov. 1988.

34. Hills, R. G., Maniaci, D. C., and Naughton, J. W., "V&V Framework," SAND2015-7455, Sept. 2015.

35. Floyd, R. W., "Nondeterministic Algorithms," *Journal of the Association for Computing Machinery*, Vol. 14, No. 4, pp. 636–644, Oct. 1967.

36. Zhang, X. D., Trepanier, J.-Y., and Camerero, R., "A Posteriori Error Estimation for Finite-Volume Solutions of Hyperbolic Conservation Laws," *Computer Methods in Applied Mechanics and Engineering*, Vol., 185, pp.1–19, 2000.

37. Mehta, U. B., "Simulation and Test Uncertainties and Their Management," Distinguished Invited Lecture, *15th AIAA International Aerospace Planes and Hypersonic Systems And Technologies Conference*, 28 Apr. – 1 May 2008.

38. Roy, C. J., and Oberkampf, W. L., "A Comprehensive Framework for Verification, Validation, and Uncertainty Quantification in Scientific Computing," *Computer Methods in Applied Mechanics and Engineering*, Vol. 200, pp. 2131–2144, 2011.

39. American Society of Mechanical Engineers (ASME), *An Illustration of the Concepts of Verification and Validation in Computational Solid Mechanics*, ASME V&V 10.1-2012.

Code Verification for Phenomenological Modeling and Simulation Software

Patrick Knupp[1]

Sandia National Laboratories,[2] *Albuquerque, NM 87185*

1. Introduction

Code verification is an important activity in software development that logically occurs prior to many other activities within the process of verification and validation (V&V). Code verification supports these other activities by providing an assessment of the software or portions thereof. The American Society of Mechanical Engineers (ASME) gives a high-level definition of code verification in their guide as *"Activities that 'establish confidence, through the collection of evidence, that the mathematical model and solution algorithms are working correctly'"* [1].

In addition, the guide provides an extensive description of code verification. Although the guide is an excellent resource, those engaged in the field do not express a unanimous opinion on the end products of code verification. The use of the ambiguous phrase "working correctly" is appropriate given the lack of agreement. In addition, there are important research issues in code verification that remain to be settled. The intention of this chapter is to provide context for code verification and to describe the main concepts and practices, and does not try to resolve the debate over the meaning of "working correctly."

Within V&V, there are two kinds of verification: code verification and solution (or calculation) verification. Code verification provides an assessment of relevant portions of the software and codes, while solution verification provides an assessment of the particular numerical solutions that are to be compared to specific experiments. The relation between code verification and solution verification is covered in Section 5 of this chapter.

Ideally, code verification occurs as part of the software engineering cycle. Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, and encompasses knowledge, tools, and methods for defining software requirements, software design, computer programming, user interface design, software testing, and maintenance. The development of requirements is a critical early stage in software development and takes into account such things as the platforms on which the software will run, the input/output specifications, speed and memory requirements, code maintenance, version control and tracking, and many other concerns. To demonstrate that the requirements have been met, the software is tested to show, for example, that the code will compile on all the

[1] At Dihedral, LLC, Bozeman, MT at the time of publication

[2] Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

required platforms, that all components are properly linked and interact correctly, that speed and memory requirements are met, that the expected behavior occurs, that errors and exceptions are properly handled, that the answer today is the same as the answer yesterday (unless it was intentionally changed), and that the expected answer is produced. Testing methods include unit and function, integration, stress, performance, memory, portability, and regression tests [3, 4].

Software engineering applies to all software projects whether the application is business, banking, medical, engineering, or gaming. Different aspects of software engineering can be emphasized, depending on the requirements of a given software development project. Reference 2 says:

"Software [engineering] processes determine whether the development products of a given activity conform to the requirements of that activity and whether the software satisfies its intended use..."

2. Code Verification Concepts

This discussion is focused on codes whose primary objective is to solve partial differential equations (PDEs) for heat transfer, fluid flow, and chemical kinetics found in chemical propulsion and energetics modeling, although much of it also applies to other PDE-solving codes. Codes whose primary objective is to solve simpler mathematical models, such as discrete models, networks, and sets of equations, generally require less extensive types of tests than PDE-codes. To be verified, PDE codes require, in addition to the standard types of testing, specialized tests not commonly used within the framework of software engineering. These specialized tests, although part of code verification, have often been overlooked in the past. This is partly due to the complexity of the numerical algorithms that are used to solve PDEs and partly due to the fact that PDE codes constitute a small fraction of the software universe. These specialized tests will be described later in this chapter.

An important fact about code verification is that it is not necessary that the mathematical model be an adequate representation of physical reality. The function of code verification, as understood by the V&V community, is simply to ensure that the numerical algorithms that solve the mathematical model (physically correct or not) are working properly, so that one is not mislead when using the code. The mathematical model and its software implementation is an input to the code verification process. In order to perform code verification, one must know the details of the mathematical model so that its implementation can be appropriately tested. It is usually not enough to say that the mathematical model is, for example, the heat equation. Rather, one must know detailed descriptions of the governing equations, boundary and initial conditions, assumptions on the modeled domain, PDE coefficients (including material models), and source terms. Additionally, the mathematical model includes the data or input associated with the values of the parameters and coefficients. Ideally, this type of information is readily available.

Numerical analysis methods are used for discretizing PDEs to achieve consistency, stability, convergence, and for understanding important properties such as the discretization error and order-of-accuracy of a particular algorithm. Numerical algorithms are sets of instructions describing the specific computations one must perform to solve the PDEs. As is often the case, in the development of PDE-solving software, it is unclear in advance as to which of several potential discretizations and numerical algorithms are best (in terms of accuracy, efficiency, and robustness) to solve the given mathematical model, particularly with regard to a challenging application. Code verification, as understood by the V&V community and as applied to PDE

codes, must therefore include the following two types of specialized testing:

- *Numerical Algorithm Property Verification (NAPV)*: Determining whether the relevant numerical algorithms within a code have been implemented in agreement with their proven mathematical properties, such as order-of-accuracy, stability, and consistency.
- *Numerical Algorithm Adequacy Verification (NAAV*): Determining whether relevant numerical algorithms within a code satisfy the accuracy, robustness, and speed requirements of their intended use.

Numerical algorithm property verification is concerned with whether the implementation in the code is consonant with the theoretically proved algorithm properties (if any). This can be determined by comparing the numerical results that the code produces to reference solutions. An example of NAPV is order verification, in which one determines whether the numerical solution converges to an exact solution at the same rate as its theoretical order-of-accuracy predicts. If it does not, then experience has shown that the relevant numerical algorithms may have (1) been implemented incorrectly, or (2) the theoretical order-of-accuracy is not what one thought. Most agree that order verification is an essential step in PDE code verification due to its particularly effective ability to determine the correctness of algorithm implementations.

Numerical algorithm adequacy verification is concerned with whether or not the algorithm is adequate (in terms of accuracy, robustness, and speed) to solve the mathematical model associated with a particular application such as propulsion modeling within a V&V process or within a modeling and simulation project. NAAV is important because even if an algorithm has been correctly implemented, it does not necessarily mean that it is adequate for its intended use. Thus, it is the application requirements that will determine the adequacy of the algorithm and code. It is helpful to keep in mind the distinction between NAPV and NAAV, as these characteristics can help sort out the purpose of different tests and determine task priorities within the code verification process. For a given algorithm, NAPV should usually be performed prior to NAAV because the adequacy of an algorithm often cannot be properly established if it has not been correctly implemented.[3]

Both NAVP and NAAV can, and should, be performed by the developers of the code in question. Developers usually have a broad set of applications in mind for their code and, as such, the intended use of the code (as stated) may be fairly general. In this setting, code verification must be broad in order to cover all the intended uses of the code. On the other hand, when the code is to be used in a V&V exercise (or in an end-use case), the intended use of the code becomes quite specific. Code verification is relevant in all these cases.

[3] If the properties of a numerical algorithm are not mathematically proven, then NAPV is precluded for that algorithm. However, the properties of the algorithm can still be explored through computational experiments. For example, there may be no proof that a given algorithm is second order accurate, but one can infer the order-of-accuracy by comparing the numerical solutions the algorithm produces to an exact solution. It may turn out that the algorithm exhibits only first order accuracy. This activity does not ensure that the algorithm is correctly implemented, but is useful within NAAV: is a first order algorithm good enough for the intended application?

3. Code Verification Testing Practices

Testing is an important part of code verification because it plays a major role in the collection of evidence. Various kinds of tests are used, including unit tests, stability tests, robustness tests, tests of numerical options (e.g., patch tests, limiter tests), benchmark tests involving comparison to exact solutions to the PDE, and mesh refinement studies, with or without exact solutions. Some tests can be called NAPV tests since they are geared toward the investigation of implementation correctness by showing that the results agree with proven mathematical properties of the algorithms. Other tests can be called NAAV tests since they investigate questions related to algorithm adequacy.

One of the best and most revealing practices in algorithm verification is the use of tests involving systematic mesh refinement and the comparison of the numerical solutions to a predetermined exact solution in order to compute the discretization error when in the asymptotic regime. The acceptance criteria for such a test can be either that (1) the error decreases to zero as the mesh is systematically refined (consistency verification) or (2) the error decreases at the expected rate (order verification) [9]. Assuming the test setup was not flawed, failure to meet the acceptance criteria in tests of this kind generally means that (if the algorithm is well-ordered) the algorithm was not correctly implemented.

Mesh refinement can be valuable as a NAAV test, for example, to determine an unknown theoretical order-of-accuracy or to identify algorithmic deficiencies by comparing the family of numerical solutions to some exact solution (including manufactured ones). In that case, the order-of-accuracy of algorithm is not verified, but rather established by computational means.

A collection of tests that are performed for the purpose of code verification is called a verification test suite (VERTS). Verification test suites should be well designed, comprehensive, and maintainable. A well-designed VERTS is one in which the collective set of tests, when passed, supply compelling evidence to show that the solution algorithms are verified and adequate. A comprehensive VERTS is one in which the test suite properly tests all the code's features and capabilities with respect to the intended application, ensuring there are no coverage gaps.

An important tool in the creation of complete VERTS is the Method of Manufactured Solutions (MMS) [10-12]. MMS is a form of order verification in which the exact solution is created in a non-traditional manner. Exact solutions created in the traditional way begin with the statement of the mathematical problem (the PDEs, domain, boundary conditions, coefficients, etc.), and given the problem and associated data, the exact solution is found using classical methods such as separation of variables, series solutions, integral methods, or Green's functions. A major limitation of this approach is that classical solutions can usually be found only by simplification of the mathematical model (e.g., by reducing it to one-dimension, by assuming constant coefficients, by using simple geometric domains, or by linearization). For the purpose of rigorous code verification, this simplification is inadequate because, in many of the intended uses of the code, the algorithms are used in their higher-dimensional form. Establishing that the algorithm can correctly reproduce a 1D solution does not prove that the full higher-dimensional algorithm is correctly implemented. Therefore, if one is restricted to only classical solutions, the test suite will contain significant gaps. The MMS can fill many of these gaps since the method by which the exact solution is created is different. It chooses the exact solutions first and then applies the PDE operators to the chosen solution to find a source term that balances the equations. In this manner, one can create tests to cover the general features and capabilities of the code, including algorithms, which apply in higher dimensions, in terms of showing that the PDEs are solved correctly. MMS is primarily intended as a technique for NAPV testing because its use of order-

verification is aimed at establishing that algorithms have been implemented correctly.

MMS is sometimes criticized due to the occasional necessity of introducing one or more numerical source terms, thus modifying the very code one is to test. This objection is less serious than it might seem because MMS is a self-correcting procedure. For example, if a coding mistake is introduced into the code due to the addition of a source term, the MMS test will very likely detect it due to its extreme sensitivity. Furthermore, if it is established by MMS that the algorithms in the code (including the source term) are correctly implemented, then how can it be incorrect without the source term?

There are important open questions concerning MMS and, more specifically, order verification. For example, there are numerical algorithms used in shock physics, structural dynamics, and fire combustion whose theoretical convergence properties are poorly understood and thus it is unclear in these examples how to interpret a failure to converge to a stationary solution when mesh refinement is employed.

Manufactured solutions are rarely suited for use in nearby problems intended to establish adequacy. Nearby problems require physically realistic exact solutions, a weakness of manufactured solutions.

Finally, in a well-maintained VERTS, all tests can be run on demand. In other words, the tests will run with the latest release version of the code so that one can perform regression-like verification tests as the code progresses through different versions to ensure that solution algorithms that worked in earlier release versions also work in the current release version. Further, this process ensures that the evidence collected to show that the code has been properly verified pertains to the latest versions, and that the evidence is available to anyone that needs to see it, whether they be the developers, the code users, or other stake-holders.

A commonly overlooked requirement when developing modeling and simulation software is the need to ensure that the code is verifiable [13]. Five important attributes of verifiable code are: (1) complete documentation of the mathematical model and the solution algorithms, (2) creation of a map linking the mathematical model directly to the software, (3) the ability to input distributed source term data for both the interior equations and the boundary conditions to permit MMS tests, (4) the distributed source terms are incorporated into the mathematical model and numerical algorithms, and (5) the ability to automatically compute discretization error in NAPV tests by comparing the numerical solution to the exact solution.

4. Benefits of Code Verification

Code verification, when done properly, has many benefits. First and foremost, it plays an important role in the V&V process because it is the first step in the sequence of code verification, solution verification, simulation and model validation, sensitivity analysis, and uncertainty propagation and quantification. All the activities in this sequence are put at risk if code verification is not done or not done well, because incorrect or inadequate algorithms can mean misleading numerical results or even no numerical results at all. Code verification has various other benefits including: (1) it assists in the development of improved numerical algorithms; (2) codes accompanied by evidence of proper verification will be more likely to be used than codes which do not have such evidence; (3) when performed as an integral part of code development, it ensures that coding mistakes do not mislead algorithm developers as to the adequacy of their algorithms and ensures that their experience over time is based on sound code; (4) order

verification encourages the use of ordered discretization models and discourages the use of non-ordered approximations; (5) it increases code/algorithm developer confidence, knowledge, and professionalism; and (6) it assists in developing the software and analysis products right.

5. Code Verification and Solution Verification

In practice, it may happen that not all parts of a given code have been properly verified prior to its use in V&V or even in end-user activities. This can happen for a variety of reasons such as (1) overlooking a relevant part of an algorithm, (2) improperly testing it, (3) insufficiently testing it, (4) not adequately documenting the verification, or (5) not including the tests within the set of regression tests. Therefore, it behooves anyone engaged in solution verification, validation, uncertainty quantification, and anyone who is an end-user to identify the specific algorithms within the code that their application will use.[4] Identifying the relevant parts of the code should be done prior to use and it should then be determined whether or not the relevant parts have been verified and to what extent. If testing or documentation deficiencies are found, then the relevant algorithms in the code must be additionally verified, through appropriate use of NAVP and NAAV, prior to use of the code.

Within the context of a specific intended use of the code (say, within a solution verification exercise), one can say that a collection of algorithms used in one or more simulations that are performed for the purpose of the prediction and its verification is adequate if, within the memory and efficiency limits of the selected computing platform, the algorithms produce correct numerical solutions that are both converged and asymptotic. In this definition, "converged" means that both the solver and non-linear iteration loops were terminated by satisfying acceptably low relative error tolerances on the residual or on other numerical quantities of interest; and "asymptotic" means that the numerical solution, as a function of mesh-discretization size, can be shown via mesh refinement studies to be in the asymptotic range of mesh convergence. Under this narrow definition, adequacy is directly determined during the process of creating numerical solutions for the calculation of numerically precise error estimates and error bars, (i.e., in a validation exercise or in the process of using the code as an end-user). In this context, the activities involved in both NAVP and NAAV are often considered to be part of the solution verification process.

Code users, including those engaged in various V&V processes desire evidence that provides them with confidence that mathematical model and its solution algorithms are working correctly. Although there are technical software engineering and code verification drivers, another driver of the need for confidence is due to the fact that there is often a division of labor in which one group of people function as code developers and another group functions as code users. The first group develops and tests code capabilities, while the second group (sometimes referred to as analysts) performs the predictions and their calculation verification. Both groups require evidence that gives them confidence in the code and algorithms. The preferred evidence is frequently based on running the code on nearby problems, which involve performing a test that resembles the ultimate

[4] This determination can be systematically addressed through code coverage tables, which explain the mapping between the tests in the verification test suite and the portions of the code that are exercised in a particular prediction simulation [5, 6].

application in one or more of its essential aspects, but removes non-essential features of the ultimate application so that important behavior can be isolated and the testing is less cumbersome. In this context, passing a series of nearby problem tests provides evidence that the code is working correctly.[5] Examples of algorithm adequacy testing performed by developers include: efficiency tests, robustness tests, iterative convergence tests, numerical artifact tests, symmetry tests, benchmark tests [8], and many others. The confidence issue arises in several contexts:

1. The decision by the users regarding the selection of the code that is to be used in the V&V exercise and beyond.
2. The hand-off between the code developers and the analysts who will perform calculation verification.
3. The subsequent use, improvement, and advertisement of the code after a V&V exercise are completed.

References

1. American Society of Mechanical Engineers, *Guide for Verification and Validation in Computational Solid Mechanics*, American Society of Mechanical Engineers (ASME) V&V 10 – 2006, 36p, 2006.
2. Institute of Electrical and Electronics Engineers, *Standard for Software Verification and Validation*, IEEE Standard 1012, IEEE Computer Society, 2004.
3. Bezier, B., *Software Testing Techniques*, International Thompson Press, 1990.
4. Sommerville, I., *Software Engineering*, Addison-Wesley, 2005.
5. Dowding, K., Blackwell, B., and Knupp, P., "Demonstrating Code Verification Methods with Calore," Sandia National Laboratories report SAND2007-5612, Albuquerque NM, 2007.
6. Knupp, P., and Ober, C., "A Code-Verification Evidence-Generation Process Model and Checklist," Sandia National Laboratories report SAND2008-4832, Albuquerque NM, 2008.
7. Oberkampf, W. L., and Trucano, T. G., "Verification and Validation in Computational Fluid Dynamics," Sandia National Laboratories report SAND2002-0529, 2002.
8. Oberkampf, W., and Trucano, T., "Verification and Validation Benchmarks," Sandia National Laboratories report SAND2007-0853, 2007.
9. Knupp, P., Ober, C., and Bond, R., "Measuring Progress in Order-Verification within Software Development Projects," *Engineering with Computers*, Vol. 23, 2007.
10. Roache, P. J., "Verification of Codes and Calculations," *AIAA Journal*, Vol. 36, No. 5, May 1998.

[5] It should be noted, however, that passing a set of nearby problem tests does not guarantee that the code will then provide an acceptable solution to the simulations to which they are supposedly "near."

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

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

13. Roache, P., "Building Codes to be Verifiable and Validable," *Computing in Science & Engineering*, pp. 30-38, September/October 2004.

Procedures for Calculation Verification

William J. Rider[*], James R. Kamm[†1], V. Gregory Weirs[*]

[*]*Sandia National Laboratories, Albuquerque, NM 87185*
[†]*Los Alamos National Laboratory, Los Alamos, NM 87545*

1. Overview

Calculation or solution verification is a process by which the discretization or numerical error is estimated in simulations of problems of interest. Calculation verification employs techniques of both error estimation and uncertainty quantification (UQ). There are defined procedures by which numerical error estimates can be converted to numerical uncertainty estimates. Each of these techniques will be discussed here. We take the two terms calculation verification and solution verification to be synonymous. Code verification is a related, but distinct process where the correctness of a software implementation of a numerical algorithm is evaluated, typically by comparison against an exact solution.

Numerical methods that are used to obtain approximate numerical solutions of continuum models unavoidably lead to errors in the computed results. These errors are associated with the numerical method *alone* and have nothing to do with any assumptions related to the form of the continuum models (e.g., model-form errors). The challenge of calculation verification is to help provide estimates of such numerical errors. These errors are of four general types: (1) round-off errors, (2) sampling errors, (3) iterative (linear and nonlinear) solver errors, and (4) discretization errors.

Round-off error is an inevitable consequence of finite-precision representation of numbers on computers; with high-precision arithmetic, round-off errors typically have a smaller effect on computed results than the other primary error sources. Sampling errors occur, e.g., with methods that are inherently stochastic (e.g., Monte Carlo methods) or in sampling a system response quantity of interest. Iterative solver errors are associated with the solution of any linear or nonlinear systems of equations encountered during the course of the computational solution update procedure. In practice, iterative errors may be difficult to reduce to the point where they can be neglected. Discretization errors are a direct consequence of the numerical scheme used to obtain a discrete approximation of continuous equations (e.g., finite difference, finite element, or finite volume methods) and the solution approach used on those discrete equations. For time-dependent problems, both the spatial and temporal discretizations enter into the evaluation of these errors. Many researchers contend that discretization error is often the dominant source of numerical error in scientific computing simulations. This is consistent with much of the authors' experience, although nonlinear solver error can dominate strongly coupled (stiff) problems.

[1] Retired at the time of publication

The important concepts to consider regarding iterative solver error are covered by Eça and Hoeckstra [6] and Oberkampf and Roy [22]; we briefly summarize the issues here. The use of direct solvers for linear systems of equations has become impractical for reasons of efficiency; so iterative solvers are the usual approach. The iterative solution of linear systems of equations is necessarily inexact and it generates a source of error that should be estimated. Usually, the iteration will terminate based on criteria not related to the actual error in the solution. This is common if the iterative solver is developed as part of a general-purpose software package. In this case, the estimation of error associated with the iterative solver is essential. Furthermore, the default termination criteria may be unsatisfactory for the application-specific context. In any case, the uncertainty contribution must be assessed with care. The iterative solution of nonlinear systems, which typically involves both linear and nonlinear stopping criteria, has the same issues.

Among the most important characteristics of discretization schemes is the order of accuracy (also called the convergence rate), which is given by the exponent in the power law relating the numerical truncation error to the value of a parameter associated with the discretization, usually given by the size of the computational cell or time step. The factor multiplying this term measures the overall error of a given scheme; thus, two different schemes that converge at the same rate may have different (absolute) discretization errors. The standard method for estimating this accuracy is systematic mesh refinement (or variation), although there are other, less general approaches [29]. The results of this approach are combined with error measurement to produce the observed rate of convergence, which is compared with the ideal or theoretical rate-of-convergence of the underlying algorithm. In calculation verification, unlike in code verification, the use of an analytical or exact solution to a problem is not available as an unambiguous fiducial solution. Instead, the comparisons are made between solutions using different grid resolutions.

To aid analysts in conducting solution verification analyses, the following workflow process for calculation verification is proposed.

1. Starting with an algorithm implementation (i.e., code) that has passed the appropriate level of software quality assurance and code verification, choose the software executable to be examined.
2. Provide an analysis of the numerical method as implemented, including accuracy and stability properties. (This information should be available from the code verification analysis.)
3. Produce the code input to model the problem(s) of interest.
4. Select the sequence of mesh discretizations to be examined for each problem, and the input necessary to accomplish these calculations.
5. Run the code and provide the means of producing appropriate metrics to evaluate the difference between the computed solutions, based on numerical parameters within the control of the code user. This can also include the numerical method chosen (order of approximation or scheme).
6. Use the comparison (i.e., difference) in Step 5 to determine the sequence of estimated errors corresponding to the various discretizations and tolerances.
7. Use the mesh sequence to allow the determination of the rate-of-convergence for the method, which is then compared to the theoretical rate. For iterative solver errors, the error as a function of the stopping criteria and the discretization must also be determined.
8. Using these results, render an assessment of the accuracy (level of error estimated) for the simulation for a given set of numerical settings.

9. Examine the degree of coverage of features in an implementation by the verification testing.

This report contains detailed descriptions of many aspects of calculation verification, as well as a discussion of the general workflow outlined above. The views contained herein are not meant to be a complete prescription by which to conduct solution verification. Instead, they are intended to guide the conscientious analyst in conducting studies that will contain defensible estimates of the numerical errors inherent in computed solutions for complex engineering simulations.

2. Verification and its Relation to Other Similar Activities

Numerical simulations are increasingly used to: increase understanding; "solve" problems; design devices, vehicles and buildings; predict responses of complicated engineering systems; and to inform decision makers of the level of simulation credibility. Different assessment techniques are employed to address the different ways numerical simulations can fail to provide accurate information. Ultimately, the combined application of all the techniques provides enhanced confidence that the accuracy of the numerical simulation process is adequate for a particular scenario.

The purpose of scientific simulation software differs from that of most commercial software. Verification is needed for scientific simulation codes because this software is designed to produce approximate solutions to mathematical problems for which (i) the exact solution is not known and (ii) knowledge of the error is potentially as valuable as knowledge of the solution Due to these distinguishing and critical aspects of scientific simulation codes, software quality practices from the broader industry (e.g., regression testing) are necessary, but not sufficient particularly for high-consequence scientific simulation codes.

Verification analysis of scientific simulation codes is an example of the assessment of a complex system for which the systematic gathering of appropriate evidence is required. While tests may demonstrate that software is manifestly incorrect, there is no clear-cut procedure with which to "prove" unambiguously that software behavior is, indeed, correct. Thus, the process by which relevant verification evidence is generated and interpreted requires knowledge of the entire simulation and analysis chain. Such knowledge includes understanding of:

- The system being simulated (e.g., the relevant physics, physics models, and these models' representations in mathematical equations);
- The nature of the simulation (including the algorithms used to obtain approximate solutions to the mathematical equations, these algorithms' limitations, the associated numerical analysis, and the software implementation of those algorithms); and
- The process by which the code results are analyzed in the verification process (including, e.g., theory, implementation, and interpretation of convergence analysis).

This body of knowledge is both large and multi-faceted; consequently, the determination of appropriate verification problems requires guidance from and consensus among experts in each of these fields.

Decision makers and code analysts should bear in mind that simulation software represents intricate numerical algorithms coupled with a complicated hardware/system-software platform. Stated another way, code users and their customers should recognize that simulation software is *not* a "physics engine" that generates instantiations of physical reality. Hence, documented,

quantitative verification analysis is a necessary component for developing code confidence and credibility.

At its core, verification of scientific simulation software both quantifies numerical errors and defines a rigorous basis for believing that quantification. Providing an error estimate for complex problems falls under the purview of solution (or calculation) verification, while providing the rigorous basis for such estimates is achieved with code verification. The overall activity of verification is the combination of both code and solution verification.

More precisely, the different kinds of verification can be defined as follows:

- Calculation or solution verification: using the demonstrated convergence properties of the code to estimate numerical errors in solving the model for a problem of interest, involving the evaluation of results of the code alone. The credibility of calculation verification is predicated on producing error estimates from a code that has passed appropriate and relevant code verification analysis.
- Code verification: comparing the results of a coded algorithm (i.e., instantiated in the software) with an analytical or exact (i.e., "closed form") solution or highly accurate solution obtained by some other means[2], for the purpose of assessing the code.

Both code verification and calculation verification focus on numerical errors. Code users practice the latter to estimate the numerical error in their simulations in order to provide a justifiable, best-estimate solution. These estimates are not guaranteed to be rigorous—they, too, are approximations. A key point is that the activity of estimating numerical errors is part of the necessary due diligence for conducting predictive simulations. In contrast, code verification is an activity executed by code developers, mathematicians, and numerical analysts. It compares the behavior of the actual, observed error from numerical simulations to the expected error behavior as derived from rigorous mathematical proofs and estimates.

A common confusion with regard to code verification is associated with software quality assurance (SQA), which is a vital, but essentially unrelated activity that comprises an important discipline in its own right. Verification typically flourishes in a culture focused on high quality, but good verification practices are neither necessary nor sufficient for good SQA practices—and vice-versa. Each area of expertise should be independently developed and supported, although the practice of each is mutually supportive.

Software verification is defined as the checking for correct functioning of the software system on a particular platform. This is often used to determine whether the software has properly implemented the defined requirements of the algorithms instantiated in the code.

Such software testing is a critical element of software development. The testing that is closest to code development centers on software engineering techniques, such as unit testing and regression testing, both of which address the correct functioning of software. These types of tests use generic success metrics that apply to almost all classes of software. In contrast, code and solution

[2] For example, for certain problems governed by specific partial differential equations (PDEs), the equation can be reduced to an ordinary differential equation (ODE), the solution of which can often be obtained very accurately, sometimes exactly. A highly accurate numerical solution of an ODE could be used as the de facto "exact" solution for purposes of code verification analysis of a PDE solver for that problem.

verification are assessment techniques for software that provides approximate solutions, with metrics specialized to the particular type of algorithm.

Sometimes, there is also confusion between code verification testing and regression testing; however, these are different testing procedures with completely different goals. Regression testing is a software engineering technique that assesses the robustness of software to frequent changes. Regression tests reduce to a (typically large) collection of relatively simple problems that are executed at a regular (typically frequent) time interval. Regression testing principally seeks to reduce the amount of software rework that is created by the introduction of mistakes in software modifications. This reduction is achieved by comparing today's code with yesterday's code via execution of the regression test suite. Thus, regression testing evaluates software stability, not mathematical correctness.

This mathematical correctness is the purview of code and calculation verification. Rephrasing our earlier definitions, calculation verification estimates numerical errors for a problem of interest based on an assumed relationship between numerical error and resolution (as measured by the discretization parameter); code verification tests whether this assumption is satisfied for a known algorithm on a problem with a known solution.

The outcome of code verification analyses should provide defensible evidence of mathematical consistency—or inconsistency—between the mathematical statements of the physics models and their discrete analogues as implemented with numerical algorithms in the simulation codes. The necessity of calculation verification must be emphasized. In the absence of satisfactory calculation verification evidence, "good agreement" of calculations with experimental data could be accidental, i.e., "the right answer for the wrong reasons."

Another activity that is confused with calculation verification is mesh sensitivity. Mesh sensitivity is the process of comparing the solutions computed on two or more grids for the express purpose of determining whether the solution is qualitatively dependent on mesh resolution. The degree of dependence is usually assessed only in a qualitative fashion, although the calculations, if done properly, can serve as the basis for calculation verification. Without the quantitative verification component of this activity, non-convergent calculations may remain unidentified.

Unlike verification analyses, validation tests whether a model is a sufficiently accurate representation of the physical processes in a particular problem. For detailed discussions on the relation of verification and validation, see the many published reviews on verification and validation (References 1, 2, 7, 9, 11, 14, 18-32, 34, 39, 41, 42, 44, and 45 and the references therein). In scientific simulations the model refers to the governing equations, which include initial conditions, boundary conditions, constitutive relations, and so forth. To simulate essentially any nontrivial physical configuration, these equations must be solved computationally, which depends on numerical algorithms, corresponding software implementations, and appropriate use of that software, including suitable assignment of model parameters. Therefore, validation entails comparisons of approximate solutions of the governing equations which are an imperfect representation of the relevant physical processes) to experimental data, which also contain inaccuracies. Quantitative comparison of experiments (having, e.g., physical and diagnostic uncertainties) with simulations (having concomitant modeling, algorithmic, and solution errors) remains a challenging and strongly problem-dependent undertaking. It is a regular experimental practice to provide error bars showing the degree of uncertainty in physical data, and solution verification can help provide estimates of the numerical error in the corresponding simulations.

Every simulation requires a number of inputs to specify the problem to be solved, including the choices among various numerical methods and how they are applied, as well as the assignment of values for parameters in specific component models (e.g., specifying material response). These values may not be known exactly or may represent some average or even ad hoc value that depends on the particular situation. The process of identifying and quantifying the effects of the uncertainty of these simulation inputs on the results of the simulation falls under the rubric of uncertainty quantification. Typically, the most common way of estimating these effects is to run a large number of simulations, in which the values of the inputs are taken from user-specified distributions, to determine the corresponding distribution of the results of interest. Unlike code verification, solution verification, or validation, uncertainty quantification deals with the variation of simulation results relative to how the problem is specified, rather than how well that problem is solved.

Code verification is the foundation upon which the other assessment techniques rest. The premise of solution verification is that the code converges at a known rate as the resolution is increased; code verification establishes that this is, in fact, the case. To compare experimental results to code results, both the experimental and numerical errors must be meaningfully quantified, so that they can be properly accounted for in the comparison. Validation relies on solution verification to provide estimates of the numerical error, a procedure that, in turn, relies on code verification. The process of uncertainty quantification accepts the model, in this case the code that produces numerical simulations, as a valid source of data. Inferences drawn about the system that the model represents are inherently limited by the accuracy of the model. Code verification ensures the model is correctly implemented and underlies solution verification and validation processes that quantify the accuracy of the model. Code verification is foundational in that without it, the conclusions of the other assessment techniques are suspect, but the converse is not true.

3. Defining a Verification Test Problem

To conduct a verification analysis, one must have (i) a clear statement of the problem with sufficient information to run a computer simulation, (ii) an explanation of how the code result and solution (i.e., the benchmark) are to be evaluated, and (iii) a description of the acceptance criteria, including how the calculation errors are to be estimated, for a specific simulation code's results on a particular problem. These concepts are adapted from the notion of a "strong sense verification benchmark," proffered by Oberkampf and Trucano [21], and are intended to enhance the value of verification analyses by reducing the ambiguity of problem statements, evaluation techniques, and their interpretation.

Here, the problem statement should include not only a mathematical description of the problem but also a discussion of the processes modeled (i.e., what does this problem test?), the initial and boundary conditions, additional numerical information (what convergence criteria are used?), the principal code features tested, and the nature of the test. This latter element is addressed in Ref. 21, with a set of different categories of benchmarks; Kamm et al. [11] provide examples of such problem statements. Given the complexity of many problems of interest, however, such problem descriptions may still not be definitive, i.e., there may remain unspecified choices in problem set-up that the code analyst must make. Nevertheless, such a description provides a starting point for setting up the problem as well as a touchstone against which one can compare descriptions of the

"identical" verification problem, run by different analysts with different simulation codes on different platforms at different times.[3] In the written description and analysis of verification problems, it is imperative that researchers describe as thoroughly as possible the complete specification and set-up of the problem up to and including the code input deck in the written report).

An explanation of the evaluation process by which results are to be evaluated is imperative for both completeness and repeatability. This evaluation process includes, e.g., in code verification what the "exact" solution is and how it is evaluated. Such an explanation may also include how system response quantities (the convergence of which is to be calculated) are evaluated from computed solution variables.

Also required in the problem statement is a description of which errors are to be estimated, along with an explanation of precisely how those errors are estimated. So that the analysis can be meaningful, acceptance criteria on the error behavior for the quantities of interest on the particular problem are also required, in the spirit of what might be found, say, in a requirements document. In the absence of specific criteria, a codified process for acceptance is necessary.

3.1. Code Verification Responsibility: Code Developers, Mathematicians, and Algorithm Engineers

Just as complex simulations cannot be proven mathematically correct, complex simulation software cannot be proven to be mathematically correct. Consequently, the accumulation of quantitative evidence remains the exclusive basis for inferring the mathematical correctness; as in calculation verification, this evidence is accumulated over time through the ongoing processes of code development and code usage. Thus, the results of code verification analyses are affected by the manner in which software is generated and the proper specification and/or execution of the verification problems. While the former is the purview of code developers and algorithm designers, responsibility for the latter falls upon those who describe the verification problem both in documentation and in actual code input.

Verification evidence emerging from code development is generated by software engineering processes applied during that development, and by the specific testing practices employed by the development team. Code usage evidence is a more nuanced and diverse body of information that emerges from a heterogeneous group of users. Testing executed under the umbrella of code development is not restricted to the verification approaches discussed here. Unlike other testing procedures applied by code developers (including, e.g., unit tests and the restricted cases applied in regression testing), verification test problems strive to be relevant to code users, as well.

Verification test suites can be implemented, managed, and applied by code developers in the same manner as regression test suites. The major differences are: (i) the development and execution of verification test suites typically requires more resources (people, computers, time);

[3] Analyses containing the possible variations mentioned here differ markedly from "code comparison" exercises, problematic aspects of which are pointed out by Trucano et al. [44]. Verification acceptance criteria should be sufficiently forgiving, however, to allow small variations associated with imperfect specification of verification problems.

(ii) the time interval of execution of a verification test suite may be different than for regression testing; and (iii) the direct methods for comparing today's regression test suite results versus yesterday's baseline should be augmented by greater human involvement in judging the quality of the verification tests, where possible. On the other hand, subjective judgment is intrinsically associated with the quality of software at some fundamental level. For example, a verification study rarely produces results that exactly produce the theoretical convergence rate; in fact, the observed rates can vary greatly. The judgment of whether the result is close enough to expectations remains largely a decision for a subject-matter expert matter. Improvement of this state of affairs is an active ongoing research area. This increased human element required to assess the execution of verification tests emphasizes that an important value of verification tests is their use in engaging the user community around a code.

3.2. Calculation Verification Responsibility: Code Users

Complex simulations cannot be proven mathematically correct, or even convergent to the "true" solution. Consequently, the accumulation of quantitative evidence remains the exclusive basis for assessing the mathematical correctness and its relation to physical correctness. The practical view is that this evidence is accumulated over time. This accumulation occurs throughout the ongoing processes of code use and the considered analysis of the computational results. Along with the results, the input to the code used to specify the calculation must be carefully examined for correctness. This process is similar to the manner in which software is examined. Code verification is the purview of code developers and algorithm designers, whose responsibilities should include providing descriptions of verification problems both in documentation and in actual code input. For those using a code to conduct analysis, their responsibility is to act mindfully regarding the quality of the code, and to keep in mind the relevance of the testing to their problems of interest. At best, they should act as advocates for quality control measures such as rigorous and—to the degree possible—for extensive verification because it supports confidence building in the calculations for which the code is ultimately to be used.

4. Verification Techniques

The workhorse technique for estimating discretization error is systematic mesh refinement (or derefinement, i.e., coarsening), while the method for estimating iterative error involves systematic changes in the stopping criteria for the iteration. A fundamental expectation for a numerical method is the systematic reduction in solution error as the characteristic length scale associated with the mesh is reduced. By the same token, iterative errors are assumed to be smaller as the stopping criterion is decreased in numerical value. For mesh refinement, in the asymptotic limit where the mesh length scale approaches zero, a correct implementation of a consistent method should approach a rate of convergence equal to that defined by numerical analysis (often obtained with the aid of the Taylor series expansion). In practice, this expectation is not always met, i.e., calculations might not be in the asymptotic range. This circumstance does not obviate the need for some estimate of the numerical error, however imprecise that estimate may be; in fact the necessity may be increased under these conditions.

To conduct analysis using this mesh refinement approach, a sequence of grids with different intrinsic mesh scales is used to compute solutions and their associated errors. The combination of errors and mesh scales can then be used to evaluate the observed rate of convergence for the numerical method used in the code. In order to estimate the convergence rate, a minimum of two grids is necessary (giving two error estimates, one for each grid). The convergence tolerance for iterative solvers can be investigated by making simple changes in the value of the stopping

criteria. Assessing iterative convergence is complicated by the fact that the level of error is also related to the mesh through a bounding relation in which the error in the solution is proportional to the condition number of the iteration matrix. Most investigations of iterative solver error only consider the impact of the stopping criteria.

Another tool used in verification is error estimation. These methods are commonly derived for the finite element method (FEM) for solving elliptic partial differential equations (PDEs). The best-known method is the Zienkiewicz-Zhu error estimator [47]. One can use the error estimate constructively to drive adaptive mesh refinement, or to produce an estimate of the error on a given mesh. While the error estimate is produced, this approach does not necessarily produce the sort of evidence basis for calculation convergence, however, because the rate of convergence is not produced in the process. Moreover these tools are more strongly predicated upon the solution being well behaved. Without solution smoothness in the asymptotic range, these tools are highly questionable for precise error estimation. We focus on techniques that can be applied without regard to whether the computations are in the asymptotic range, because it is seldom encountered in practice.

5. Choice Of Metrics

Several metrics can be used to conduct code or calculation verification. For a mathematically rigorous result, one seeks to evaluate the quantitative difference between two sets of numbers, where each set corresponds to some aspect of the solution computed in a self-consistent manner. For example, integration over the same physically defined region provides a well-defined metric with which to gauge such a difference. Generally speaking, one should employ the metric that follows naturally from the function space in which numerical analysis proofs of convergence are conducted. One often cannot apply this approach in practical circumstances.

For simple scalars (say, the value of some scalar property, e.g., temperature, at a particular location at a specified time), the absolute value of the difference between two values is the obvious choice. This notion generalizes naturally to higher dimensional cases involving the difference between (discrete) function values from the computational mesh. As examples, these values could be a time series of temperature at a particular location (the relevant computational mesh being in time) or the pressure field over a specified, fixed three-dimensional volume at a specified time (the relevant computational mesh being in space). In such cases, one often uses the familiar "*p*-norm" of functional and numerical analysis. The *p*-norm of the function *g* is given by:

$$\|g\|_p \equiv \left(\int_a^b |g(x)|^p \, dx \right)^{1/p} \tag{1}$$

For example, for finite volume methods applied to discontinuous functions, the use of the 1-norm is recommended, while, say, properties of inherently smooth functions are most appropriately measured in the energy or 2-norm. It is can be enlightening to evaluate several norms, e.g., 1-, 2-, and ∞-norms, where, following from the equation above:

$$\|g\|_\infty \equiv \max_{x \in [a,b]} |g(x)| \tag{2}$$

In the following, we use the double-bar notation "||" without a subscript to denote any appropriate norm. Ideally, the practical results should be connected to rigorous mathematical analysis, but this is often intractable.

In practice, the metrics tested for code and solution verification are generally different. For code verification, numerical analysis of the algorithm defines the metric and the expected convergence rate, as well as the assumptions under which convergence is expected. For most spatial and temporal discretizations, the metric is an appropriate norm of the error of the solution; that is, the error of a field variable integrated over the computational domain. The numerical analysis specifies how the error is defined. Depending on the properties of the numerical method and the solution, the theoretical convergence rate may be expected to apply for point values of the solution as well.

For calculation verification, the analyst or code user chooses metrics that are important to the decision maker; these metrics are sometimes called Figures of Merit (FOMs), Quantities of Interest (QOIs), or System Response Quantities (SRQs). These metrics provide the "answer" to the question the analyst is trying to address; determining their values is the reason for running the set of simulations. These metrics may be full-field solutions, but more often are values or functionals of the solution at a particular location or time. The convergence of such metrics will depend directly on the mathematical relationship between the FOM and the numerical field solution provided by the simulation. In practice, the quality of FOM error estimates often suffers because the rigorous connection between the numerical field solution and each FOM can be difficult to establish. For example, while very popular as FOMs, point values of the solution are often much more sensitive than integrated quantities and, when reliable error estimates can be obtained for them, the estimated errors are comparatively greater.

Generally, convergence can be proven only under extremely limited circumstances. For example, a discontinuous solution of a hyperbolic system of equations can be proven to converge in the L1 norm of the solution [17]. Other functional relations will not have a precisely defined convergence rate or character. More broadly speaking, the convergence depends upon the smoothness of the solution: the smoother the solution (i.e., having higher derivatives that are well defined), then the better the chances for convergence in higher norms. The max norm (L_∞) is generally the most difficult norm in which to obtain (or expect) convergence. The max norm is the largest error in the solution and it is defined at a single location. FOMs involving only a single point in the solution often mimic the behavior of the L_∞ norm, thus carrying the same expectations for convergence.

6. Observed Convergence Behavior

In this section we will explore the basic categories and purposes of convergence behavior. This includes the ideal and non-ideal behaviors and the implications of these behaviors on error estimation.

6.1. Ideal Asymptotic Convergence Analysis

In this section, we examine the case of ideal asymptotic convergence analysis. The axiomatic premise of asymptotic convergence analysis is that the computed difference between the reference and computed solutions can be expanded in a series based on some measure of the discretization of the underlying equations. Taking the spatial mesh as the obvious example, the ansatz for the error in a 1-D simulation is taken to be:

$$\left\| g^f - g^c \right\| = A_0 + A_1 (\Delta x)^p + o\left((\Delta x)^p \right) \tag{3}$$

In this relation, g^f is the reference solution, which for calculation verification is computed on a refined mesh, g^c is the computed solution, Δx is some measure of the mesh-cell size, A_0 is the zero-th order error, A_1 is the first-order error, and the notation "$o\left((\Delta x)^p\right)$" denotes terms that approach zero faster than $(\Delta x)^p$ as $\Delta x \to 0^+$. For consistent numerical solutions, A_0 should be identically zero; we assume this to be the case in the following discussion. For a consistent solution, the exponent α of Δx is the convergence rate: p =1 implies first-order convergence, p =2 implies second order convergence, and so forth.

Assume that the calculation has been run on a coarse mesh (subscript c), and a very course mesh (subscript vc) characterized by Δx_c (Δx_{vc}), which we hereafter also denote as Δx. The error ansatz implies:

$$\left\| g^f - g^c \right\| = A_0 + A_1 \left(\Delta x\right)^p + \ldots \tag{4}$$

Let us further assume that we have computational results on a fine mesh Δx_f (subscript f), where $0 < \Delta x_f < \Delta x_c$ with $\Delta x_c / \Delta x_f \equiv \sigma > 1$. In this case, the error ansatz implies:

$$\left\| g^f - g^c \right\| = \sigma^{-p} A_1 \left(\Delta x\right)^p + \ldots \tag{5}$$

Manipulation of these two equations leads to the following explicit expressions for the quantities p and A_1:

$$p = \left(\log \left\| g^f - g^c \right\| - \log \left\| g^c - g^{vc} \right\| \right) \Big/ \log \sigma \tag{6}$$

$$A_1 = \left\| g^f - g^c \right\| \Big/ \left(\Delta x\right)^p \tag{7}$$

These two equalities are the workhorse relations that provide a direct approach to convergence analysis as a means to evaluate the order of accuracy for code verification.

For QOIs or FOMs the above development can be utilized without resorting to error norms. The quantity, G, is defined without the use of a norm with the following related error model:

$$\tilde{G} = G^k + A_1 \left(\Delta x\right)^p + \ldots \tag{8}$$

with the remainder of the development proceeding as above, provided the approach toward $\tilde{G}$, the mesh converged solution, is monotonic. In the case where a solution is not monotonically approached, the above error model can still be utilized as long as the error in absolute terms is diminishing monotonically. This recommendation is in some clear opposition to the existing literature although error norms themselves are typically positive definite quantities. The more general cases are described next.

6.2. Non-Ideal Asymptotic Convergence Analysis

The term "non-ideal" refers to situations for which the error does not monotonically decrease with increasing mesh refinement in the simulation. The computed solutions may diverge under mesh refinement, in either a monotonic or non-monotonic fashion, or the computed values may

have no overarching characteristics beyond simply being bounded. Some researchers contend that one cannot make further statements about calculations that do not exhibit the ideal asymptotically convergent behavior described in Section 6.1. Insofar as such non-ideal behavior is often seen in practical engineering simulations, however, this section contains a discussion of the analysis of such non-ideal cases.

Several authors have offered schemes by which the behavior of computed solutions on three or more grids can be categorized. These approaches seek to automatically classify the convergence behavior into one of the categories shown in Figure 1. The salient parameters in these analyses are the theoretical convergence rate of the underlying numerical scheme, p_{th}, and the observed convergence rate, based on Richardson Extrapolation (see, e.g., Ref. 22), corresponding to the actual simulation results, p_{RE}. Xing and Stern [46], who build upon earlier work of Eça and Hoeckstra [4], studied this approach. The following discussion is based on these publications; for further background, see the earlier works, for example, Stern et al. [41].

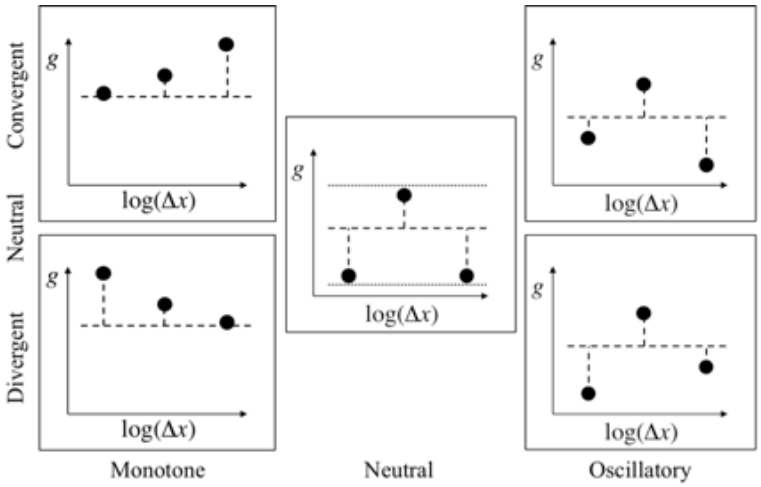


Figure 1. Schematic representation of possible behavior for computed solutions under mesh refinement. The individual plots are notional graphs of computed solution, g, versus the log of the mesh spacing of the calculation, Δx; the dashed lines are intended only to guide the eye. The vertical axis in the overall figure goes from divergent to convergent, and the horizontal axis goes from monotone to oscillatory. The ideal case of monotone convergence under mesh refinement is shown in the upper left of the figure, and the limiting case of neutral behavior is in the center.

For the estimation of convergence properties of scalar quantities, Eça and Hoeckstra [4] approach the problem of estimating the observed convergence rate as a least-squares solution to the governing error ansatz equation:

$$\left|G^f - G^c\right| = A_1 \left(\Delta x\right)^p + \ldots \tag{9}$$

Let p denote the convergence rate corresponding to the least-squares solution for this equation based on the *signed* differences of results computed with adjacent grid spacings, i.e., where both positive and negative values are allowed on the left-hand side of Eq. 9. Similarly, let p^* represent the convergence rate corresponding to the *absolute* differences of results computed with adjacent grid spacings, i.e., where absolute values of the left-hand side of Eq. 9 are used. Then, the behavior of the set of computations is determined from the following algorithm:

`If` $p > 0$ `then` monotonic convergence

`Else if` $p < 0$ `then` monotonic divergence

`Else if` $p^* < 0$ `then` oscillatory divergence

`Else` oscillatory convergence

A second way to characterize the nature of the convergence is to examine the ratio of signed error in the solution from one mesh refinement level to the next:

$$R = \left(G^f - G^c\right) \Big/ \left(G^c - G^{vc}\right) = \Delta^{cf} / \Delta^{cvc}, \tag{10}$$

`If` $R < 1$ `then` monotonic convergence

`Else if` $R > 1$ `then` monotonic divergence

`Else if` $R <$ -1 `then` oscillatory divergence

`Else` (-1 < R < 0) oscillatory convergence

6.3. Estimation of Associated Numerical Uncertainty

Once the nature of the solution has been properly categorized, the numerical uncertainty can then be estimated as part of the overall uncertainty estimate.[4] The Grid Convergence Index (GCI) of Roach (Refs. 26 and 27) is perhaps the original attempt to codify the numerical uncertainty associated with inferred convergence parameters. Roache [26] claims that there is evidence for the numerical uncertainty based on the GCI method (with a safety factor of 1.25) to achieve a 95% confidence level. This approach was extended to the Correction Factor (CF) method of Stern et al. [41]. Xing and Stern [46], however, take issue with both of these approaches, stating, "…there is no statistical evidence for what confidence level the GCI and CF methods an actually achieve" and, more specifically, that their analyses "…suggest that the use of the GCI_1 method is closer to a 68% than a 95% confidence level." As we describe below, Xing and Stern come to a different conclusion regarding an approach that does meet the 95% confidence level empirically.

Eça and Hoeckstra [4] propose heuristics by which to estimate the numerical uncertainty associated with the fundamental behavior of a set of computed results. These suggestions appear to be based on the assumption that the underlying numerical scheme has a theoretical convergence rate of two; however, for many multiphysics (and some single-physics) problems, the theoretical convergence rate is unity, for which the specific prescription of Eça and Hoeckstra should be modified. It is worth noting here that the convergence rate is both a function of the scheme employed and the nature of the solution sought itself. For example, a second-order method applied to a problem with a discontinuous solution cannot product a second-order convergent result. Hence the expected theoretical convergence rate is to be considered a function of both the method used and the solution sought.

[4] The proceedings of the 1st, 2nd, and 3rd Workshops on Computational Fluid Dynamics (CFD) Uncertainty Analysis [5] provide an interesting reference on many aspects of uncertainty analysis for CFD.

Xing and Stern [46] take a different approach. To evaluate the numerical uncertainty associated with these solution verification estimates, they performed a statistical analysis of 25 sets of computational data, covering a range of fluid, thermal, and structural simulations, to arrive at various parameters for their estimations of simulation uncertainty. The parameters obtained by Xing and Stern provide computational uncertainty estimates that demonstrably satisfy the 95% confidence level for the data sets upon which that analysis is based. They suggest that the formula below provides a safety factor with empirical, statistical support. We suggest following this approach whenever the grid sequence provides a convergent sequence.

$$U_{num} = F_S\left|\delta_p\right| = \begin{cases} \left(2.45 - 0.85P\right)\left|\delta_p\right|, \text{if } 0 < P \le 1 \\ \left(16.4 - 14.8P\right)\left|\delta_p\right|, \text{if } P > 1 \end{cases} \tag{11}$$

where $P = \alpha_{RE}/\alpha_{th}$ defines whether the solution is asymptotic in observed nature. The numerical error magnitude comes from the Richardson extrapolation toward the monotonically mesh converged solutions as:

$$\delta_p = \frac{G^f - G^c}{\sigma^p - 1} \tag{12}$$

or the related error estimate for monotonically decreasing error as:

$$\delta_p = \frac{\left|G^f - G^c\right|}{\sigma^p - 1} \tag{13}$$

In the case where the solution is not convergent, the numerical uncertainty should nonetheless be estimated, however rough those estimates may be. It is the authors' experience that users of codes will generally move forward with calculations and—without guidance to the contrary—may offer *no* numerical uncertainty estimates whatsoever. We maintain that this practice is potentially more dangerous than providing a weakly justified estimate. We offer the important caveat that this bound is not rigorously justified; instead, it is perhaps appropriately viewed as a heuristic estimate that can be produced given limited information. The simplest approach is to examine the range of solutions produced and multiply this quantity by a generous safety factor:

$$U_{num} = 3\left(\max G - \min G\right) \tag{14}$$

The safety factor, set to 3 in Eq. 14, might assume different values in different computational science applications. This heuristic approach is similar to that advocated by Eça and Hoekstra [4].

An alternative approach would involve postulating that the numerical error has a portion that vanishes under mesh refinement and a portion that is constant:

$$\left|G^f - G^c\right| = \tfrac{1}{3}U_{num} + A_1 \Delta x \tag{15}$$

the uncertainty is then the constant portion, which for a divergent calculation will be larger than any of the values computed.

The errors associated with iterative (linear and nonlinear) solutions of systems of equations can be approached similarly. We note that many investigations have sought to merely render these errors too small to be significant. While such an approach can work, it may not be generally possible in practice. A more complete approach is to quantify these errors in terms of the controllable numerical parameters, usually the stopping criteria for such iterations.

First, consider the dependence of the error in a solution on the stopping criteria:

$$\frac{\|r\|_k}{\|r_0\|} < \text{tol} \tag{16}$$

where $r_k = Ax_k - b$ is the residual, A is the linear system, b is the right-hand side and x_k is the solution vector after the kth iteration. The error in the solution is then bounded as:

$$\|\tilde{x} - x_k\| \sim \kappa(A)\frac{\|r\|_k}{\|r_0\|} \tag{17}$$

with $\kappa(A)$ being the condition number of A defined by the product of the operator norm of A^{-1} with the operator norm of A, which reduces to a ratio of maximum to minimum eigenvalues in the case of matrices [12]. The condition number of a matrix defined by the discretization of a PDE is related to the mesh spacing. For example, an elliptic PDE will usually have a matrix representation that scales as $1/(\Delta x)^2$.

Returning to the estimation of error, the simplest numerical parameter to control is the tolerance of the linear solution. We can use the same approach as for mesh spacing and propose that the normed error in the solution be:

$$\left\|g^f - g^c\right\| = B_1(tol)^\beta \tag{18}$$

The remaining verification procedures discussed above can be applied in this case, although we do not have sharp expectations about the power-law dependence of the solution on the tolerance. We propose the use of a simple uncertainty estimate of:

$$U_{tol} = 1.25\left|\delta_\beta\right| \tag{19}$$

where

$$\delta_\beta = \frac{\left\|g^f - g^c\right\|}{q^\beta - 1}; q = \frac{(tol)^c}{(tol)^f} \tag{20}$$

The overall uncertainty from numerical approximation uses the root means square (RMS) of these two estimates as:

$$U_{approx} = \sqrt{U_{num}^2 + U_{tol}^2} \tag{21}$$

6.4. Verification Subtleties

There are several subtle but important—and, in some cases, open—issues associated with the estimation of the quantities mentioned above. While the following topics may be considered by some to be arcane, they should be kept in mind by those devising and conducting verification analyses, as well as by code analysts.

6.4.1. Nondimensionalization

The above discussion of the error ansatz and the associated convergence parameters contain no assumptions regarding the dimensions of the associated variables. Consequently, parameters in the resulting scaling relations, e.g., Eqs. 3 and 4, may have inconsistent units. One way to avoid this issue is to nondimensionalize all quantities prior to conducting such an analysis. For example, one can choose representative quantities G and X with which to nondimensionalize the computed quantity g and the representative mesh scale Δx:

$$g' = g/G \quad \text{and} \quad \Delta x' = \Delta x/X \tag{22}$$

The nondimensional error ansatz is posited to be:

$$\left\| g'^{f} - g'^{c} \right\| = A_1 (\Delta x)^p + \ldots \tag{23}$$

where all terms in this equation are now dimensionless. In this case, care must be taken to nondimensionalize consistently throughout the analysis, and to properly dimensionalize results, e.g., if one were to use this relation to estimate errors at another mesh size.

- Dimension: For problems in multiple space dimensions, e.g., 2-D Cartesian (x,y), the spatial convergence analysis described above can be assumed to carry over directly, such that, e.g., the ansatz of Eq. 3 follows identically. That is, one typically does not assume separate convergence rates in separate coordinates. This is a reasonable assumption in almost all cases; the exception is time-convergence, since the time-integration scheme for a PDE may be of a different order than the spatial integrator. For a more thorough discussion and examples of combined space-time convergence, see Refs. 8 and 43.
- Frame: Spatial convergence analysis is idealized to refer to a fixed mesh, i.e., the Eulerian frame. Approaches have been taken to extend convergence analysis simplistically to the Lagrangian frame (e.g., Kamm, et al. [10]). More sophisticated approaches, however, are needed. For example, since the fundamental Lagrangian equations are discretized with respect to *mass* and not *space*, an error ansatz analogous to Eq. 3, with the characteristic mesh discretization parameter Δx, replaced by a characteristic mass discretization parameter Δm would be more faithful to the underlying formulation.
- Non-uniform Meshes: The intention behind the expression "Δx" in Eq. 3 is that it is a meaningful measure of the characteristic length-scale of mesh cells of the discretized equations. If either adaptive mesh refinement (AMR) or an arbitrary Lagrangian-Eulerian (ALE) approach is used, however, such a quantity—if one exists—is likely to change during the course of a calculation. Again, straightforward approaches for non-uniform and AMR meshes have been examined (e.g., Li, et al. [15]), but these are topics of open research.

- Norm Evaluation: The expression for the norm in Eq. 1 is appropriate, e.g., for Cartesian geometries. This term must be appropriately modified for non-Cartesian geometries. For example, for 1-D spherically symmetric calculations, the integral of the norm is properly expressed as:

$$\|g\|_p \equiv \left(\int_a^b |g(x)|^p \, dx \right)^{1/p} = \left(\int_a^b |g(x)|^p \, 4\pi r^2 \, dr \right)^{1/p} . \tag{24}$$

In general, when evaluating the norm one must be mindful of the domain of the integral as well as any symmetries associated with the problem.

- Norm Evaluation and Interpolation on Different Meshes: The expression for the convergence rate α in the calculation verification (Eq. 6) implies a direct comparison of computed solutions on three different meshes. The analogous expression (Eq. 6) for code verification requires an indirect comparison of computed solutions on two different meshes. To evaluate the differences between two calculations, a common mesh is required. This begs the question: should one extrapolate ("restrict") fine-mesh values to the coarse mesh, or interpolate ("prolong") coarse-mesh values onto the fine mesh? Margolin and Shashkov [16] provide a rationale for the former: "…by moving each of the simulation results to the coarsest mesh, we average out the smaller scales and eliminate them as a source of error in studying convergence, thus isolating the discretization error." The detailed manner by which one should move solutions between different meshes remains an open research area. Particular attention should be paid to accurately interpolating solutions near discontinuities.
- Tracers as Figures of Merit: A tracer is an imaginary particle that does not influence the simulations; it is effectively a point in the computational domain at which the values of the field variables are recorded over time and is used as a diagnostic. Most often tracers are Lagrangian (i.e., they move with the material) or Eulerian (i.e., they are associated with a fixed spatial location.) As FOMs, tracers present several challenges: they record point values, and, as mentioned earlier, point values may not converge at the same rate as the error integrated over the computational domain; they may be more sensitive than integrated quantities; interpolating field variables to tracer locations adds interpolation error (a form of discretization error) to the values recorded at the tracer; and, for tracers that move with respect to the mesh, the location of the tracer is itself subject to discretization error. For these reasons, FOMs associated with tracers can have relatively large numerical uncertainties. Research effort to address these uncertainties is warranted, however, as tracers remain a common type of FOM.

6.4.2. Example —Direct Numerical Simulation Of Turbulence

In this section, we use data from turbulent flow simulations to illustrate a sample calculation verification analysis focused on estimating the numerical uncertainty associated with the estimated discretization error. This analysis was conducted using a series of calculations carried out on a sequence of uniformly refined meshes.

The foundation of this approach is the standard error ansatz utilized in verification, as discussed previously. Here, the quantity δ denotes the extrapolated difference in the solution from the fine grid to the desired mesh converged solution assuming the convergence rate is constant.

We first consider some idealized examples that illustrate possible outcomes for mesh refinement studies. The figures below provide a schematic representation of experiment and calculation comparison with increasing levels of mesh refinement (Figures 2–6). These figures are intended to give a notional representation of current practice and how it can advance toward a desired practice. The figures begin with generally accepted current practice in fluid dynamics and show how, with modest additional effort, one can achieve improved analysis results including calculation verification. They do not, however, include a detailed uncertainty assessment, which is also needed.

Figure 2 depicts the fundamental comparison of experimental data (the symbols and associated vertical error bars) with computational results (the continuous line). The graphical comparison of the values in this example provides some qualitative evidence that a single calculation and the experiment are approximately congruent.

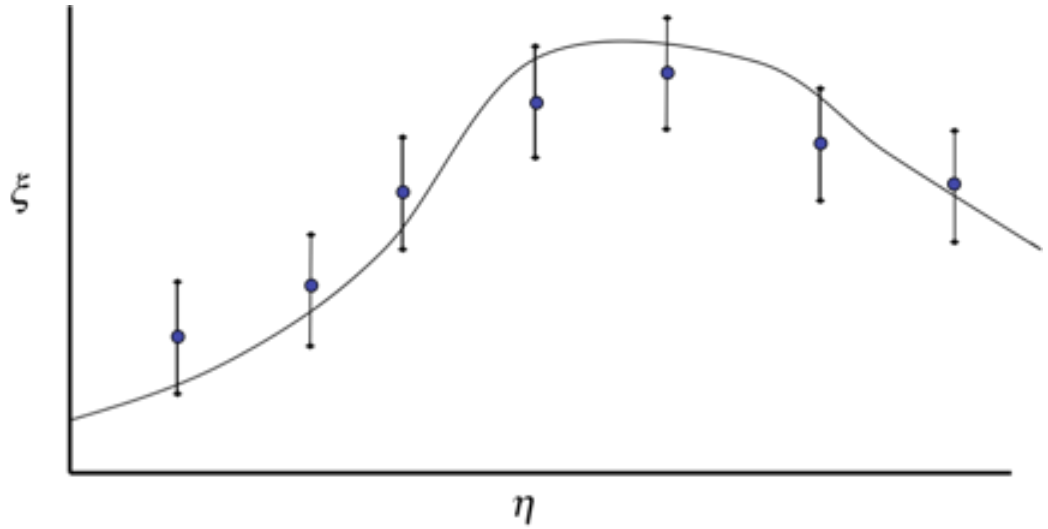


Figure 2. Typical comparison of experiment and calculation. The line denotes the calculation and the symbols denote the data with error bars.

A useful next step is illustrated in Figure 3, which adds the results of a comparable *but more coarsely discretized* calculation (dashed line) to the basic experiment-calculation graphic of Figure 2. The evidence provided by such a plot can provide qualitative hints (i.e., mesh sensitivity) as to whether the computed results might approach or diverge from the experimental data under mesh refinement. Also, the resource requirements of the additional calculation (on a more coarsely discretized mesh) are reduced relative to the original calculation. Such graphical comparisons do not, however, constitute any form of verification of the calculation. The procedures we describe in this document provide the means to produce such a verification analysis.

A valuable further step is illustrated in Figure 4, which adds to Figure 3 the results of an even more coarsely discretized calculation (dotted line), with a mesh resolution ratio equal to that of the results in Fig. 3. In the situation depicted, the difference between any two pairs of nearby calculations remains approximately constant under a uniform refinement. Such results imply that the calculations are *not* converging. This is an example of possible monotone, divergent behavior.

Independent of the lack of convergence of the calculations under mesh refinement, the evidence provided in this notional plot provides qualitative evidence suggesting that the computed results diverge from the experimental data under mesh refinement. And, again, the resource requirements

for this third calculation are reduced relative to the previous two calculations. This plot alone does not constitute verification; however, with this third calculation, the necessary information is now available to evaluate quantitative error and convergence rate estimates.

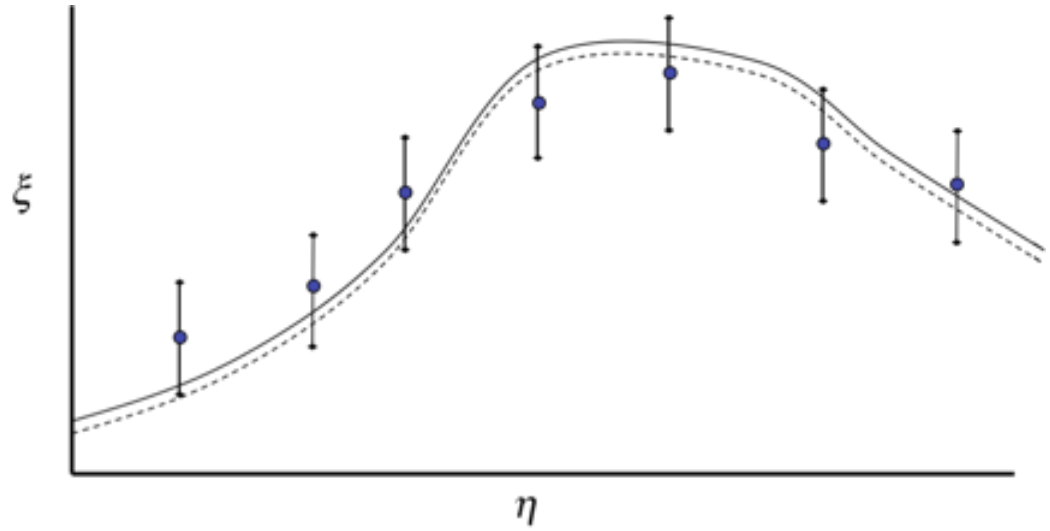


Figure 3. The usual mesh sensitivity representation. The dashed line is a more coarsely discretized calculation, and the closeness of the lines is meant to reassure the observer of the accuracy of the calculation. This does *not* constitute verification of the calculation. This is an example of how mesh sensitivity is often presented; that is the two calculations appear to be close, but no convergence is quantitatively assessed.

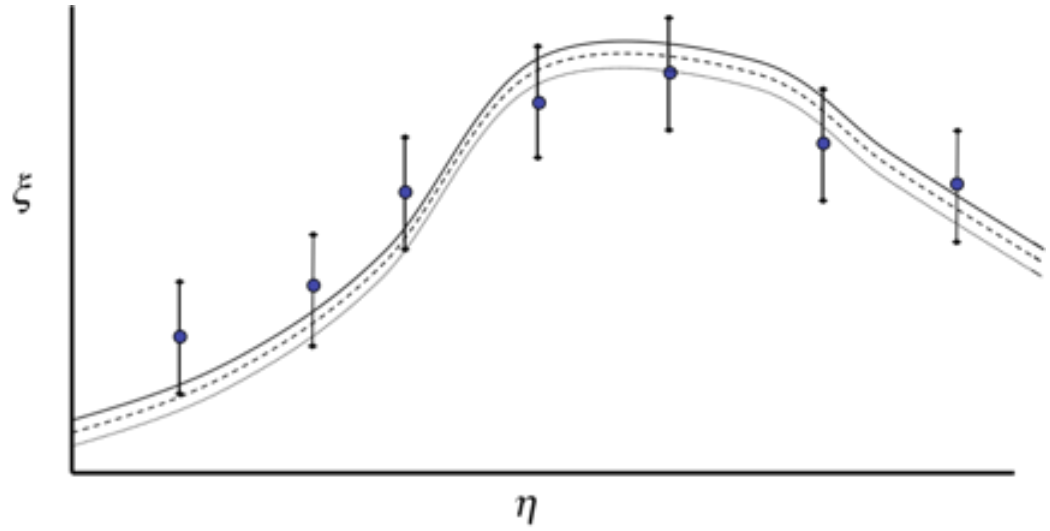


Figure 4. The finest grid used is denoted by the solid line (which is above the dotted line), and the coarsest grid result is given by the dotted line. In the case depicted, the error in the calculations does not shrink as the grid is refined, an indication of non-convergent behavior. The danger of mesh sensitivity is illustrated here with this non-convergent sequence of calculations.

A different possible outcome of a third calculation is shown in Figure 5. The most coarsely discretized calculation (dotted line), again with a mesh resolution ratio equal to that of the results in Figure 3, now suggests that the difference between any two pairs of nearby calculations decreases under uniform refinement. That is, the difference between the dotted and dashed line looks greater than the difference between the dashed and solid lines. Thus, this plot provides

qualitative evidence that the sequence of computations appear to be converging under mesh refinement. This is an example of possible monotone, convergent behavior. Again, this plot alone does not constitute verification, but it contains the necessary information to evaluate quantitative error and convergence rate estimates.

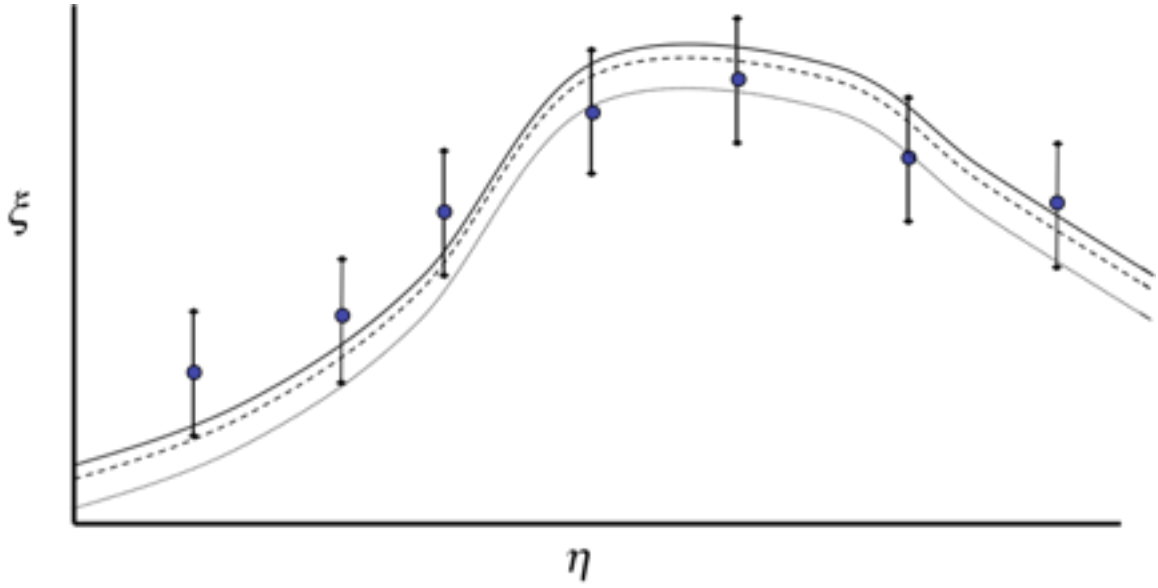


Figure 5. A first-order convergent example is shown in schematic form. The finest grid is the black solid line and the dashed lines correspond to two coarser resolutions. For a mesh-doubling procedure in which a factor of two refinement is used, the level of error in the calculation with the refined mesh spacing also halves. For example, a second order result would produce an error level that is reduced by a factor of four as the grid is refined.

Figure 6 includes the graphical results of such a quantitative analysis. The red line in this diagram represents the extrapolated solution, which is computed based on the convergence rate estimate obtained from the sequence of three calculations. This extrapolated solution can be compared against the experimental data to provide an estimate of the error between experiment and simulation.

We now turn to a quantitative analysis of such a situation. This analysis is based on the scalar values calculated for the turbulent mixing simulations of Donzis [3]. In the discussion that follows we apply calculation verification to the simulations defined in Table 1, below. The values in this table relate to characteristics of the different simulations. Note that, per the value of grid points (N^3) reported in the second row, there is a uniform mesh refinement ratio between computed results. Also, the value in the fourth row, $k_{max}\ \eta$, the product of the maximum wavenumber resolvable on the mesh (k_{max}) and the Kolmogorov scale (η), corresponds to the inverse of the mesh size, $1/\Delta x$, so that smaller values of $k_{max}\ \eta$ represent coarser mesh resolution.

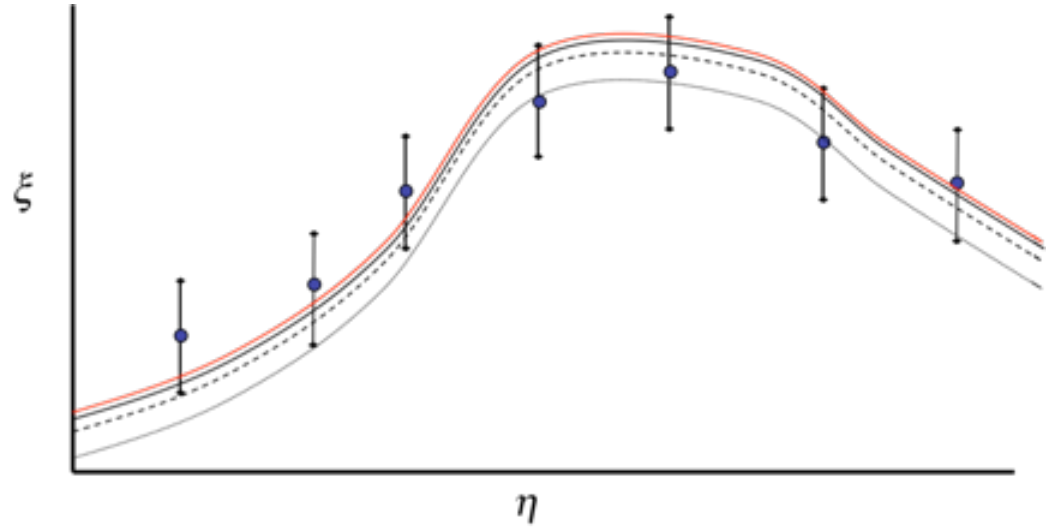


Figure 6. A mesh sequence and extrapolated solution is shown. The solid red line represents the solution extrapolated from the finest grid solution. The black solid line show the finest grid solution and the dashed and grey lines correspond to two coarser resolutions.

Table 2 contains results of the calculations and simulations for various ensemble-averaged moments of the dissipation ε' and enstrophy Ω'. The values contained in this table are examples of statistics that one would expect to be well behaved in a direct numerical simulation (DNS). For the $R_\lambda \approx 140$ case, there are four calculations at four different mesh resolutions, with each pair of mesh spacings differing by a factor of two, while for the $R_\lambda \approx 240$ case, there are computed results at three different mesh resolutions. Recall that smaller values of $k_{max}\eta$ correspond to coarser mesh resolution. We analyze the results in Table 2 using Eqs. 10–14.

Table 1. Table 3.1 from Donzis [3], containing several computed quantities for turbulent mixing simulations conducted on four different meshes. The row starting with N3 contains the number of mesh points used in the corresponding calculations.

Table 3.1: DNS parameters: Taylor-microscale Reynolds number $R_\lambda = u'\lambda/\nu$, number of grid points N^3, viscosity ν, resolution measured by $k_{max}\eta$ and $\Delta x/\eta$, number of independent realizations N_r and length of the simulation T normalized by eddy turnover time $T_E = L/u'$.

R_λ	140	140	140	140	240	240	240
N^3	256^3	512^3	1024^3	2048^3	512^3	1024^3	2048^3
ν	0.0028	0.0028	0.0028	0.0028	0.0011	0.0011	0.0011
$k_{max}\eta$	1.4	2.8	5.7	11.1	1.4	2.8	5.4
$\Delta x/\eta$	2.10	1.05	0.52	0.27	2.08	1.04	0.55
N_r	11	16	18	11	13	12	14
T/T_E	10.0	7.2	8.5	6.0	9.4	5.4	5.4

Table 2. Table 3.2 from Donzis [3], containing several computed quantities for turbulent mixing simulations conducted on four different meshes for two cases, Rl≈140 (top) and Rl≈240 (bottom).

Table 3.2: Ensemble averaged moments of dissipation and enstrophy at $R_\lambda \approx 140$ (top) and 240 (bottom) with 90% confidence intervals.

$R_\lambda \approx 140$

$k_{max}\eta$	1.4	2.8	5.7	11.1
$\langle(\epsilon')^2\rangle$	2.53 ± 0.04	2.85 ± 0.07	2.77 ± 0.06	2.82 ± 0.08
$\langle(\epsilon')^3\rangle$	14.1 ± 0.6	21.5 ± 1.6	19.9 ± 1.4	20.7 ± 2.1
$\langle(\epsilon')^4\rangle$	153 ± 14	388 ± 58	341 ± 48	364 ± 81
$\langle\epsilon^4\rangle/\langle\epsilon^2\rangle^2$	23.9	47.8	44.5	45.8
$\langle(\Omega')^2\rangle$	4.52 ± 0.09	5.19 ± 0.18	5.07 ± 0.19	5.20 ± 0.23
$\langle(\Omega')^3\rangle$	63.0 ± 3.1	100.0 ± 9.3	94.2 ± 9.9	97.6 ± 13.1
$\langle(\Omega')^4\rangle$	2022 ± 179	5315 ± 989	4920 ± 965	4751 ± 1200
$\langle\Omega^4\rangle/\langle\Omega^2\rangle^2$	99.2	197.1	191.3	175.9

$R_\lambda \approx 240$

$k_{max}\eta$	1.4	2.8	5.4
$\langle(\epsilon')^2\rangle$	3.07 ± 0.05	3.17 ± 0.07	3.15 ± 0.06
$\langle(\epsilon')^3\rangle$	25.3 ± 1.3	29.1 ± 1.8	28.8 ± 1.7
$\langle(\epsilon')^4\rangle$	488 ± 53	696 ± 83	697 ± 89
$\langle\epsilon^4\rangle/\langle\epsilon^2\rangle^2$	51.9	69.3	70.4
$\langle(\Omega')^2\rangle$	5.81 ± 0.13	5.99 ± 0.18	5.93 ± 0.12
$\langle(\Omega')^3\rangle$	133 ± 8	150 ± 14	142 ± 9
$\langle(\Omega')^4\rangle$	8364 ± 1017	11222 ± 1869	10211 ± 1503
$\langle\Omega^4\rangle/\langle\Omega^2\rangle^2$	247.7	312.8	290.6

7. Data Reduction Using Calculation Verification

The results will be displayed below for some representative quantities taken from these tables. For the R_λ≈140 case, the first mesh sequencing looks positive, but the last mesh casts doubt on these inferences, as the convergence rates drop in every case. Consequently, the numerical uncertainty grows larger and, in some cases, becomes undetermined due to divergence. In the case of these calculations the nature of the flow including the formation of (quasi-)singular

structures would indicate the expected convergence rate to be first-order irrespective of the numerical method used.

Before presenting results, some comments are in order. First, all the differences shown use the absolute value of the difference in the solutions; hence, the numerical uncertainty is plus or minus the indicated value. If the solutions are monotonically approaching a value, then the numerical uncertainty would be one sided, for example $\left\langle (\varepsilon')^2 \right\rangle$, and the appropriate error bar would be asymmetric. The fourth grid is given the subscript of "*v*" to denote "very fine." All mesh refinement ratios, *r*, are two. The large change in the convergence rate may indicate that the solutions might not be converged even at the finest mesh used. A situation that would provide a higher degree of confidence in the solution would show similar convergence rates for quantities across the full sequence of meshes used. Divergence indicates that these values may be unreliable from the perspective of numerical simulation.

The results of our analysis of the $R_\lambda \approx 140$ case are given in Table 3. In this table: Δ_{cm} denotes the difference between the coarse and medium solutions (see Eq. 10); α_{cf} is the inferred convergence rate using the coarse-medium-fine results (see Eq. 10); and U_{cf} gives an estimate of the associated numerical uncertainty on the finest of the corresponding meshes (see Eqs. 11–14). The analogous values subscripted *mf*, *fv*, and *mv* correspond to calculated results using the medium, fine, and very fine computational grids.

We evaluate the inferred uncertainty results by comparing the computed values of numerical uncertainty *U* (in the last two columns of Table 3 and last column of Table 4) with the corresponding statistical (i.e., ensemble) uncertainty values in the corresponding quantities in Table 2 (i.e., the values immediately after the "±" sign). Inspection of these values shows that on the coarser sequence of grids, the numerical errors are larger in magnitude than the statistical uncertainty (with similar confidence bands of 90% for DNS and 95% for the numerical uncertainty estimators). For example, for the dissipation quantity given by $<(\varepsilon')^2>$ in the $R_\lambda \approx 140$ case, the numerical uncertainty associated with the discretization error on the coarse-medium-fine grid results, 0.48, is greater than the corresponding ensemble-averaged statistical uncertainty, 0.06. To repeat, the estimated numerical uncertainty associated with the discretization error is greater than—but of the same order-of-magnitude as—the ensemble-averaged statistical uncertainty.

The DNS is computed using a fourth-order Runge-Kutta time integrator and employs spectral differencing in space, which might lead to the conclusion that a fourth-order convergence should be expected. Moreover, using a fourth order expected convergence rate, the error estimates yield errors not supported by the change in solution to the very fine grid. This lends greater credence to our expected first-order convergence rate. On the finer grid sequence, the numerical errors are much smaller by comparison, now comparable to the statistical errors measured.

Table 3. The results for the $R_\lambda \approx 140$ calculations.

Quantity	Δ_{cm}	Δ_{mf}	Δ_{fv}	a_{cf}	a_{mv}	U_{cf}	U_{mv}
$\left\langle (\varepsilon')^2 \right\rangle$	0.32	0.08	0.05	2.00	0.68	0.48	0.16
$\left\langle (\varepsilon')^3 \right\rangle$	7.40	1.60	0.80	2.21	1.00	9.46	1.28
$\left\langle (\varepsilon')^4 \right\rangle$	235.00	47.00	23.00	2.32	1.03	273.54	46.48
$\left\langle \varepsilon^4 \right\rangle / \left\langle \varepsilon^2 \right\rangle^2$	23.90	3.30	1.30	2.86	1.34	16.94	6.12
$\left\langle (\Omega)^2 \right\rangle$ *	0.67	0.12	0.13	2.48	-0.12	0.68	0.39
$\left\langle (\Omega)^3 \right\rangle$	37.00	5.80	3.40	2.67	0.77	31.32	8.65
$\left\langle (\Omega)^4 \right\rangle$	3293.00	395.00	169.00	3.06	1.22	1904.57	668.18
$\left\langle \Omega^4 \right\rangle / \left\langle \Omega^2 \right\rangle^2$ *	97.90	5.80	15.40	4.08	-1.41	19.02	63.60

*denotes a divergent calculation.

The results of our analysis of the $R_\lambda \approx 240$ case are given in Table 4. In this table, all the quantities are converging at a higher than expected rate. Given the results at the lower Reynolds number, we expect the uncertainty estimates to decrease as the mesh is refined. For most quantities given, the numerical error is larger than the statistical variability measured. For example, for the enstrophy, $\langle (\Omega')^2 \rangle$, in the $R_\lambda \approx 240$ case, the numerical uncertainty associated with the discretization error on the coarse-medium-fine grid results, 0.34, is greater than the corresponding ensemble-averaged statistical uncertainty, 0.12.

Table 4. The results for the $R_\lambda \approx 240$ calculations.

Quantity	Δ_{cm}	Δ_{mf}	a_{cf}	U_{cf}
$\left\langle (\varepsilon')^2 \right\rangle$	0.10	0.02	2.32	0.12
$\left\langle (\varepsilon')^3 \right\rangle$	3.80	0.30	3.66	1.16
$\left\langle (\varepsilon')^4 \right\rangle$	208.00	1.00	7.70	0.54
$\left\langle \varepsilon^4 \right\rangle / \left\langle \varepsilon^2 \right\rangle^2$	17.40	1.10	3.98	3.75
$\left\langle (\Omega)^2 \right\rangle$	0.18	0.06	1.58	0.34
$\left\langle (\Omega)^3 \right\rangle$	17.00	8.00	1.09	21.58
$\left\langle (\Omega)^4 \right\rangle$	2858.00	1011.00	1.50	5416.22
$\left\langle \Omega^4 \right\rangle / \left\langle \Omega^2 \right\rangle^2$	65.10	22.20	1.55	122.40

We note that the discretization error is the only type of error estimated in this example. The linear system of equations associated with solving for the pressure in these calculations is accomplished "exactly" (i.e., the iteration errors are decreased to the roundoff limit) through a spectral transformation, and the time integration is explicit for all nonlinear terms. Nonetheless, even when the mesh resolution is extremely coarse by traditional standards of the turbulent flow community, the numerical uncertainty associated with discretization error is nearly equal or larger than the statistical uncertainty associated with turbulence.

7.1. Example: Vortex Shedding Validation Study

7.1.1. Drekar Description

Drekar is an unstructured, fully implicit, finite element Navier-Stokes solver with the capability to include coupled conjugate heat and mass transfer effects [37-39]. It is based on the earlier Charon code, but in relative terms is immature and under considerable development. The spatial discretization is currently an equal-order stabilized finite element (SFE) approach that can employ linear or quadratic interpolation. Convection stabilization is achieved through the streamline-upwind-Petrov-Galerkin (SUPG) approach, and discontinuity-capturing operators can also be included as part of the stabilized finite element formulation [35, 36, 38, 39]. The temporal discretization is based on a variable-order fully implicit multi-step backward differentiation formula (BDF) method that can vary from first-order (also known as Backward Euler) to a fifth-order BDF method (BDF5). In this work the second-order BDF2 is utilized. A parallel fully coupled nonlinear solution procedure based on a scalable algebraic multilevel preconditioned Newton-Krylov method is used to solve the discretized system of equations [13].

7.1.2. Drekar Validation and Error Estimates

Because Drekar is a relatively new code, we decided to proceed with a validation study on a phenomenon related to Grid-to-Rod-Fretting (GTRF). Specifically, the vortex shedding from the mixing vanes and grid spacers is essential to effectively simulate this phenomenon. For this reason, the vortex shedding frequency associated with the classical flow over a right circular cylinder was examined. This classical problem has been examined at a large range of Reynolds numbers, and its solution should provide a confidence building exercise for Drekar. The vortex shedding is characterized by the Strouhal number, which is defined by $St= (d\ l)/U,$ where U is the flow velocity, d is the cylinder diameter, and l is the frequency of vortex shedding (the Reynolds number $Re=d\ U/\mu$, where μ is the viscosity is the other dimensionless quantity of importance). The classical experimental studies can be found is Schlichting's text and we reproduce them in Table 5.

Table 5. The relevant Strouhal numbers from classical experiments with error bars estimated from the plots of the historical data [33].

Flow	Re=60	Re=100	Re=400	Re=1000
St	0.136±0.006	0.164±0.005	0.205±0.005	0.212±0.006

Drekar was used with second-order time and space differencing (with linear interpolation and SUPG) on a sequence of grids. For all calculations shown here, we expect second-order results for mesh-converged global quantities. We measured the grid scale by the RMS length of the mesh. We note that this grid is far more regular and classical in structure than the highly irregular mesh used for the GTRF calculations.

Figures 7 and 8 show the grid and typical solution for the vortex shedding simulations with Drekar. Figure 9 shows the velocity variations associated with the vortex shedding and the accompanying data reduction used to determine the Strouhal number.

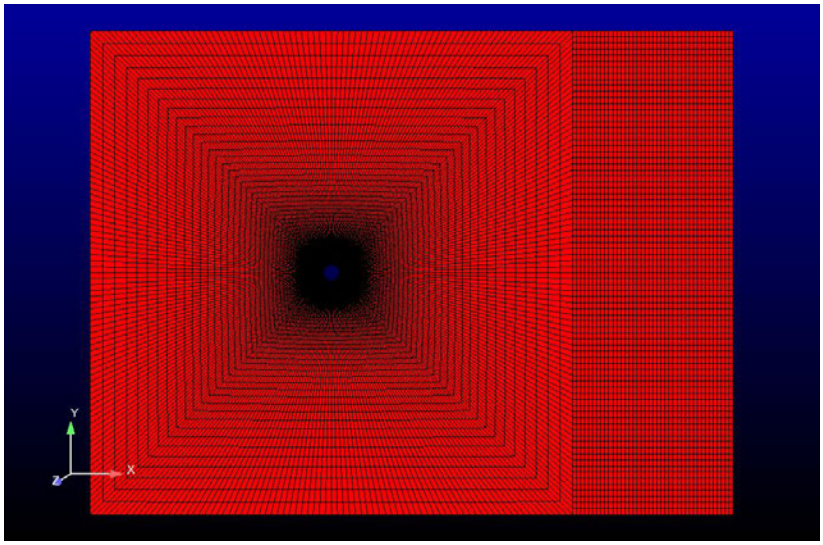


Figure 7. A snapshot of the grid used for the vortex shedding simulations.

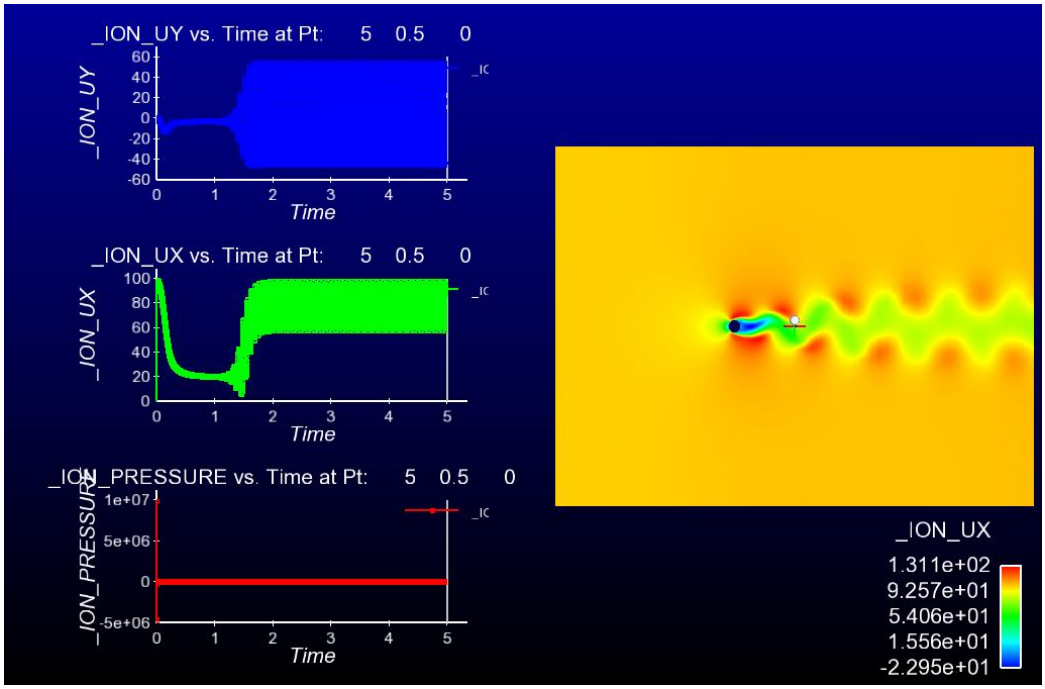


Figure 8. A snapshot of the vortex shedding simulation showing the flowfield, and several of the variables plotted in the form of a time series.

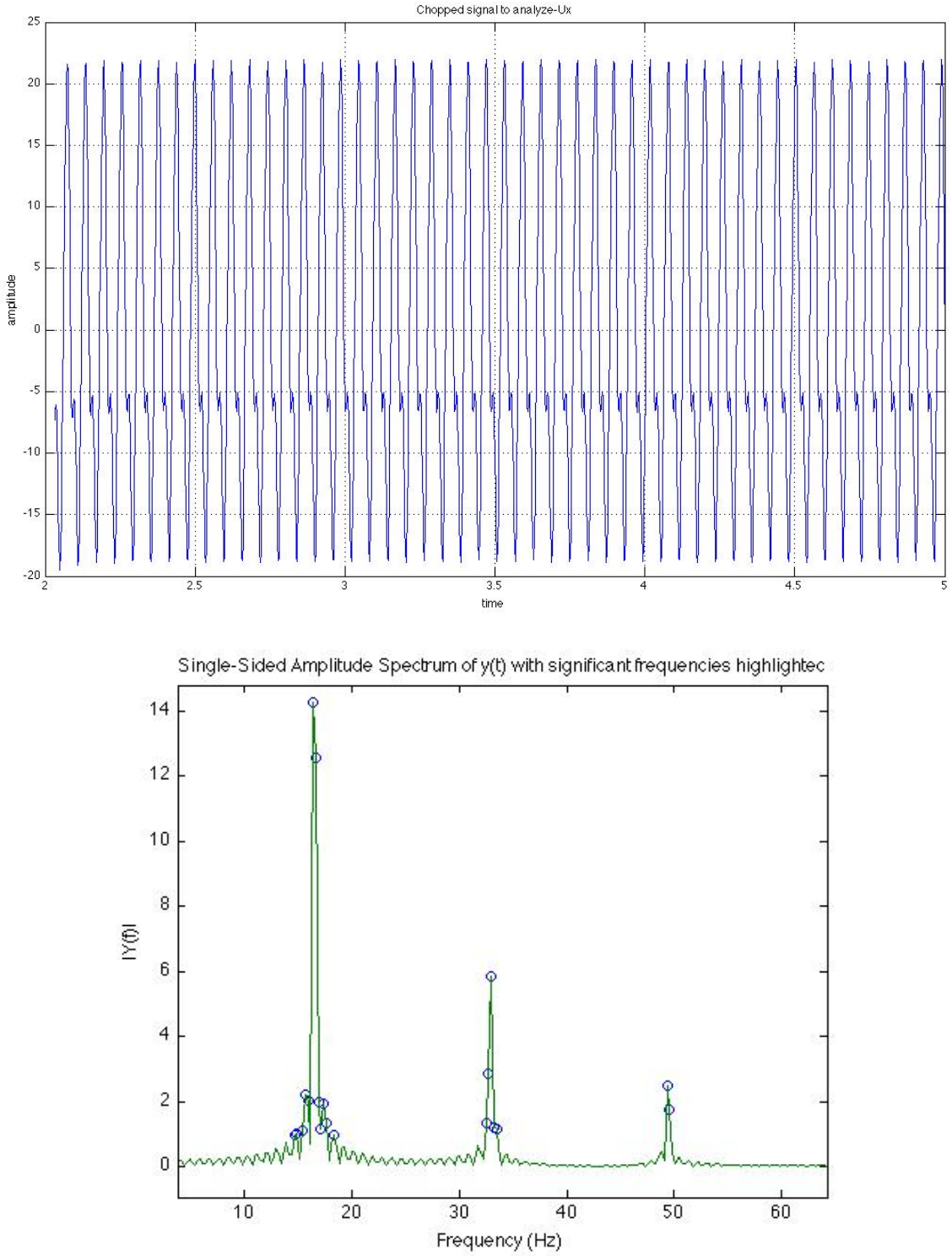


Figure 9. Data reduction is accomplished the FFT of the fluctuating velocity at a location in space and reporting the frequency where the FFT has its largest peak.

Table 6. Drekar results for vortex shedding at Re=100.

Case	Δt	RMS h	St	Δ(St) 95%
1	0.002	0.054111988	0.110474853	0.001219
2	0.002	0.023801688	0.152492294	0.004537
3	0.002	0.010786082	0.164777976	0.004979
4	0.002	0.005264375	0.165127187	0.004369
5	0.008	0.010786082	0.134265285	0.005306
6	0.004	0.010786082	0.154923058	0.00756
7	0.001	0.010786082	0.165999038	0.00491
8	0.0005	0.010786082	0.1661293	0.005012

Table 6 presents Drekar results for vortex shedding at Re = 100. We will use the following error ansatz to verify the behavior of these solutions:

$$St(h,\Delta t) = St_0 + C_h h^{\alpha} + C_{\Delta t}(\Delta t)^{\beta} \tag{25}$$

The average Strouhal number estimate has the following structure:

$$St(h,\Delta t) = 0.170935 - 7.94234h^{1.69637} - 47.434(\Delta t)^{1.50502} \tag{26a}$$

For the lower bounding estimate given statistically, the convergence is:

$$St(h,\Delta t) = 0.167213 - 5.83905h^{1.61558} - 21.7375(\Delta t)^{1.33337} \tag{26b}$$

and the upper bounding estimate is:

$$St(h,\Delta t) = 0.17470 - 10.3214h^{1.76271} - 164.206(\Delta t)^{1.77237} \tag{26c}$$

Overall, the solutions indicate that the code is performing well for this problem although the meshes are a bit under-resolved. Furthermore, the solution is converging to a solution that is 1.8 to 6.5% too high. This may be related to the size of the domain or three-dimensional effects neglected in the simulation. In this case it make sense to recomputed the convergence and include only the triplet of the finest three resolutions in the convergence rate. We find that this produces a significantly lower level of numerical uncertainty. The results for this application of our procedure are:

$$St(h,\Delta t) = 0.168561 - 190.70h^{2.5584} - 5.7963(\Delta t)^{1.2418} \tag{27a}$$

for the average Strouhal number, and:

$$St(h,\Delta t) = 0.162377 - 1726.87h^{3.1575} - 474.21(\Delta t)^{2.0380} \quad (27b)$$

$$St(h,\Delta t) = 0.171947 - 15428h^{3.7435} - 1774.7(\Delta t)^{2.2061} \quad (27c)$$

for the lower and upper bound respectively. The lower bounds value is almost identical to the experimental observed value, and the range of asymptotic Strouhal numbers is smaller. with higher rates of observed convergence. This would indicate that the coarser meshes and larger time steps are insufficient to resolve the problem. The uncertainty quantification (UQ) estimate computed using these values is quite small with a value of 0.36% of the Strouhal number. This is smaller than the range supported by the statistically bounded values. We suggest taking the larger reported bound as the uncertainty, which evaluates to 5.83% of the Strouhal number.

8. Detailed Workflow Procedure

In this section, we expand on the details of the proposed workflow. There are other workflow approaches (see, e.g., Johnson, et al. [9]), and the steps described below are by no means exhaustive. The proposed steps do, however, standardize a calculation verification workflow that can be conducted by a code team (developers and testers) for the purpose of estimating numerical uncertainty. Ideally, the code verification process should be conducted regularly (as well as on demand) so that incorrect implementations impacting mathematical correctness are detected as soon as possible. The general consensus in software development is that the cost of bugs is minimized if they are detected as close as possible to their introduction.

This procedure assumes that the code team is using a well-defined software quality assurance (SQA) process, and the code verification is integrated with this activity. Such SQA includes source code control, regression testing, and documentation, together with other project management activities. For consistency and transparency, we recommend performing the code verification in the same manner and using the same type of tools as other SQA processes.

1. **Starting with an implementation (i.e., code) that has passed the appropriate level of SQA and code verification scrutiny, choose the executable to be examined.** Calculation verification can be a resource-intensive activity involving substantial effort to perform. Calculation verification should be applied to the same version of the code that analysts would use for any important application. The notion that verification and validation should be applied to the same code is important to keep in mind. This process should be applied to the specific version of the code used throughout the entire V&V UQ activity.
2. **Provide an analysis of the numerical method as implemented including accuracy and stability properties**. The analysis should be conducted using any one of a variety of standard approaches. Most commonly, the von Neumann-Fourier method could be employed to analyze the stability. For nonlinear systems, the method of modified equation analysis can be used to define the expected rate and form of convergence. The form and nature of the solution being sought can also influence the expected behavior of the numerical solution. For example, if the solution is discontinuous, the numerical solution will not achieve the same order of accuracy as for a smooth solution. Finite element methods can be analyzed via other methods to define the form and nature of the convergence (including the appropriate norm for comparison).

3. **Produce the code input to model the problem(s) for which the code verification will be performed**. Each problem is run using the code's standard modeling interface, as for any physical problem that would be modeled. It can be a challenging task to generate code input that correctly specifies a particular problem[5]; e.g., special routines to generate particular initial or boundary conditions that drive the problem may be required, and these routines must be correctly interfaced to the code. It is advisable to consider the complexities and overhead associated with such considerations prior to undertaking such code verification analyses.
4. **Select the sequence of discretizations to be examined for each solution**. Verification necessarily involves convergence testing, which requires that the problem be solved on multiple discrete representations (i.e., grids or meshings). This is consistent with notions associated with h-refinement, although other sorts of discretization modification can be envisioned. The mathematical aspects of verification are typically most conveniently carried out if the discretizations are factors of two apart.
5. **Run the code and provide a of means of producing appropriate metrics to compare the numerical solutions**. The solutions to the problem are computed on the discretizations. Most commonly and as discussed above, these metrics take the form of norms (i.e., p-norms such as the L2 or energy norm). The selection of metrics is inherently tied to the mathematics of the problem and its numerical solution. The metrics can be computed over the entire domain, in subsets of the domain, on surfaces, or at specific points. The domain over which the metrics are evaluated and the analysis to be conducted must be free of any spurious solution features (due, e.g., to numerical waves erroneously reflected from computational boundaries).
6. **Use the comparison to determine the sequence of errors in the discretizations**. Using the well-defined metrics for each solution, the error can be computed for each discrete representation. Ideally, there will be a set of metrics available, providing a more complete characterization of the problem and its solution.
7. **The mesh sequence allows the determination of the rate-of-convergence for the method, which is compared to the theoretical rate**. With a sequence of errors in hand, the demonstrated convergence rate of the code for the problem is estimated. The theoretical convergence rate of a numerical method is a key property. Verification relies upon comparing this rate to the demonstrated rate of convergence. Evidence supporting verification is provided when the demonstrated convergence rate is consistent with the theoretical rate of convergence. This can be a difficult inference to draw, because the theoretical rate of convergence is a limit reached in an asymptotic sense, which cannot be attained for any finite discretization. As a consequence, there are unavoidable deviations from the theoretical rate of convergence, to which judgment must be applied.
8. **Using the results, complete an assessment of the method's implementation correctness**. Based on the discrete solutions, errors, and convergence rate, a decision on the correctness of a model can be rendered. This judgment is applied to a code across the full suite of verification test problems.
    - The assessment can be positive, i.e., the convergence rate is consistent with the method's expected accuracy.

[5] Trucano et al. [45] refer to this concept as the "alignment" between a code and a specific problem (either verification or validation).

- The assessment can be negative, i.e., the convergence rate is inconsistent with the method's expected accuracy.
- The assessment can be inconclusive, i.e., one cannot defensibly demonstrate clearly uniform consistency or inconsistency with the method's expected accuracy. For example, the convergence rate is nearly the correct rate, but the differences between the expected rate and the observed rate is uncomfortably large, potentially indicating a problem.

Figure 10 and Figure 11 show the entire process in diagrams that conceptually expand the line for code verification in Figure 4. This process should be repeatable and available on demand. As noted in the introduction to this section, having the code verification integrated with the ongoing SQA activity and tools can greatly facilitate this essential activity. The solution verification process is not monolithic, but is instead flexible, and should meet the needs of the specific application. For this reason we include two versions of the flowchart to facilitate this mindset.

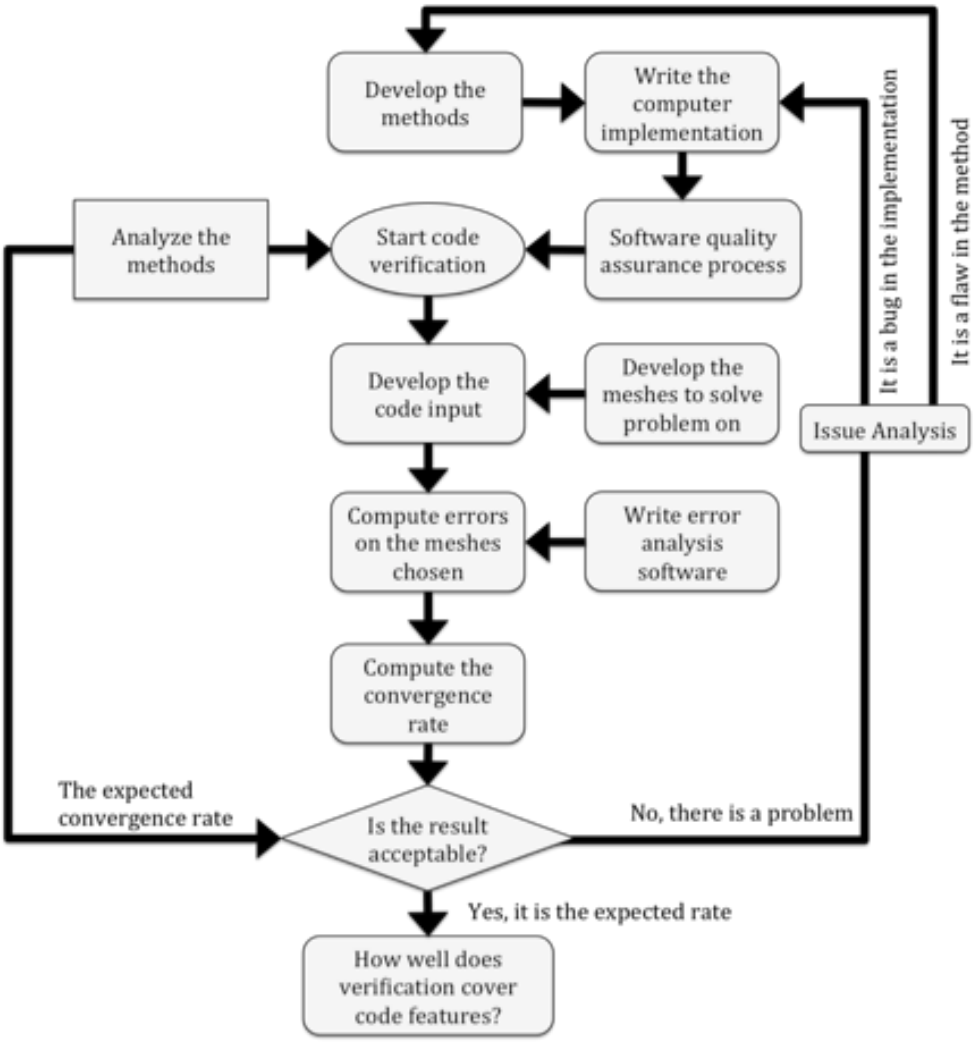


Figure 10. The flowchart version of the list of activities for code verification, which can be interpreted as an expansion of the simple expression of this activity.

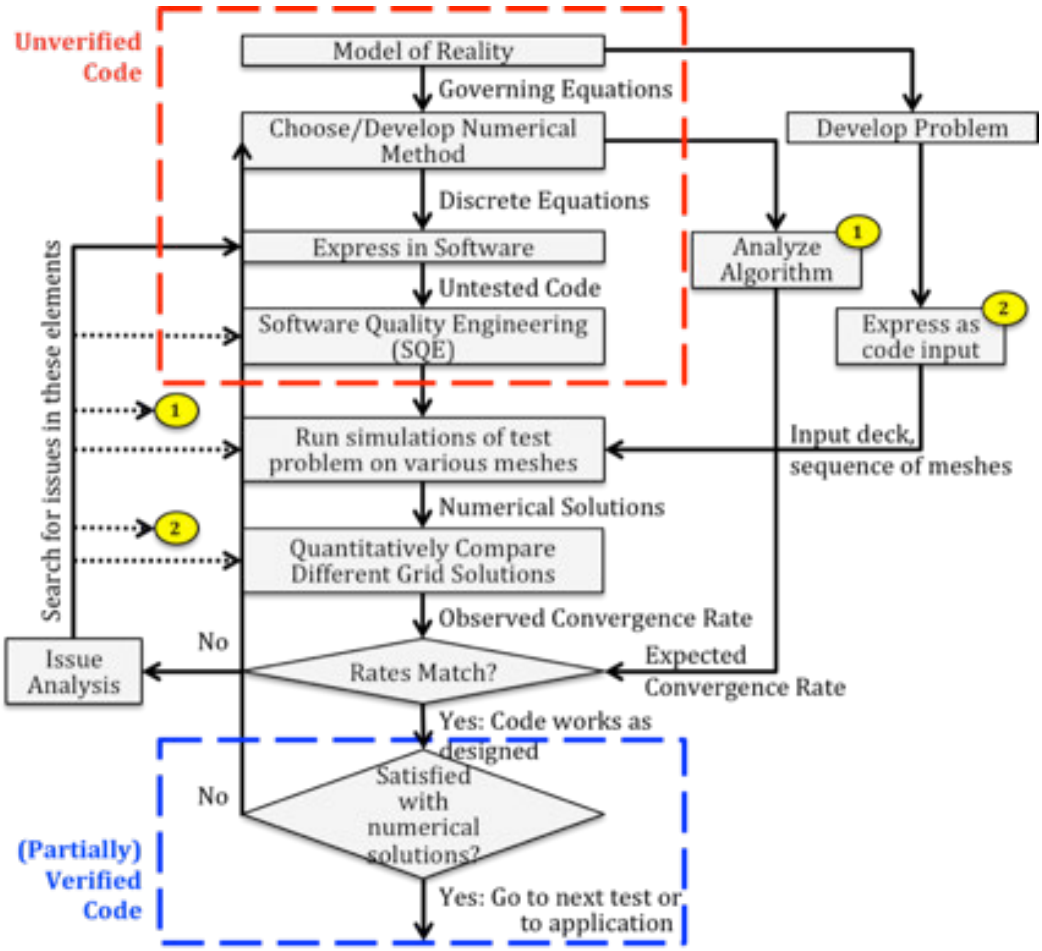


Figure 11. The flowchart version of the list of activities for code verification, which can be interpreted as an expansion of the simple expression of this activity.

9. Conclusions

In this chapter, we have described the concept of verification, a well-defined process by which the correctness of the implementation of a numerical algorithm in scientific software can be evaluated. There are two complementary activities: (i) code verification, which quantitatively compares the theoretical order of accuracy of a method with the empirical order of accuracy for a problem with a known solution, and (ii) calculation verification. We have provided a detailed workflow for conducting calculation verification.

While approaches to verification are reasonably well codified and fairly widely used, there remain aspects of these analyses that can be difficult to resolve. Most verification cases encountered by the code user will likely be typical; however, difficult cases almost always arise, eventually. Unless the analyst has chosen exceedingly simple problems, particular verification problems may present their own challenges that will require insight, innovation, and determination on the part of the analyst to resolve. Despite these obstacles, verification is a necessary part of the "due diligence" of a scientific code development project and an essential element in producing superior simulation code that can be used for high-consequence analysis and decision-making.

Acknowledgements

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

References

1. American Institute of Aeronautics and Astronautics, "Guide for the verification and validation of computational fluid dynamics simulations," AIAA-G-077-1998, American Institute of Aeronautics and Astronautics, Reston, VA, 1998.
2. American Society of Mechanical Engineers, *Guide for Verification and Validation in Computational Solid Mechanics*, V&V 10 – 2006, American Society of Mechanical Engineers, New York, NY, 2006.
3. Donzis, D. A., "Scaling of Turbulence and Turbulent Mixing Using Terascale Numerical Simulations," Ph.D. Thesis, Georgia Institute of Technology, 2007.
4. Eça, L., Hoekstra, M., "Discretization Uncertainty Estimation Based on a Least Squares Version of the Grid Convergence Index," *Proceedings of the Second Workshop on CFD Uncertainty Analysis*, Lisbon, Oct. 2006.
5. Eça, L., Hoekstra, M., eds., *Proceedings of the 1st, 2nd, and 3rd Workshops on CFD Uncertainty Analysis*, http://maretec.ist.utl.pt/html_files/CFD_workshops/
6. Eça, L., Hoekstra, M., "Evaluation of numerical error estimation based on grid refinement studies with the method of manufactured solutions," *Computers & Fluids*, Vol. 38, 2009, pp. 1580–1591.
7. Hanson, K. M., Hemez, F. M., "A Framework for Assessing Confidence in Computational Predictions," *Experimental Techniques*, Vol. 25, pp. 50–55, 2001.
8. Hemez, F. M., "Non-Linear Error Ansatz Models for Solution Verification in Computational Physics," Los Alamos National Laboratory report LA-UR-05-8228, 2005.
9. Johnson, R. W., Schultz, R. R., Roache, P. J., Celik, I. B., Pointer, W. D., and Hassan, Y. A., "Processes and Procedures for Application of CFD to Nuclear Reactor Safety Analysis," Idaho National Laboratory report INL/EXT-06-11789, 2006.
10. Kamm, J., Brock, J., Rousculp, C., and Rider, W., "Verification of an ASCI Shavano Project Hydrodynamics Algorithm," Los Alamos National Laboratory report LA-UR-03-6999, 2003.
11. Kamm, J. R., Brock, J. S., Brandon, S. T., Cotrell, D. L., Johnson, B., Knupp, P., Rider, W. J., Trucano, T. G., and Weirs, V. G., "Enhanced Verification Test Suite for Physics Simulation Codes," Los Alamos National Laboratory report LA-14379, Lawrence Livermore National Laboratory report LLNL-TR-411291, Sandia National Laboratories report SAND2008-7813, 2008.
12. Kelley, C. T., "Iterative Methods for Linear and Nonlinear Equations," SIAM, Philadelphia, PA, 1995.
13. Knoll, D. A., and Keyes, D. E., "Jacobian-free Newton-Krylov methods: a survey of approaches and applications," *Journal of Computational Physics*, Vol. 193, pp. 357-397, 2004.

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

15. Li, S., Rider, W. J., and Shashkov, M. J., "Two-Dimensional Convergence Study for Problems with Exact Solution: Uniform and Adaptive Grids," Los Alamos National Laboratory report LA-UR-05-7985, 2005.

16. Margolin, L. G., Shashkov, M. J., "Finite volume methods and the equations of finite scale: A mimetic approach," *International Journal of Numerical Methods* in Fluids, Vol. 56, pp. 991–1002, 2008.

17. Majda, A., and Osher, S., "Propagation of error into regions of smoothness for accurate difference approximations to hyperbolic equations," *Communications in Pure and Applied Mathematics*. Vol. 30, pp. 671–705, 1977.

18. Nelson, R., Unal, C., Stewart, J., and Williams, B., "Using Error and Uncertainty Quantification to Verify and Validation Modeling and Simulation," Los Alamos National Laboratories report LA-UR-10-06125, 2010.

19. Oberkampf, W. L., Trucano, and T. G., "Verification and Validation in Computational Fluid Dynamics," *Progress in Aerospace Sciences*, Vol. 38, pp. 209–272, 2002.

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

21. Oberkampf, W. L., and Trucano, T. G., "Verification and Validation Benchmarks," *Nuclear Design and Engineering,* Vol. 23, pp. 716–743, 2007; Sandia National Laboratories report SAND2007-0853, 2007.

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

23. Oreskes, N., Shrader-Frechette, K., and Belitz, K., "Verification, validation, and confirmation of numerical models in the earth sciences," *Science,* Vol. 263, pp. 641–646, 1994.

24. Rider, W. J., Kamm, J. R., and Weirs, V. G., "Code Verification Workflow in CASL," Sandia National Laboratories report SAND2010-7060P, 2010.

25. Rider, W. J., Kamm, J. R., and Weirs, V. G., "Verification, Validation and Uncertainty Quantification Workflow in CASL," Sandia National Laboratories report SAND2011-0234P, 2011.

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

27. Roache, P., *Fundamentals of Verification and Validation*, Hermosa Publishers, Albuquerque, 2009.

28. Roy, C. J., "Review of Code and Solution Verification Procedures for Computational Simulation," *Journal of Computational Physics*, Vol. 205, pp. 131–156, 2005.

29. Roy, C. J., "Review of Discretization Error Estimators in Scientific Computing," 48th AIAA Aerospace Sciences Meeting, AIAA Paper 2010-126, 2010.

30. Roy, C. J., and Oberkampf, W. L., "A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing," *Computer Methods in Applied Mechanics and Engineering*, Vol. 22, pp. 2131-2144, 2011.

31. Sargent, R. G., "Verification and Validation of Simulation Models," *Proceedings of the 1998 Winter Simulation Conference*, Medeiros, D. J., Watson, E. F., Carson, J. S., and Manivannan, M. S., eds., pp. 121–130, 1998.

32. Sargent, R. G., "Some Approaches and Paradigms for Verifying and Validation Simulation Models," *Proceedings of the 2001 Winter Simulation Conference*, Peters, B. A., Smith, J. S., Medeiros, D. J., and Rohrer, M. W., eds., pp. 106–114, 2001.

33. Schlichting, H., *Boundary Layer Theory*, McGraw-Hill, 1979.

34. Schwer, L. E. "An Overview of the PTC 60 / V&V 10 Guide for Verification and Validation in Computational Solid Mechanics," American Society of Mechanical Engineers (ASME), 2006. http://cstools.asme.org/csconnect/pdf/CommitteeFiles/24816.pdf

35. Shadid, J., Salinger, A.G., Pawlowski, R. P., Lin, P. T., Hennigan, G. L., Tuminaro, R., and Lehoucq, R. B., "Large-Scale Stabilized FE Computational Analysis of Nonlinear Steady-State Transport / Reaction Systems," *Computational Methods in Applied Mechanics and Engineering*, Vol. 195, pp. 1846–1871, 2006.

36. Shadid, J. N., Pawlowski, R. P., Banks, J. W., Chacon, L., Lin, P. T., and Tuminaro, R., "Towards a Scalable Fully-Implicit Fully-Coupled Resistive MHD Formulation With Stabilized FE Methods," *Journal of Computational Physics,* Vol. 229, No. 20, pp. 7649–7671, 2010.

37. Shadid, J. N., Smith, T. M., Pawlowski, R. P., Cyr, E. C., Weber, P. D., and Turner, D. Z., "Initial Drekar LES CFD Simulations for Grid-to-Rod Fretting (GTRF)," Sandia National Laboratories, 2011.

38. Shadid, J. N., Smith, T. M., Pawlowski, R. P., and Cyr, E. C., "A Summary of SNL Drekar1 Large-scale Parallel CFD Code for Use in the Demonstration and Evaluation of Methods for VERA-CFD," Sandia National Laboratories, 2011.

39. Smith, T. M., Shadid, J. N., Pawlowski, R. P., and Cyr, E. C., "Reactor Core Sub-Assembly Simulations Using a Stabilized Finite Element Method," 14th International Topical Meeting on Nuclear Reactor Thermal-Hydraulics, NURETH Paper 14-500, 2011.

40. Sornette, D., Davis, A. B., Ide, K., Vixie, K. R., Pisarenko, V., and Kamm, J. R., "Algorithm for Model Validation: Theory and Applications," *Proceedings of National Academy of Sciences*, Vol. 104, pp. 6562–6567, 2007.

41. Stern, F., Wilson, R. V., Coleman, H. W., Paterson, E. G., "Comprehensive Approach to Verification and Validation of CFD Simulations—Part 1: Methodology and Procedures," *Journal of Fluids Engineering*, Vol. 123, pp. 793–802, 2001.

42. Stern, F., Wilson, R. V., Shao, J., "Quantitative V&V of CFD Simulations and Certification of CFD Codes," *International Journal for Numerical Methods in Fluids*, Vol. 50, pp. 1335–1355, 2006.

43. Timmes, F. X., Fryxell, B., and Hrbek, G. M., "Spatial-Temporal Convergence Properties of the Tri-Lab Verification Test Suite in 1D for Code Project A," Los Alamos National Laboratory report LA-UR-06-6444.

44. Trucano, T. G., Pilch, M., and Oberkampf, W. L., "On the Role of Code Comparisons in Verification and Validation," Sandia National Laboratories report SAND2003-2752, 2003.

45. Trucano, T. G., Swiler, L. P., Igusa, T., Oberkampf, W. L., and Pilch, M., "Calibration, validation, and sensitivity analysis: What's what," *Reliability Engineering and System Safety*, Vol. 92, pp. 1331–1357, 2006.

46. Xing, T., and Stern, F., "Factors of Safety for Richardson Extrapolation," *Journal of Fluids Engineering*, Vol. 132, pp. 061403-1– 061403-13, 2010.

47. Zienkiewicz, O. C., and Zhu, J. Z., "The Superconvergent Patch Recovery and a Posteriori Error Estimates. Part 2: Error estimates and adaptivity," *International Journal of Numerical Methods in Engineering*. Vol. 33, pp. 1356–1382, 1992.

Error Transport Methods in Solution Verification

Peter A. Cavallo

Combustion Research and Flow Technology, Inc., Pipersville, PA 18947

Abstract

Error transport methods are a class of solution verification techniques that seek to model the origination and subsequent manifestation of simulation error due to discretization of the computational domain. Using these methods, one may quantify the degree of numerical error due to the mesh. With this error quantification in hand, the analyst is better able to assess whether poor agreement with data is due to inadequate mesh resolution or improper physical modeling. A review of solution verification techniques based on the solution of error transport equations (ETE) is presented, as well as a summary of current state of the art methodology in ETE techniques. Verification procedures for the use of ETE solutions are also discussed, and several 2D and 3D applications are offered that demonstrate the effectiveness of the approach.

1. Introduction

1.1. Errors in Computational Fluid Dynamics Simulations

As computational fluid dynamics (CFD) solvers have matured, their role in the support of system design and evaluation has expanded considerably. The increasing utilization of unstructured grid methods in particular, with their relatively fast setup process, has contributed to the growing use of CFD in many applications of interest. The design and evaluation of new and emerging air vehicles has increasingly relied on CFD analyses for an improved understanding of the aerodynamic characteristics and to supplement ground test data. These analyses can assist in examining scaling issues in the transition from ground test to flight test, and in addition, offer a means of assessing dynamic loads over time scales greater than those achievable in ground tests.

The sole use of CFD as an analysis tool is not yet an accepted practice, but has instead been employed as a complement to experiments and other analyses. While CFD analysis plays an ever-increasing role in supporting the design and experimental testing of air vehicles, it is wind tunnel data that is generally regarded as the "truth" upon which design decisions are made. The perceived risks associated with using CFD results may be somewhat mitigated if the solution errors are better understood, automatically controlled, and reduced. Such processes would increase the trust in numerical predictions. This is particularly true for simulations of propulsive phenomena where physical modeling is often more complex and plays a vital role. While solution verification is generally considered important, there remains little consensus among the CFD community on how to bound the error or uncertainty in any given calculation.

Solution errors are inherent in any CFD simulation. The numerical simulation and prediction of complex fluid flow phenomena for realistic scenarios of interest are a challenging endeavor, and

computations obtained using CFD codes are often not reliable. There are numerous potential sources of error in any given CFD simulation, many of which are summarized in Table 1. Specific areas of concern include spatial and temporal accuracy, improper physical modeling, and human factors. With many potential error modes, there is no single path towards establishing solution credibility. Rather, the task of solution verification should be addressed on multiple fronts.

Table 1. Types of errors in CFD simulation.

Spatial	**Temporal**
• Truncation error of scheme • Resolution of solution features with varying length scales • Mesh quality considerations • Poor representation of surface curvature	• Formal accuracy of integration scheme • Size of time step • Solution stability constraints • Convergence of inner iterations • Competing time scales (fluid vs. chemical)
Physical Modeling	**Other Sources**
• Choice of governing equations • Choice of near-wall treatment • Limitations of turbulence model • Limitations of reaction set • Limitations of two-phase model • Validity of boundary conditions	• Programmatic errors (bugs) • Usage errors (improper inputs, etc.) • Geometry errors (CAD model) • Numerical scheme (artificial dissipation, upwinding, limiters, etc.) • Insufficient iterative convergence

Spatial errors are perhaps the most recognizable and readily addressed. In all CFD applications, grid resolution is a fundamental issue. While it is readily accepted that poor mesh resolution has a detrimental impact on solution accuracy, the generation of a suitable mesh often requires inputs from an experienced CFD analyst. One of the challenges in generating meshes for complex flow fields is that solution gradients and regions of critical concern are generally not known *a priori*. As a result, computational meshes generated with inadequate point distributions may produce solutions that do not fully reflect reality. Examples include smeared shock structures with reduced downstream pressures, vortices that dissipate prematurely, and shear layers that mix too rapidly due to mesh-induced diffusion. At worst, a poorly resolved mesh may fail to capture critical relevant flow features. In this regard, structured grid solvers possess an inherent advantage over unstructured grid schemes, as it is far easier to implement higher-order schemes on structured meshes. These higher-order stencils naturally serve to minimize mesh-induced diffusion, and more readily lend themselves to applications in acoustics and similar disciplines where high fidelity is required. For unstructured grids, extensions beyond second-order accuracy are prohibitively expensive, as demonstrated by Barth [1]. Therefore, there is a greater need to address spatial accuracy concerns for unstructured solvers, which are generally limited to second-order.

However, the errors present in CFD simulations are not limited to mesh discretization. Errors may be introduced by invalid or incomplete physical modeling. This is particularly true for phenomena that are not well understood, such as ignition chemistry, particle-wall interactions, and boundary layer transition. Often models of complex processes are incorporated in an empirical fashion, and include tunable coefficients based on experimental curve fits, such as turbulence model constants and activation energies in reaction rates. Calibration of these models naturally leads to some

limitations in their applicability, with validity restricted to a particular range in the dependent variables (e.g., Reynolds number or temperature range). Other sources of error include temporal errors, round off and precision errors, programmatic errors, and use of the simulation code. Use of a validated code in no way guarantees credible simulations as the analyst himself can play a large part in establishing simulation credibility [2]. The human element introduces different forms of error in the analysis, as noted by Mehta [3], including making decisions based on insufficient information, errors in interpretation and risk assessment, and subjective personal attachment to the work performed, causing the analyst to perceive the results as reality without sufficient evidence to support the claim.

With so many contributing factors, CFD analysts, particularly those lacking years of experience and insight on which to rely, have a definitive need for tools to assess the various potential sources of solution error. For example, discretization error has well-known consequences in the form of numerical diffusion, and the generation of an appropriate grid is subject to the judgment and experience level of the analyst. Once a solution is obtained, alteration of an initial mesh is again left in the hands of the user. Guidance in where and how to improve subsequent meshes, or automated mesh adaptation would facilitate this task. Systematic grid refinement studies have historically been one means of controlling and reducing spatial errors, however, they are seldom done in practice due to the considerable time and labor costs involved. Any means of expediting, automating, and reducing the man-in-the-loop effort associated with this process is therefore extremely desirable.

The need for assessing solution accuracy has been recognized for some time. Archival publications such as the AIAA Journal and the Journal of Fluids Engineering have sought to enforce standards on the presentation of numerical analyses obtained using CFD, and comparison with experimental data alone is no longer sufficient. In the late 1980s, the American Institute of Aeronautics and Astronautics (AIAA) Committee on Standards for Computational Fluid Dynamics was formed, with the goal of improving the value of solutions obtained from CFD tools. Their work resulted in the AIAA Guide for Verification and Validation (V&V) of CFD Simulations [4]. However, as pointed out by Frietas [5], neither these editorial policies nor the guide define procedures for how numerical uncertainty is to be addressed. The AIAA Guide, for example, sought to formalize definitions and methodology, and does not present suitable techniques. While recent workshops in computational aerodynamics have raised the standard in solution accuracy by stressing mesh refinement studies [6-9], few practical tools are available to CFD analysts for assessing and reducing the errors present in their results.

This is particularly true for the propulsion community. For the past several years, the Joint Army Navy NASA Air Force (JANNAF) Modeling and Simulation Subcommittee have held a series of workshops on V&V with the intention of compiling a set of recommended practices similar to the AIAA Guide, but with an emphasis on the physics and modeling requirements specific to propulsive applications. Several representatives from the CFD community, including the author [10], have presented their perspectives on the subject of simulation credibility.

While the terms V&V are often referred to in tandem, they address separate matters. Verification is concerned with certifying the numerical accuracy of the simulation, namely verifying that the selected governing equations are solved correctly. The process of validation answers the question of how well the selected governing equations and physical models reflect reality. Typically the former is answered by demonstrating grid convergence, while a comparison with test data is generally required for the latter. Only when grid-induced errors, iterative convergence errors, and usage errors are bounded and understood, can physical modeling errors be properly addressed.

1.2. Challenges of Propulsive Flow Simulations

While it is known that grid refinement affects the overall accuracy of numerical simulations, it has an even more marked impact on the solution of the species transport equations in reacting flow models. Unlike the standard mass, momentum, and energy conservation equations, the species conservation equations often feature stiff source terms representing chemical reactions that are sensitive to grid resolution and grid quality. Accurate prediction of species production impacts the prediction of pressure and temperature in turn. True assessments of reduced reaction mechanisms, turbulent combustion models, etc. are only valid once grid-induced errors are quantified and minimized. Thus, solution verification must go hand-in-hand with model validation, and effective tools for performing the former are needed.

Thermal and species turbulent transport in high-speed flows occur at different rates than turbulent momentum transport. These rates are often faster and the difference can be accounted for in traditional RANS simulations by using turbulent Prandtl and Schmidt numbers in the diffusion coefficient for the energy and species transport equations. Scalar fluctuations impact combustion by their effect on thermal/species transport and by their influence on the chemistry. The use of scalar fluctuation models is particularly important for combusting flows where the species and/or temperature gradients are more severe than in non-reacting mixing problems [11-14].

In non-reacting CFD simulations, the integration time step Δt generally scales with the fluid dynamic time scale. In a reacting simulation, the time scales associated with the chemistry also must be accounted for. Implicit single-step chemistry solution algorithms are often used, where the solution is advanced in time by the time step Δt and the instantaneous chemical source terms and the Jacobian matrix of the chemical source terms are evaluated only once, since only one unified time step is taken. However, under certain conditions, accurately capturing ignition phenomena and rapid intermediate species build-up can require very small time steps and thus be extremely computationally expensive, even for relatively small chemical systems. This occurs when the characteristic chemical time scale to be resolved is much smaller than the characteristic fluid-dynamic time scale, as shown in Figure 1.

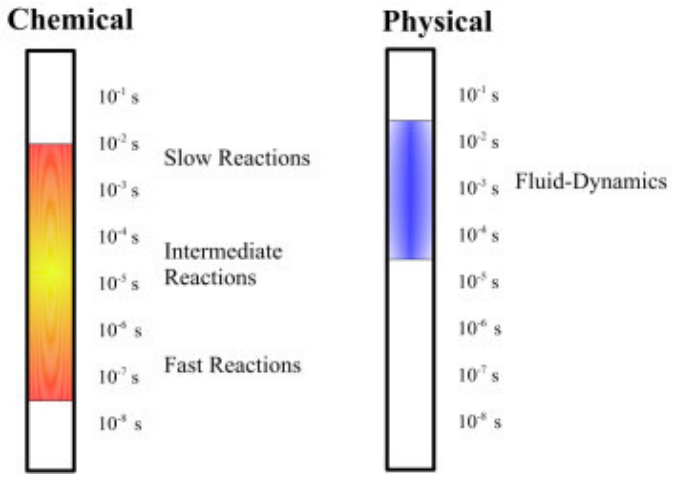


Figure 1. Disparity of characteristic time scales in a chemically reacting system.

In order to allow for larger time steps that scale with the fluid-dynamic time scale while guaranteeing a solution that is accurate and mass-conservative, multi-dimensional CFD codes often decouple the solution of the chemistry part of the governing equations from the convective-

diffusive part based on an operator splitting numerical scheme. The resulting chemistry step entails solving a system of stiff ordinary differential equations (ODEs) for the chemical species, with the instantaneous species net production rates, $\dot{\omega}_k$, as the source terms. For a given Δt, a dedicated ODE solver may be employed that integrates the system of ODEs by automatically taking several sub-steps to accurately advance the thermo-chemical state in time, which is based on the required fractional time step dictated by the stiffness of the system of ODEs. These solution procedures are generally multi-step implicit methods with error correction and guarantees solution accuracy to a user-specified tolerance [15]. Ultimately, the chemical production term $\bar{D}(\bar{Q})$ is dependent on the local temperature and species mass fractions.

$$\bar{D}\left(\bar{Q}\right) = f\left(y_i, T\right) \tag{1}$$

Note that in some extended mechanisms, the rate coefficients and the production term may also be pressure dependent. Equation (1) may be evaluated numerically using a variety of techniques, including implicit treatment, stiff ODE solvers, In-Situ Adaptive Tabulation (ISAT) methods, etc., such as those found in Reference 16.

As an illustrative example, the 3D SCHOLAR angled fuel injector mixing and combustion experiments conducted at NASA Langley Research Center provide data for CFD model validation for an environment that typifies the complex chemistry found in a scramjet combustor [17, 18]. Mean temperature, wall pressure, and species measurements are available, making these data sets useful for analyzing the capabilities of the scalar fluctuation model (SFM) to capture turbulent mixing effects in more complex flows involving hydrogen injection into three-dimensional cross-flows. Data was obtained at four stream-wise planes downstream of the injector. Computational analyses performed by Rodriguez and Cutler showed sensitivity to boundary conditions, including temperature, species composition, and inflow turbulence parameters [19]. They found their solutions to be most sensitive to the values used for the turbulent Prandtl and Schmidt numbers, Pr_t and Sc_t, and concluded that it is unlikely that constant values for these parameters can be calibrated to be valid for a wide range of flows. They suggest that the use of models providing for variable Pr_t and Sc_t values are necessary to properly simulate the SCHOLAR combustion experiment. Keistler, et al. proposed a model for variable Pr_t and Sc_t values based on enthalpy and concentration variances [20], which was used to analyze the SCHOLAR combustor experiment, however, comparisons with the data have not shown overall consistency.

In Figures 2–4, we present a CFD analysis that clearly exhibits mesh-dependent behavior. The numerical analysis predicts variable Pr_t and Sc_t values using the SFM formulation, with compressibility corrections. The facility nozzle from the upstream heater was modeled to provide inflow boundary conditions for mean and turbulence quantities, to the test section. Near-wall modeling was used for no-slip walls, along with a constant wall temperature. Three levels of grid refinement were performed, denoted as Grid A, Grid B, and Grid C for subsequent levels of refinement, ranging from 868,000 to 6.4 million cells. A representative view of Grid C in the vicinity near the injector is provided in Figure 2. This domain was decomposed into an isolator section upstream of the injector, the injector section containing the angled fuel injector, and a downstream combustor section. The injector nozzle was included in a block that extended into the duct.

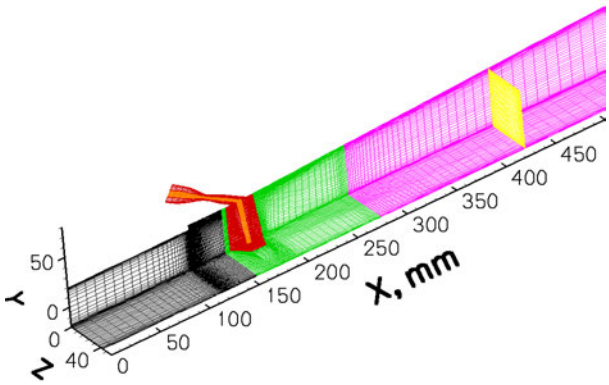


Figure 2. Schematic of SCHOLAR grid near injector with every other grid point shown for reference.

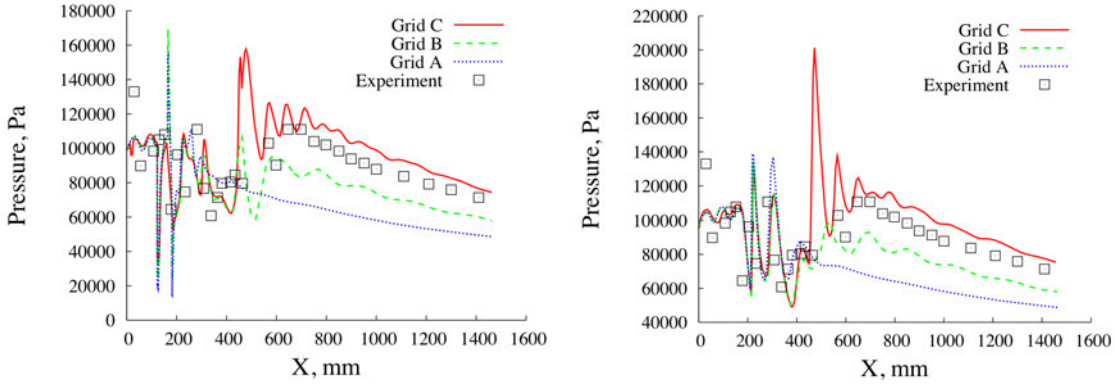


Figure 3. Lower (left) and upper (right) wall pressure grid sensitivity of the SCHOLAR combustion experiment.

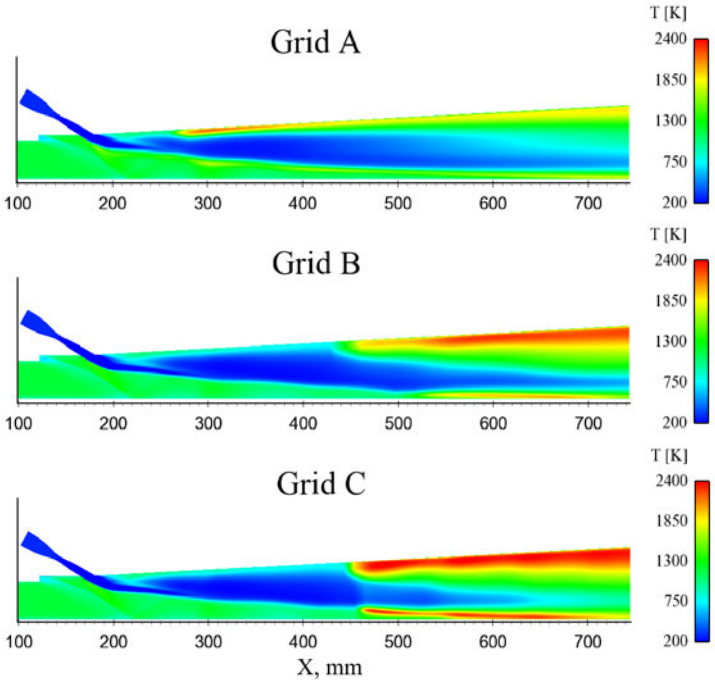
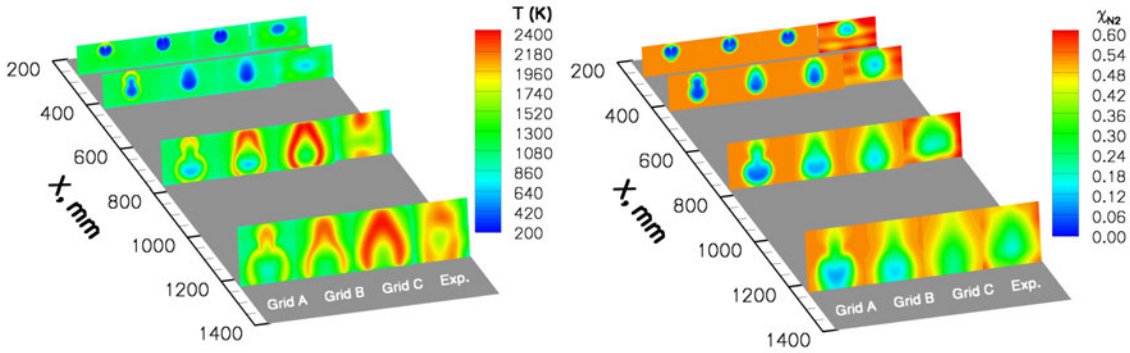


Figure 4. Temperature grid sensitivity of the SCHOLAR combustion experiment.

Mixing and combustion cases were considered for all three grids using SFM. Wall pressure comparisons to data for all three grids for the combustion case are shown in Figure 3. Note that experimental error bars are not available. Substantial improvement downstream of x = 400mm is noted when going from Grid A to Grid C. Grid A was very coarse stream-wise in this section due to the long combustor, which caused a much more diffusive flow-field than that of more refined grids. The contours of mean temperature shown in Figure 4 indicate combustion in Grid A starting to take place well upstream of the other two grids. However, the amount of combustion taking place is much less, which coincides with the wall pressure result. The combustion onset location along the top wall for Grids B and C are in similar locations, providing some consistency in the results. Figure 5 shows mean temperature and N_2 mole fraction contours compared to measured data for all three grids. As in the mixing case, the core jet mixing is under-predicted, even with the substantial increase in the number of grid points. More combustion appears to be taking place on planes 3 and 4 of Grid C compared to the experiment, but the overall contours provide a better match to data than coarser grids. These results clearly reveal an improved match to data with grid refinement going from Grid A to Grid C.



a) Mean Temperature b) N_2 Mole Fraction

Figure 5. Grid-sensitivity comparison of the SCHOLAR combustion experiment.

1.3. Conventional Approach to Solution Verification

Richardson extrapolation has typically been used as a means of verifying grid convergence in numerical simulations. This technique has traditionally been applied in systematic refinement of structured meshes, where refinement ratios are typically constant. Richardson extrapolation is an effective means of performing formal solution verification and error assessment, wherein results may be extrapolated in the limit of local grid spacing going to zero. Such quantities may include global values such as aerodynamic coefficients, or local solution variables.

The basic premise of Richardson extrapolation is that the local error is assumed to scale with the local mesh width h for each of three meshes considered: h_1, h_2, and h_3.

$$F \approx F_i\left(h_i\right) + Ch_i^p + O\left(h_i^{p+1}\right) \tag{2}$$

For a uniform refinement ratio $r = h_1/h_2 = h_2/h_3$, the order of convergence p is

$$p = \log\left(\frac{F_2 - F_1}{F_3 - F_2}\right) \Big/ \log r \tag{3}$$

Once this is known, the extrapolated result becomes

$$F_{extrap} = F_3 + \frac{F_3 - F_2}{\left(h_2 / h_3\right)^p - 1} \tag{4}$$

Richardson extrapolation has been extended to more general cases, and demonstrated for unstructured meshes that may be either uniformly or adaptively refined [21, 22]. In the work by Baker [23], for example, the mesh width *h* for each grid is defined by the number of vertices.

$$h_1 = \left(1/N_1\right)^{1/3}, \quad h_2 = \left(1/N_2\right)^{1/3}, \quad h_3 = \left(1/N_3\right)^{1/3} \tag{5}$$

For unstructured meshes, the relationship between successive mesh widths is generally not described by a constant refinement ratio *r*, and for such non-uniform refinements, the order of convergence *p* must be solved iteratively.

$$\left(F_3 - F_2\right)h_1^p + \left(F_1 - F_3\right)h_2^p + \left(F_2 - F_1\right)h_3^p = 0 \tag{6}$$

At this point, it is important to note a significant restriction. To apply Richardson extrapolation, the solution must exhibit monotone behavior. That is,

$$F_1 > F_2 > F_3 \quad or \quad F_1 < F_2 < F_3 \tag{7}$$

If the solution is oscillatory on the mesh sequence, Richardson extrapolation is not valid, *even if the incremental change in the solution is small*. Coleman et al. [24] have further noted that monotonic behavior may be difficult to obtain if the solution behavior with grid refinement is truly oscillatory. Oscillatory convergence is an issue encountered in practical CFD simulation, and researchers have attempted to address the phenomenon.

Baker demonstrates that the order of convergence may yield very useful information on the sequence of solutions. Specifically, he argued that *p* might be larger than the formal order of accuracy of the solver, if point placement on the refined meshes is strategic and selective, as with adaptive meshes. This is due to the fact that adaptive schemes aim to reduce error locally, equally distributing the error, whereas uniform mesh refinements scale the error uniformly in the domain. In contrast, Baker also noted that suspect values for *p* that are markedly lower than the solver's formal order of accuracy may indicate a problem with the mesh sequence and/or the solutions. Examples may include incomplete iterative convergence or dissipative limiters. It is also possible that the sequence of meshes is not yet in the asymptotic range where Richardson extrapolation is valid, which may occur if mesh refinement introduces newly resolved features that were not present on coarser grids.

1.4. The Problem With Richardson Extrapolation

In practice, grid convergence is often argued by inspection, which shows that little change in important variables is observed as the mesh is refined. Richardson extrapolation, long advocated by members of the V&V community as a means of verifying grid convergence in numerical simulations [25-30], requires a sequence of at least three meshes and monotonic solution behavior. While effective, potential difficulties encountered in mesh generation and solution oscillation on the grid sequence (even if the changes are small) may prevent the use of this technique and require more than three meshes. In extreme cases, some advocates of this approach

have demonstrated that it may take upwards of 10 grids to attain the asymptotic range where Richardson extrapolation is valid [25], which renders the technique impractical. Furthermore, reports of grid convergence index (GCI) [27] in conference papers and journal articles are quite rare, indicating that such a practice has not gained popular acceptance outside of the V&V community itself.

An informal review of available literature, summarized in Table 2, shows that Richardson extrapolation is presented in 1% of peer-reviewed journal articles relevant to the propulsion community, despite editorial policy. Just over one third of the articles surveyed included any solution verification. Upon closer inspection, several of those only consider "coarse and fine" grids to satisfy the editorial policy requirement. Often, the refinement study is performed on a unit problem and not the actual problem of interest. Many times the grid refinement study is referenced but not presented in the article itself. Simply put, the task of verification is perceived as a distraction from the study of interest, and in numerous practical and complex simulations it will not be done. Verification will only be performed if (a) it is absolutely, strictly required, and (b) it is practical to do so. Thus it remains true that "The more common, but not rigorous, meaning of grid convergence is that little change in important dependent variables is demonstrated during grid and time step refinement" [31].

Table 2. Survey of CFD verification studies from various journals, 2010-2012.

Journal	Issues	Date Range	CFD Papers	Verification Performed	3 Grid Refinement Study	Richardson Extrapolation Used
AIAA Journal	24	7/2010-6/2012	117	49	29	1
Journal of Aircraft	9	7/2010-12/2011	48	13	10	1
Journal of Fluids Engineering	6	7/2010-12/2010	9	4	4	0
Journal of Propulsion and Power	9	7/2010-12/2011	39	15	9	0
Journal of Spacecraft and Rockets	9	7/2010-12/2011	19	6	4	1
TOTALS	**57**		**232**	**87 (37.50%)**	**56 (24.14%)**	**3 (1.29%)**

1.5. Basic Premise of Error Transport

With this motivation, the concept of an error transport equation (ETE) has received attention by a number of researchers as a means of quantifying the spatial discretization error for any given grid and solution [32-47]. The basic premise of the ETE is that solution errors may be generated in one location of the domain and transferred elsewhere as erroneous waves propagated by the characteristics of the Euler/Navier-Stokes equations, where they become apparent and manifest as interpolation errors.

If the increments provided by grid refinement can be reliably predicted, a higher level of confidence may be placed in lower fidelity meshes. This would potentially preclude the need for fine grid analysis if the numerical errors predicted for a medium resolution mesh are judged to be

acceptable. In addition, if discretization errors are bounded and experimental data lies outside numerical error bars, it would likely indicate a deficiency in physical modeling, as one source of error has been isolated. Such information would be of considerable benefit to the CFD analyst in evaluating simulation results.

1.6. Scope of This Chapter

This chapter describes our approach for solution verification and error reduction in CFD simulations. Using this approach, a quantitative analysis of discretization errors is possible for any given grid and solution. The software system also contains a mesh adaptation capability for three-dimensional mixed-element unstructured grids that can key local grid refinement to the location of error sources identified by the ETE solver. The complete CRISP CFD package is demonstrated on realistic applications of relevance to the propulsion community, and shown to be a distinct solution verification capability. It should be noted that, in cases where experimental data are included for comparison, no experimental error bars are available. While experimental error bars are truly needed to validate the results, the emphasis in the examples that follow is on verification, not validation.

2. Review of Error Transport Methodology

2.1. Early Research in ETE Methods

Over the past several years, the concept of an error transport equation has received attention by a number of researchers as a means of quantifying the spatial discretization error in a numerical simulation. Major investigations of ETE models in 1D and 2D have been made by Qin and Shih [32-36], Celik et al. [37, 38], and Zhang et al. [40-43]. Van Straalen et al. [44], Hay and Visonneau [45], and Wilson and Stern [46] have also explored the concept, and the reader is encouraged to review these varied efforts. Most recently, Tong and Luke demonstrated an inviscid ETE [47].

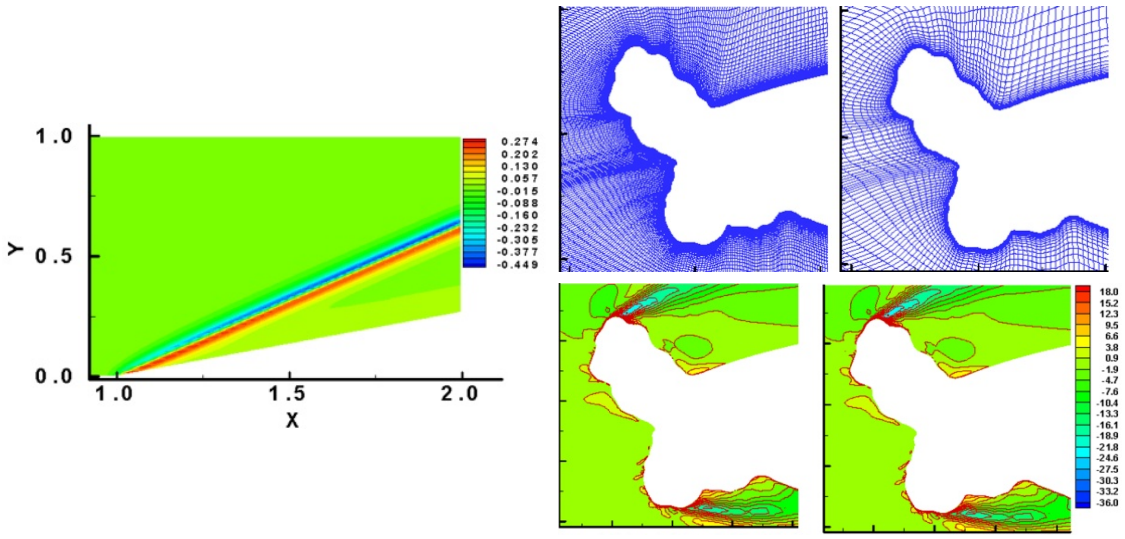
The basic premise of the error transport equation is that solution errors may be generated in one location of the domain and transferred elsewhere, where they become apparent and manifest as interpolation errors. These errors may travel as erroneous waves, which are propagated by the characteristics of the Euler/Navier-Stokes equations. The ETE has also been developed as an alternative to traditional Richardson extrapolation, which requires at least three meshes to estimate a solution error if the order of convergence is not assumed. Ideally, the ETE should provide a quantitatively accurate assessment of the spatial error for any given grid and solution. The primary challenge lies in formulating a suitable source term or residual for the ETE.

Each research group investigating ETE methods has taken the same fundamental approach, with varying degrees of success and application. The error equation is derived from the original PDE by recognizing that the finite volume approximation creates a residual when substituted into the PDE for a given discretized grid. The leading terms of the modified equation are used to model the residual or source term in the ETE. The modified equation is the PDE that is actually solved when a finite difference or finite volume method is applied. This equation represents the discrete equation whereas the PDE does not. Derivation of the modified equation from Taylor series expansions is detailed in most introductory CFD texts, such as Reference 48. The right hand side of the modified equation is the truncation error, the difference between the original PDE and the discretized equation. Examination of these terms yields the dominant error terms resulting from

the spatial discretization of the mesh. These terms form the residual or source term in the error transport equation.

Qin and Shih et al. [32-36] have made several attempts to model the residual using the modified equation. However in most of their applications, such as the oblique shock and iced airfoil shown in Figure 6, they actually compute the grid-independent solution and determine the residual for the ETE from that. While this is useful as verification of the concept, requiring a grid-independent solution *a priori* is of little practical value. Their attempts at modeling the residual when a grid-independent solution is unknown have generally remained limited to simple 1D model equations. Celik and Hu [37, 38] implemented their ETE on uniform Cartesian structured meshes, in the context of a SIMPLE-type solver with influence coefficients and N/S/E/W notation. Their approach requires solution derivatives up to fourth order in space. While this is readily done on uniform structured meshes, such higher derivatives are not readily obtained on unstructured grids that are so prevalent in current Navier-Stokes analyses. Demonstration cases included 1D and 2D convection/diffusion, 2D Poisson equation, and 1D Burgers equation, with reasonable results.

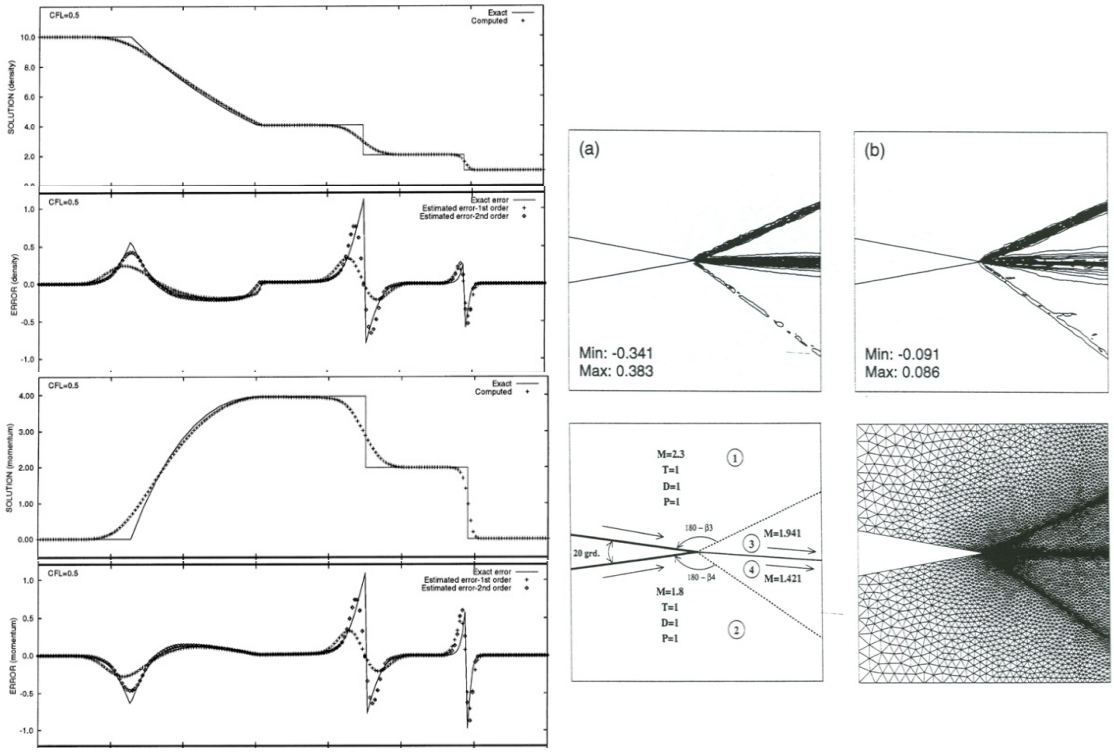
The work of Zhang et al. [40-43] appeared to be the most promising for practical application, as their work was developed specifically for the Roe flux difference scheme. The Roe scheme is employed in numerous production level Navier-Stokes solvers developed by NASA, the U.S. Department of Defense, and private industry, which are used extensively by the propulsion community. Their predicted error distributions were in excellent agreement with exact error distributions for the problems considered. In addition, their results were not limited to 1D model equations, and demonstrated error prediction for finite volume solutions of the 2D Euler equations on unstructured triangular grids, as seen in Figure 7. Mesh adaptation was also demonstrated in 1D and 2D with refinement keyed to the error source, and not the interpolation error as is the usual convention [41, 43].



a) Density error for oblique shock

b) Iced airfoil, error in u momentum from ETE (left) and difference between fine and coarse grid solutions (right).

Figure 6. ETE solutions of Qin and Shih [32, 33].



a) Transient 1D shock tube a) 2D supersonic confluence

Figure 7. ETE solutions of Zhang et al. [42, 43].

2.2. Works of Cavallo, et al.

Recent works by the author [49-53] have brought error transport methods into the realm of practical application. In these works, error quantification for aerodynamic flows in 2D and 3D using an inviscid ETE was first demonstrated, comparing predicted error bars in both local and integrated variables with the results of Richardson extrapolation and test data. The inviscid ETE was then extended to treat turbulent flows involving shear layers and boundary layers. Vital error transport and generation mechanisms neglected by other works in this field, namely viscous diffusion and production due to errors in turbulent viscosity, were developed and shown to be essential for predicting errors in mixing-dominated flows. More appropriate boundary conditions for the error at viscous walls were also included. This research demonstrated considerable promise in the ability of the method to reliably predict the magnitude of the spatial discretization error for several applications, with coarse grid error bars consistently containing both fine grid results and the results of conventional Richardson extrapolation.

A complete error analysis package was developed, in which the ETE solver is but one piece of the overall system. A complete library of functions for errors in numerous local and integrated quantities of interest is available as part of the software. Techniques for simultaneously visualizing a solution and its associated errors are included. The rendering of errors for 2D and 3D scalar and vector fields provides a more complete analysis than is possible with conventional error bars commonly found in 1D profile plots. Tools for performing conventional Richardson extrapolation, obtaining global error metrics and reporting error histograms are available to the user. Finally, the reductions of spatial discretization errors may be performed using a 3D mixed-element unstructured mesh adaptation capability for local coarsening and refinement. This set of

tools forms a complete software system for the prediction, visualization, analysis, and reduction of discretization errors in CFD simulations. Using adaptive mesh refinement in conjunction with the ETE, certifiable error reductions were obtained.

Most recently, the error transport method was extended to a unified framework suitable for use with simulations conducted using cell-centered or node-centered finite volume unstructured Navier-Stokes solvers. Support for the CFD General Notification System (CGNS) standard [54] has been implemented to permit use of the method by a broad spectrum of potential flow solvers. Several applications are presented to demonstrate the solver's utility as a practical and effective solution verification tool for a variety of turbulence models and CFD codes. The error transport method is shown to be general and of greater utility than conventional Richardson extrapolation techniques.

3. Error Prediction Methodology

3.1. Formulation of Error Transport Equation (ETE)

The error transport equation is derived from the governing equations by assuming that the discrete finite volume solution introduces a residual $\vec{R}$ that represents the amount by which the governing partial differential equation is not satisfied. Starting with the Navier-Stokes equations for a turbulent, real gas mixture in integral form, we subtract from it the approximate finite volume solution on a mesh of width h to obtain the ETE.

$$\begin{aligned} \frac{\partial}{\partial t}\iiint\left(\vec{Q}-\vec{Q}^h\right)dV+\iint\left(\left(\vec{F}\left(\vec{Q}\right)-\vec{F}\left(\vec{Q}^h\right)\right)\cdot\hat{n}\right)dA= \\ \iint\left(\left(\vec{G}\left(\vec{Q}\right)-\vec{G}\left(\vec{Q}^h\right)\right)\cdot\hat{n}\right)dA+\iiint\left(\vec{D}\left(\vec{Q}\right)-\vec{D}\left(\vec{Q}^h\right)\right)dV-\iiint\vec{R}\left(\vec{Q}^h\right)dV \end{aligned} \tag{8}$$

Here $\vec{Q}$ is the vector of conserved variables defined in Eqn. 8 and $\vec{\varepsilon}$ is the error vector, or the solution vector for the ETE, both of which include the transport equations for any chemical species and the turbulence model employed, such as the k-epsilon (k-ε) model.

$$\vec{\varepsilon}=\vec{Q}-\vec{Q}^h=\left[\begin{array}{ccccccccccc}\varepsilon_\rho & \varepsilon_{\rho u} & \varepsilon_{\rho v} & \varepsilon_{\rho w} & \varepsilon_{\rho e_0} & \varepsilon_{\rho_1} & \cdots & \varepsilon_{\rho_{NS-1}} & \varepsilon_{\rho k} & \varepsilon_{\rho\varepsilon}\end{array}\right]^T \tag{9}$$

This error transport equation is expanded from that reported in the initial works by the author to include additional transport equations for the error in the partial density of each species comprising the mixture, error source terms due to the predicted error in turbulent mass diffusivity, which is computed from error in turbulent viscosity.

The inviscid portion of the residual required in Eqn. 8 is formed from the upwind contribution of the inviscid flux at each face of the control volume, which is derived using the Modified Equation approach [41, 50]. Following the work of Zhang et al. [40, 41], the flux at the face is computed from the flux evaluated at the states to the left and right of the face, plus the upwind contribution.

$$\vec{F}\left(\vec{Q}\right)=\tfrac{1}{2}\left(\vec{F}\left(\vec{Q}^L\right)+\vec{F}\left(\vec{Q}^R\right)\right)+\tfrac{1}{2}\left(\Delta\vec{F}\right) \tag{10}$$

Note that for a preconditioned system, the upwind flux would also include the preconditioning matrix. Starting with a simple explicit finite volume Roe scheme,

$$u_i^{h,n+1} = u_i^{h,n} - \frac{\Delta t}{\Delta x}\left(f_{i+1/2}^{h,n} - f_{i-1/2}^{h,n}\right) \tag{11}$$

$$u_i^{h,n+1} = u_i^{h,n} - \frac{\Delta t}{\Delta x}\left(\tfrac{1}{2}\left(f_{i+1}^{h,n} + f_i^{h,n}\right) - \tfrac{1}{2}\left|\Delta f_{i+1/2}^{h,n}\right| - \tfrac{1}{2}\left(f_i^{h,n} + f_{i-1}^{h,n}\right) + \tfrac{1}{2}\left|\Delta f_{i-1/2}^{h,n}\right|\right) \tag{12}$$

we expand $u_i^{h,n+1}$, $f_{i+1}^{h,n}$, and $f_{i-1}^{h,n}$ via Taylor series. Derivation of the modified equation is straightforward and becomes

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = \frac{1}{2\Delta x}\left(\left|\Delta f_{i+1/2}^{h,n}\right| - \left|\Delta f_{i-1/2}^{h,n}\right|\right) - \frac{\Delta t}{2}\frac{\partial^2 u}{\partial t^2} - \frac{\Delta x^2}{6}\frac{\partial^3 f}{\partial x^3} - \frac{\Delta t^2}{6}\frac{\partial^3 u}{\partial t^3} + O\left(\Delta x^3, \Delta t^3\right) \tag{13}$$

the leading terms of which are simply the upwind terms of the Roe flux at the cell faces.

$$R = \frac{1}{2\Delta x}\left(\left|\Delta f_{i+1/2}^{h,n}\right| - \left|\Delta f_{i-1/2}^{h,n}\right|\right) \tag{14}$$

Zhang et al. also noted more complex expressions for the residual may be formed by expanding the upwind terms in Eqn. 12 as well. Furthermore, in transient applications, where the ETE is solved concurrent with the Navier-Stokes equations, additional terms appear in the residual. For a finite volume scheme, the inviscid contribution to the residual at node *i* is formed from the upwind contribution from each of the edges incident to the node.

$$\vec{R}_{INV} = -\frac{1}{2}\iint\left(\Delta\vec{F}\cdot\hat{n}\right)dA \tag{15}$$

It should be noted that although Zhang's approach was derived for a first order upwind scheme, the approach is directly applicable to a second-order MUSCL implementation of the Roe flux in which the states are reconstructed at each side of the face using gradient extrapolation from the neighboring control volumes. All applications presented were obtained using a second-order MUSCL scheme with Total Variation Diminishing (TVD) limiting.

The modified equation approach for the viscous terms results in high order derivatives that are difficult to form for practical calculations on unstructured meshes, which are emphasized in the applications that follow. As an alternative model, error in the viscous flux may be broken into two components: error in the velocity components u_i and temperature *T*, assuming μ_t is correct; and error in μ_t, assuming the velocity components u_i and temperature *T* are correct. Extending this model to real gas systems, we assume that the error in the viscous flux for the species transport equations may be represented in a similar manner: error in the species mass fractions y_i, assuming $\mathcal{D}_t$ is correct; and error in $\mathcal{D}_t$, assuming the mass fractions y_i are correct. The difference in the viscous fluxes in Eqn. 8 may then be written as follows:

$$\left(\vec{G}\left(\bar{Q}\right)-\vec{G}\left(\bar{Q}^{h}\right)\right)\bullet\hat{n}=\begin{bmatrix}0\\ n_x\left(\tau_{xx}-\tau_{xx}^{h}\right)+n_y\left(\tau_{xy}-\tau_{xy}^{h}\right)+n_z\left(\tau_{xz}-\tau_{xz}^{h}\right)\\ n_x\left(\tau_{yx}-\tau_{yx}^{h}\right)+n_y\left(\tau_{yy}-\tau_{yy}^{h}\right)+n_z\left(\tau_{yz}-\tau_{yz}^{h}\right)\\ n_x\left(\tau_{zx}-\tau_{zx}^{h}\right)+n_y\left(\tau_{zy}-\tau_{zy}^{h}\right)+n_z\left(\tau_{zz}-\tau_{zz}^{h}\right)\\ n_x\left(e_x-e_x^{h}\right)+n_y\left(e_y-e_y^{h}\right)+n_z\left(e_z-e_z^{h}\right)\\ n_x\left(q_x^{1}-q_x^{1^h}\right)+n_y\left(q_y^{1}-q_y^{1^h}\right)+n_z\left(q_z^{1}-q_z^{1^h}\right)\\ \vdots\\ n_x\left(q_x^{NS-1}-q_x^{NS-1^h}\right)+n_y\left(q_y^{NS-1}-q_y^{NS-1^h}\right)+n_z\left(q_z^{NS-1}-q_z^{NS-1^h}\right)\\ n_x\left(q_x^{k}-q_x^{k^h}\right)+n_y\left(q_y^{k}-q_y^{k^h}\right)+n_z\left(q_z^{k}-q_z^{k^h}\right)\\ n_x\left(q_x^{\varepsilon}-q_x^{\varepsilon^h}\right)+n_y\left(q_y^{\varepsilon}-q_y^{\varepsilon^h}\right)+n_z\left(q_z^{\varepsilon}-q_z^{\varepsilon^h}\right)\end{bmatrix} \tag{16}$$

$$\tau_{ij}-\tau_{ij}^{h}=\left(\mu+\mu_t\right)\left(\frac{\partial\varepsilon_{u_i}}{\partial x_j}+\frac{\partial\varepsilon_{u_j}}{\partial x_i}\right)+\varepsilon_{\mu_t}\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right) \tag{17}$$

$$\begin{aligned}e_i-e_i^{h}=&\ \varepsilon_{u_j}\left(\mu+\mu_t\right)\left(\frac{\partial\varepsilon_{u_j}}{\partial x_j}+\frac{\partial\varepsilon_{u_j}}{\partial x_i}\right)+u_j\varepsilon_{\mu_t}\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)+\left(k+k_t\right)\frac{\partial\varepsilon_T}{\partial x_i}+\varepsilon_{k_t}\frac{\partial T}{\partial x_i}\\ &+\sum_{j=1}^{NS-1}\rho\left(h_j-h_{NS}\right)\left(\left(\mathcal{D}+\mathcal{D}_t\right)\frac{\partial\varepsilon_{y_j}}{\partial x_i}+\varepsilon_{\mathcal{D}_t}\frac{\partial y_j}{\partial x_i}\right)\end{aligned} \tag{18}$$

$$q_i^{N}-q_i^{N^h}=\rho\left(\mathcal{D}+\mathcal{D}_t\right)\left(\frac{\partial\varepsilon_{y_N}}{\partial x_i}\right)+\varepsilon_{\mathcal{D}_t}\left(\frac{\partial y_N}{\partial x_i}\right) \tag{19}$$

$$q_i^{k}-q_i^{k^h}=\left(\mu+\frac{\mu_t}{\sigma_k}\right)\left(\frac{\partial\varepsilon_k}{\partial x_i}\right)+\varepsilon_{\mu_t}\left(\frac{\partial k}{\partial x_i}\right) \tag{20}$$

$$q_i^{\varepsilon}-q_i^{\varepsilon^h}=\left(\mu+\frac{\mu_t}{\sigma_\varepsilon}\right)\left(\frac{\partial\varepsilon_\varepsilon}{\partial x_i}\right)+\varepsilon_{\mu_t}\left(\frac{\partial\varepsilon}{\partial x_i}\right) \tag{21}$$

This leads to a diffusion term for the error and a production term or residual due to the predicted error in turbulent viscosity, thermal conductivity, and mass diffusivity.

$$\left(\vec{G}\left(\bar{Q}\right)-\vec{G}\left(\bar{Q}^{h}\right)\right)\bullet\hat{n}=\vec{G}\left(\vec{\varepsilon}\right)\bullet\hat{n}-\vec{R}_{TURB} \tag{22}$$

Equation 22 provides for a suitable diffusion mechanism for the error, as well as a production term, due to predicted errors in turbulent viscosity, conductivity, and diffusivity. These mechanisms were found to be essential in prior studies of boundary layer and shear layer flows [51], as inaccurate prediction of turbulent mixing is known to affect mean quantities. Thus the model is able to treat portions of the domain that are dominated by diffusion versus convection.

The error in turbulent viscosity is derived from the predicted errors in the turbulence model variables, from which the errors in turbulent conductivity and diffusivity follow given the turbulent Prandtl and Schmidt numbers. The error in turbulent viscosity for the *k*-ε model is given in Eqn. 23. In addition to *k*-ε, the ETE solver in CRISP CFD supports solutions obtained using the Spalart-Allmaras and *k*-ω turbulence models [53].

$$\varepsilon_{\mu_t} = \mu_t \left(\frac{2}{\rho k} \varepsilon_{\rho k} - \frac{1}{\rho \varepsilon} \varepsilon_{\rho \varepsilon} \right) \qquad \varepsilon_{k_t} = \frac{c_p \varepsilon_{\mu_t}}{\mathrm{Pr}_t} = k_t \frac{\varepsilon_{\mu_t}}{\mu_t} \qquad \varepsilon_{\mathcal{D}_i} = \frac{\varepsilon_{\mu_t}}{\rho \mathrm{Sc}_t} = \mathcal{D}_t \frac{\varepsilon_{\mu_t}}{\mu_t} \tag{23}$$

The final viscous ETE is therefore

$$\boxed{\frac{\partial}{\partial t} \iiint \vec{\varepsilon} \, dV + \iint \left(\left(A\left(\vec{Q}^h\right) \vec{\varepsilon} \right) \cdot \hat{n} \right) dA = \iint \left(\vec{G}\left(\vec{\varepsilon}\right) \cdot \hat{n} \right) dA - \vec{R}_{INV} - \vec{R}_{TURB}} \tag{24}$$

It should be noted that errors in the chemical and turbulence model source term $\vec{D}\left(\vec{Q}\right)$ are currently neglected from the error transport model. In addition, Pr_t and Sc_t are taken to be constant.

3.2. Turbulence Modeling Treatments

The error in turbulent viscosity is derived from the predicted errors in the turbulence model variables, from which the errors in turbulent conductivity and diffusivity follow given the turbulent Prandtl and Schmidt numbers. The turbulent production term in Eqn. 24 provides an error generation mechanism that accounts for error in momentum and energy transfer due to over- or under-predicted mixing. An error tensor is computed in a manner analogous to the stress tensor, in which the error in eddy viscosity appears in place of the effective viscosity.

$$\varepsilon_{\tau_{ij}} = \varepsilon_{\mu_t} \left(\frac{\partial u_i^h}{\partial x_j} + \frac{\partial u_j^h}{\partial x_i} \right) \qquad \varepsilon_{e_i} = u_j \varepsilon_{\tau_{ij}} + \varepsilon_{k_t} \frac{\partial T}{\partial x_i} \tag{25}$$

Given a prediction for ε_{μ_t}, the error in turbulent thermal conductivity may be obtained assuming a constant turbulent Prandtl number. For details on the turbulent residual the reader is referred to Ref. 51.

The formation of ε_{μ_t} itself is dependent on the turbulence model employed. To date we have implemented the one-equation Spalart-Allmaras model and the two-equation *k*-ε and *k*-ω models in the ETE solver. For each, the transport equations for the relevant turbulence quantities are included in Eqn. 24. Table 3 summarizes the error formulation for ε_{μ_t} for each model. The relations for ε_{μ_t} are derived using procedures described in the next section.

Table 3. Eddy viscosity errors for selected turbulence models.

Model	Eddy Viscosity	Error in Eddy Viscosity
Spalart-Allmaras	$\mu_t = \rho\bar{v}\left(\frac{\chi^3}{\chi^3 + c_{v1}^3}\right)$	$\varepsilon_{\mu_t} = \mu_t\left(\frac{1}{\rho}\varepsilon_\rho - \left(\frac{4c_{v1}^3}{\chi^3}+1\right)\frac{f_{v1}}{\bar{v}}\varepsilon_{\bar{v}}\right)$
$k-\varepsilon$	$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon}$	$\varepsilon_{\mu_t} = \mu_t\left(\frac{1}{\rho k}\varepsilon_{\rho k} - \frac{1}{\rho\varepsilon}\varepsilon_{\rho\varepsilon}\right)$
$k-\omega$	$\mu_t = \rho\frac{k}{\omega}$	$\varepsilon_{\mu_t} = \mu_t\left(\frac{1}{\rho}\varepsilon_\rho + \frac{1}{\rho k}\varepsilon_{\rho k} - \frac{1}{\rho\omega}\varepsilon_{\rho\omega}\right)$

3.3. Error Functions and Derived Variables

It is important to recognize that the errors predicted by the error transport equation are for the individual elements of the solver's solution vector $\bar{Q}$. Errors in all other quantities of interest are not obtained directly from the solution of the ETE. Rather, these are computed by propagating the errors contributing to the derived quantity. An approach for computing error propagation is taken from uncertainty propagation methods used in experimental data reduction [55, 56]. For a given result $R = R\left(x_1, x_2, \cdots, x_n\right)$, the incremental change in R due to errors in x_i is the sum of the contributions from the individual errors.

$$\varepsilon_R = \pm\sum\left(\frac{\partial R}{\partial x_i}\varepsilon_{x_i}\right) \tag{26}$$

Errors in all quantities of interest, including integrated quantities, are derived from the error vector using this approach. As an example, for the error in pressure resulting from predicted errors in the conserved variables $\bar{Q}$, it may be shown that

$$\bar{Q} = \begin{bmatrix} \rho & \rho u & \rho v & \rho w & \rho e_0 \end{bmatrix}^T \tag{27}$$

$$P = \left(\gamma - 1\right)\left[Q_5 - \frac{\left(Q_2^2 + Q_3^2 + Q_4^2\right)}{2Q_1}\right] \tag{28}$$

$$\varepsilon_P = \pm\left(\gamma - 1\right)\left(\left(\frac{\left|\vec{V}\right|^2}{2}\varepsilon_1\right) - \left(u\varepsilon_2\right) - \left(v\varepsilon_3\right) - \left(w\varepsilon_4\right) + \left(\varepsilon_5\right)\right) \tag{29}$$

Error in pressure coefficient may then be obtained.

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty V_\infty^2} \tag{30}$$

$$\varepsilon_{C_p} = \frac{\partial C_p}{\partial p}\varepsilon_p \text{, or } \varepsilon_{C_p} = \pm\frac{\varepsilon_P}{\frac{1}{2}\rho_\infty V_\infty^2} \tag{31}$$

Error functions derived for ideal gases assume constant specific heats and specific heat ratio. In multi-component mixtures, the specific heat is calculated through temperature dependent curve fits (i.e., JANNAF tables), and enthalpy is the integral of specific heat with respect to temperature. The errors in the species partial densities also appear in the ETE solution vector, which may be used to calculate errors in the mixture specific heat, specific heat ratio and gas constant. These new variables and their associated errors must also be considered to extend the error function expressions derived using an ideal gas model. The corresponding real gas formulations account for variable specific heats and variable gas constant, according to the local mixture composition. In our approach for real gas calculations, we assume that the sum of the errors in the partial densities equals the sum of the error in the mixture density. Additional error functions for the computation of errors in species mass and mole fractions, as well as mixture gas constant and specific heat ratio, are available for real gas simulations.

Table 4 lists all of the local and globally integrated quantities that have been derived and are available in the Error Function Library. This set of 64 error functions includes many of the standard PLOT3D functions [57], along with reports for globally integrated quantities, such as force and moment coefficients and mass flow rates across boundaries. Users may compute any number of functions during post-processing by selecting the function names desired. For each error function computed, a histogram is provided detailing the distribution of error in the selected variable among the vertices in the mesh, along with a measurement of the global error in the selected quantity. Users may compute any number of functions during post-processing by selecting the function names desired. For each error function computed, a histogram is provided detailing the distribution of error in the selected variable among the vertices in the mesh, along with a measurement of the global error in the selected quantity. Global error in a variable is defined using a volume weighted average.

$$\overline{\varepsilon}_g = \sum_{i=1}^{Nverts} \varepsilon_i V_i \Big/ \sum_{i=1}^{Nverts} V_i \tag{32}$$

Note the denominator in Eq. 8 is the volume of the computational domain. This provides an average error for the entire domain for a given variable that is not affected by the number of vertices in the mesh.

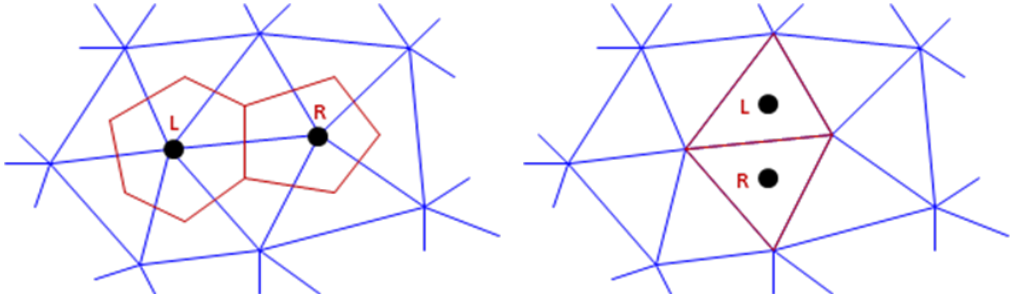
Table 4. Available error functions in the Error Function Library.

Density	**Pressure**		**Temperature**	**Enthalpy, Entropy**	**Energy**	**Chemistry**
rho	P	pitot	T	h	e	gamma
norm(rho)	norm(P)	pitotratio	norm(T)	norm(h)	norm(e)	rgas
log(norm(rho))	log(norm(P))	q	log(norm(T))	h0	e0	yjet
rho0	P0	Cp	T0	norm(h0)	norm(e0)	yext
norm(rh0)	norm(P0)	Cp0	norm(T0)	s	rhoe0	sp-NN
					ke	x-NN
					norm(ke)	y-NN
Momentum	**Velocity**		**Turbulence**	**Flow Rate**	**Aerodynamics**	
rhou	u	a	k	mdot	forces	
rhov	v	M	epsilon	vdot	moments	
rhow	w	crossflow	mut		CLCD	
momentum	vel	vprime	mul		tauwall	
	mag(vel)		SA		Cf	

3.4. Discretization Considerations

The discretization of the ETE should match that of the flow solver as closely as possible to provide a truly accurate representation of the error for a given grid. On a tetrahedral mesh, for example, the number of cells is 5-6 times the number of vertices, and analyzing a cell-centered CFD solution using a node-centered discretization for the ETE is inappropriate since a coarser grid is being employed. As such, both node-centered (NC) and cell-centered (CC) arrangements should be supported. Figure 8 illustrates the fundamental differences between the two schemes. While both approaches employ a finite volume formulation, the actual definition of the control volumes and control surfaces differ. In the node-centered arrangement, a dual volume is constructed around each vertex from the contributions of all incident cells, and the solution is stored and updated at the vertices. Fluxes are computed for each face of the control volume, which is associated with an edge in the grid connecting two vertices, and evaluated at the midpoint of the edge. In contrast, the cell-centered scheme uses the actual physical cells present in the grid as the control volumes, with the solution stored and updated at the cell centroids. Here fluxes are computed for the physical faces of the cells at the face centroid. In either case, once the state of the fluid is defined for the "left" and "right" side of the control volume face, the inviscid flux is independent of whether the discretization is node-centered or cell-centered.

The ETE solver in CRISP CFD is largely transparent to the origin of the solution being analyzed. During solver setup, the actual volumes, face normals and tangents, coordinates, etc. are formed based on a NC or CC arrangement. Upon entering the ETE solver itself, the meaning of "control volume adjacency" and other variables are defined. This is accomplished through assigning array pointers. The cell adjacency array, for example, is set to either the edge definitions, denoting a vertex adjacency as in Figure 8a, or the face-cell array, denoting the cell adjacency of Figure 8b. "Control volume coordinates" refers to either the vertex locations or the cell center locations. Once these assignments are done, the same general array names are used throughout the ETE solver.



a) Node-centered (NC) b) Cell-centered (CC)

Figure 8. Unstructured finite volume arrangements.

In the Error Function Library, most of the error functions are evaluated in a pointwise fashion, that is, a simple loop over all control volumes to process elements of the CFD solution vector and the ETE solution vector. Extending these functions to treat CC cases is straightforward, merely requiring loops over the number of vertices to instead be cast as loops over the number of control volumes. Integrated values such as aerodynamic coefficients and mass flow rates across a boundary require additional treatment. Specifically, these functions involve loops over boundary faces, and hence require data for cells adjacent to the boundaries. During initialization, pointers that identify the control volume face that corresponds to each boundary face are set up. This allows for access to various grid metrics associated with the control volume (CV) face, such as the face normal, face area, and cell number adjacent to the face. The viscous contribution to

errors in aerodynamic coefficients is based on the velocity gradient from the adjacent cell center to the boundary face, whereas in the NC arrangement, this is computed using a node normal to the wall.

4. Visualization and Interpretation of Errors

For 1D profiles, the errors predicted by the ETE may be included as conventional error bars. The predicted errors possess an algebraic sign, and may be presented on plots with only half an error bar (i.e., above or below the data point). Typically, when experimental uncertainty is presented, the convention is to apply the uncertainty both above and below the data point. While this represents a rather conservative 50% confidence interval [27, 28], this ± approach is employed in the applications that follow. However, the presentation of error bars is open to interpretation, and as ETE methods are utilized more frequently, this issue should be resolved.

To gain further understanding and insight, some form of post-processing is generally needed to compute derived quantities and extract features that are relevant to the current calculation. Visualization is often used to display higher-level extracts such as contours and isosurfaces for scalar data, or streamlines, vortex cores, and shock surfaces for vector data. Having now computed error values, it is important to ask what impact these errors have on the features of interest in a calculation. Simultaneous visualization of a variable and its associated error is needed to fully utilize the results of the ETE solver and Error Function Library.

The subject of uncertainty visualization has been investigated previously [58-60]. Several techniques have been explored for rendering the uncertainty in scalar and vector fields. Examples include geometry and attribute addition or alteration [58], vector glyphs [58-60], and envelopes [60]. In each of these prior works, the uncertainty is associated with measured data obtained through acquisition, as found in environmental studies, wind tunnel tests, and similar situations. While these methods do not appear to have been applied to simulated data (CFD) previously, such techniques could prove equally useful.

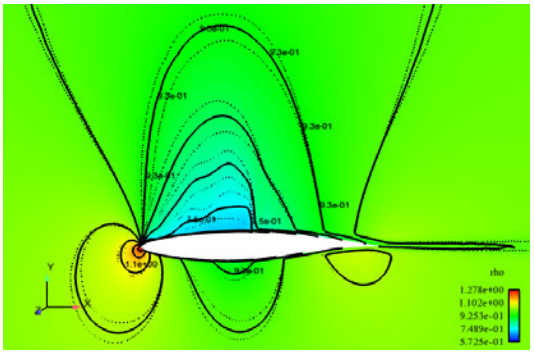
In previous work, we developed custom rendering techniques built on the EnSight framework [61] to give the user an environment where they can examine both the computed errors as well as the effect that these errors have on derived features and derived quantities. EnSight has a built-in calculator with over 100 functions which allows us to combine the solution variables and error variables in a variety of ways, as well as a full suite of visualization extracts that can be leveraged to better analyze and interpret the results of the ETE solver.

4.1. Techniques for Scalar Fields

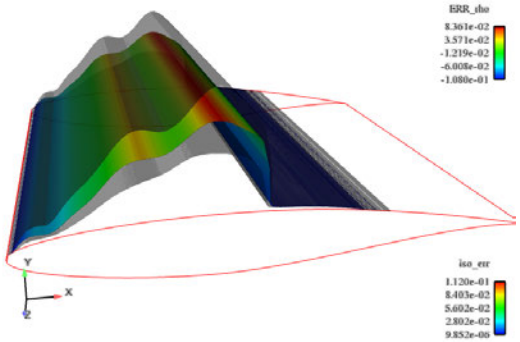
A common technique for visualizing scalar data is the extraction of isocontours (lines in 2D or surfaces in 3D). Precise positions of contour lines are very sensitive to the local gradient of the field. To illustrate the errors in a 2D scalar field with density ρ as the variable, we compute two new derived variables $\rho + \left|\varepsilon_\rho\right|$ and $\rho - \left|\varepsilon_\rho\right|$ using the EnSight variable calculator. We then compute contour lines in all three fields, using line styles and widths to graphically illustrate both the computed solution and the range of error. Figure 9a shows an example of this for the density field past the RAE2822 transonic airfoil. In essence, this method provides an envelope for each contour level. The contours associated with the errors $\rho + \left|\varepsilon_\rho\right|$ and $\rho - \left|\varepsilon_\rho\right|$ may be rendered using different attributes, such as dotted lines or other colors, so as not to detract from the primary

contours for density ρ itself. A disadvantage of this approach is that the plot may become cluttered if the contours overlap, or too many levels are displayed. When this method is used in 3D, the isocontour is a surface of constant value. We use a similar technique to compute three isosurfaces for illustration, as shown in Figure 9b. In this case, it can be more difficult to visually inspect the errors due to the 3D perspective and occlusion.

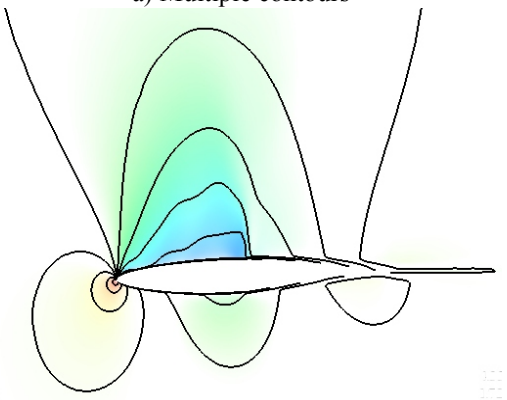
A second technique that has been explored involves changing the texture of a surface, as shown in Figure 9c. In this example, contours of density shown in black are supplemented by a flood contour of the density field, where the degree of transparency is a function of the local error in density. In regions where the predicted error is high, the flood contour is more opaque. Conversely, the colored scalar field disappears in regions of low error.



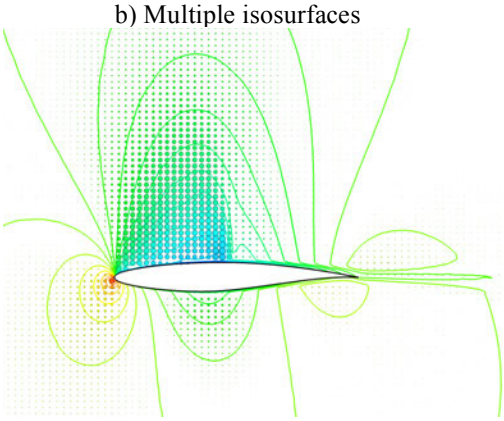
a) Multiple contours



b) Multiple isosurfaces



c) Textures with transparency defined by error



d) Error bubbles with glyph size defined by error

Figure 9. Error visualization methods for scalar fields.

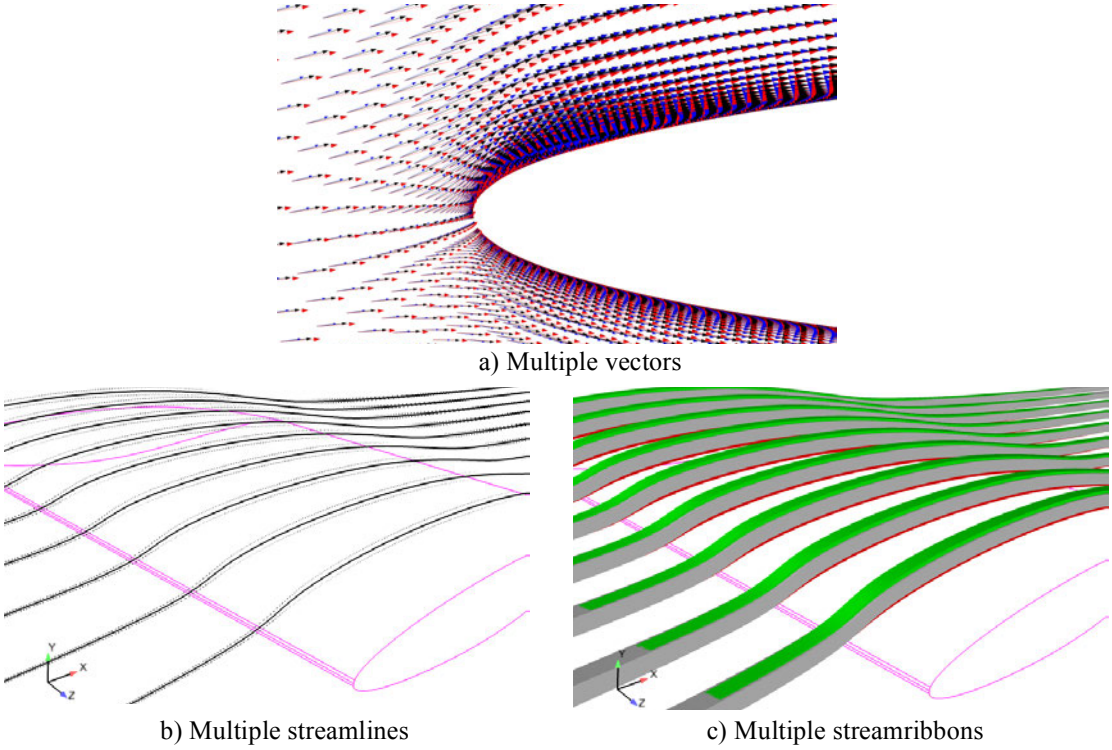
A third method considered for simultaneous rendering of a variable and its error is the bubble plot, illustrated in Figure 9d. This technique represents a boundary surface, clipping plane, or any other geometric entity or extract as a set of points. At each point, a spherical glyph is displayed whose radius varies with the predicted error. The glyphs themselves may be assigned a constant color or the color of the variable itself. Simultaneous presentation of contours or vectors and error bubbles can be quite effective in conveying the results obtained from the ETE solver and Error Function Library. This technique is applied in the case studies that follow.

4.2. Techniques for Vector Fields

In the prior works of Pang, Wittenbrink, and Lohda [58-60], uncertainties in vector fields were depicted by altering the attributes of the vector glyphs themselves, through widening, elongation,

or similar distortions. Our current approach is to display multiple vectors at each point, as in Figure 10a. In the limit of zero error, the primary vector shown in black and secondary vectors indicating error, represented here by blue and red arrows, will all coincide. Conversely, divergence of these vectors depicts the extent of the error in both magnitude and direction. As with multiple contours, a plot of vector envelopes like this may potentially become quite cluttered and difficult to interpret.

A common visualization technique for vector fields is the use of streamlines. Of particular concern for vector field error is the fact that errors can accumulate in the advection of a particle. Using similar techniques applied for scalar fields, we compute two new variables, $\vec{V} + \left|\varepsilon_{\vec{V}}\right|$ and $\vec{V} - \left|\varepsilon_{\vec{V}}\right|$, which capture the error in the velocity field and compute a set of streamlines from the same emission points. Methods similar to those employed in 2D contour plots may illustrate the range of errors by altering the streamline attributes, as depicted in Figure 10b. Figure 10c illustrates an alternate rendering wherein multiple streamribbons are colored to differentiate the error bands.



a) Multiple vectors

b) Multiple streamlines

c) Multiple streamribbons

Figure 10. Error visualization methods for vector fields.

5. Error Reduction Via Adaptive Refinement

The CRISP CFD package contains a parallel adaptive mesh refinement (AMR) and quality improvement capability for three-dimensional, mixed-element, unstructured meshes [62-63]. Meshes comprised of tetrahedral, prismatic, and hexahedral regions may be readily modified to generate more accurate flow solutions through local refinement and coarsening. Tetrahedral refinement and coarsening are accomplished using Delaunay refinement and edge collapse methods, while cell subdivisions are employed for pyramids, prisms, and hexahedral cells. Load

rebalancing of the decomposed grid is accomplished using the ParMETIS library [64]. Upon completing the mesh modifications and load rebalancing, the solution is interpolated onto the adapted mesh using a grid-transparent procedure based on nearby point clouds [65].

An estimate of the solution error is used to drive mesh adaptation. For steady state applications, the default method currently employed is an interpolation error, based on forming a higher order approximation of the solution at each mesh point using a least-squares approach. The difference between the higher order reconstruction from incident nodes and the current solution forms the error indicator. If the current mesh is sufficiently fine to support the spatial variation in the solution, the estimated error will be low, allowing coarsening to take place. Conversely, a high degree of error indicates additional refinement is needed. This approach has proven successful in a variety of applications and is capable of detecting shear layers, separation, vortical flows, and weak gradients in coarse regions, as well as shocks and expansions, with no tuning required of the user.

With the availability of quantified errors and error sources from the ETE solver, new possibilities for the use of mesh adaptation techniques exist. The notion of refining the mesh at the location of error sources predicted by the ETE has proven promising [52-66]. The ETE source strategy for mesh adaptation uses the location and magnitude of the residuals defined in Eq. 8 to form an error measurement that CRISP CFD applies to locally refine the grid where the largest discretization errors originate. This strategy is fundamentally different from the typical practice in AMR, which is to adapt in regions where the errors have actually manifested. By eliminating the sources, mesh adaptation may be conducted in a more effective and efficient manner, as predicted errors and convergence behavior have shown [52].

6. Verification Procedures and Practices

6.1. Best Practices

The goal of the error transport method is to quantify the discretization error in a particular CFD simulation. When preforming an error analysis by solving the ETE, we seek an accurate prediction of the coarse-to-fine grid increments for both local and integrated quantities of interest. Once the ETE solver is fully converged, the analyst should inspect the results. Generally speaking, error bars predicted by the ETE solution and error functions should:

1. Contain fine grid results;
2. Contain the results of Richardson extrapolation, if available;
3. Decrease in magnitude with grid refinement; and,
4. Not be overly conservative as to be unusable or unreliable.

If the fine grid results fall outside the predicted error bars on a coarser grid, this potentially indicates that new flow features have been introduced as a result of grid refinement. If test data falls outside the predicted CFD solution and error bars, it potentially indicates a deficiency in physical modeling. Ideally, it should be quantitatively shown that the predicted errors scale with the mesh size to demonstrate an apparent order of convergence, though this is not included in the applications shown.

While the original intent of the error transport method was to be an error quantification tool for any given grid and solution, in practice, that is not feasible as there is no means of knowing

whether all relevant flow structures are present given only one grid. It is important to note that the error transport method can only predict the solution increments if the solutions on meshes of varying fidelity are similar. The method cannot account for the absence or presence of a flow structure. The increments for solution sequences on meshes that predict disparate results (e.g., separated vs. attached flow) cannot be evaluated using this method. A minimum of two meshes and ETE solutions is therefore recommended to ensure that the CFD analysis is close enough to the asymptotic range where the ETE technique is considered valid. While it may be more cost effective to perform Richardson extrapolation with two meshes and solutions, this is only possible if one assumes the order of convergence. Otherwise, a three-grid sequence is needed, along with the associated labor and CPU cost.

6.2. Connection with Validation Databases

While the tasks of Verification and Validation (V&V) address separate issues—the demonstration of grid convergence and the accurate reproduction of actual physical processes, respectively—ideally these efforts ought to be combined and applied jointly. Comparison with test data by itself is insufficient as a means of establishing confidence in a given CFD code or physical model. Similarly, physical models cannot be fully evaluated without first ruling out or at least understanding the magnitude of discretization errors. As an extreme example, it is possible to perform solution verification for an inviscid solution to a backward-facing step. One can demonstrate grid convergence, even though the selected physical model is completely inappropriate and incorrect. Once this verification is accomplished, one may readily identify where the physical modeling employed falls short. Verification is thus an equally important aspect of CFD code and model development. With the development of a single-grid error prediction method, the ETE, the task of solution verification has become much less labor and time intensive.

Figure 11 depicts how the error quantification capabilities in CRISP CFD may be combined with the CRAFT Tech Automated Validation Environment (CRAVE) [67]. After implementing a new physical model, the analyst selects a suitable validation test case from the CRAVE database to assess the model. The existing package for the grid and input deck is extracted and the flow solver is run using the new model. Previously, one would then run automated scripts from CRAVE to extract the new results and compare with prior baseline simulations and stored test data. Instead, the analyst now has the option to run an error analysis for the solution, and obtain numerical error bars to include in the resulting plots. If the experimental data falls outside the predicted error bars, this indicates a discrepancy in physical modeling, not in the solution of the equations. Such insight is both useful and valuable. Time permitting, users may employ the AMR capabilities in CRISP CFD to subsequently reduce these errors and re-evaluate the solution along with test data.

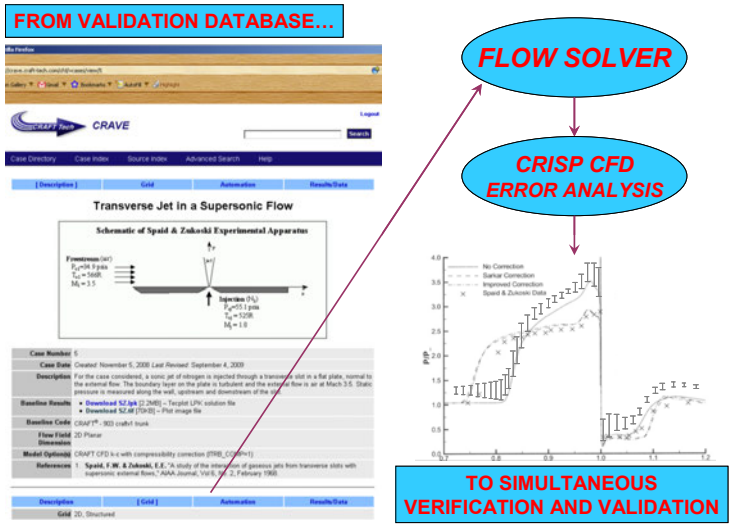
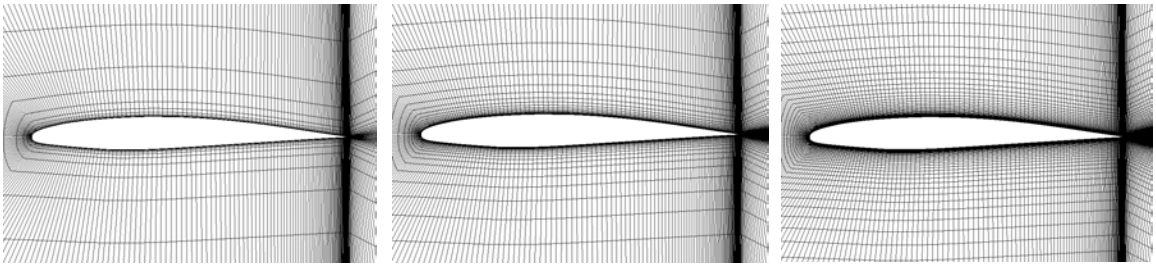


Figure 11. Use of CRISP CFD with CRAVE.

7. Example Applications

7.1. Low Speed Airfoil

Turbulent low speed flow past a two-dimensional airfoil is presented as a realistic wall bounded flow example case. The airfoil, analyzed in previous work by the author is at zero angle-of-attack in a Mach 0.3 freestream. Experimental data containing surface pressure coefficients and detailed velocity measurements is given in the paper by Nakayama [68]. Note that no experimental data for section lift and drag coefficient is available in the original paper. Figure 12 illustrates the mesh sequence used in the current study, while Figure 13a depicts the flowfield through nondimensional velocity contours on the fine grid, along with selected stations in the wake where velocity profile data is available for comparison.

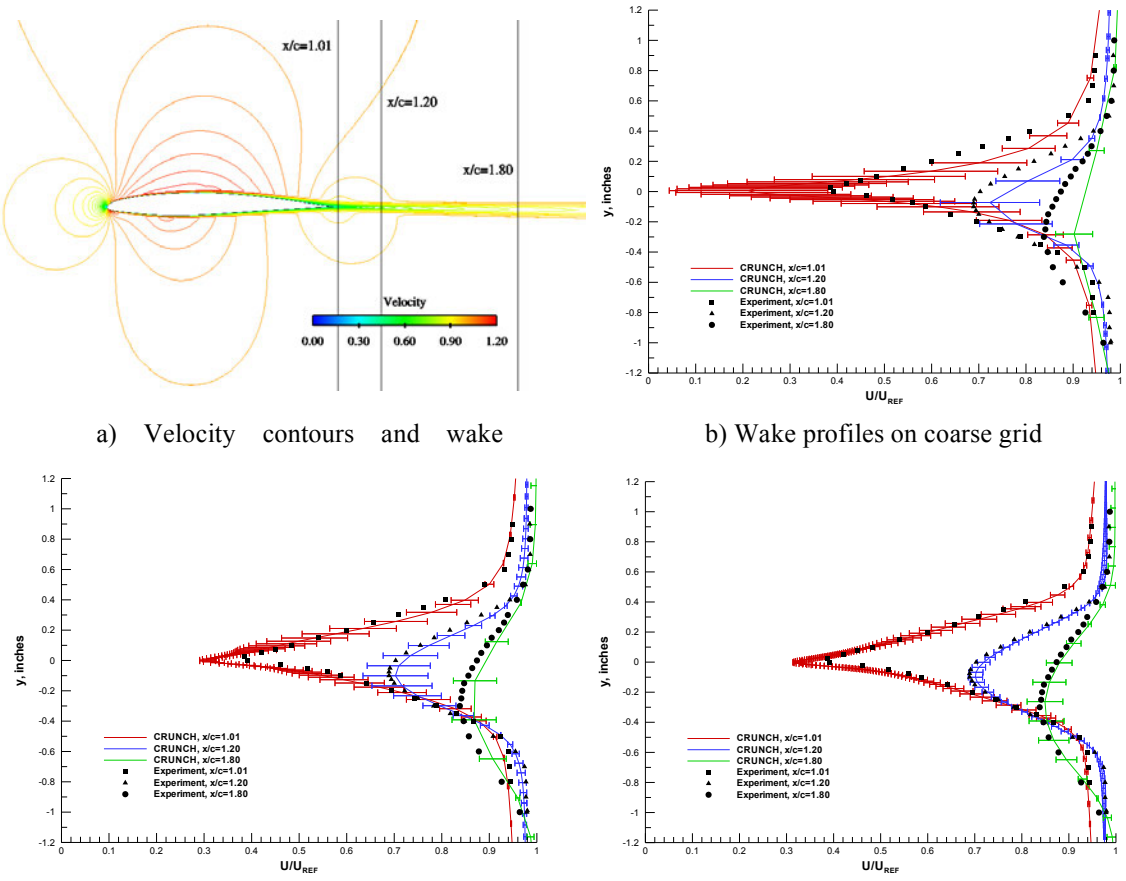


a) Coarse b) Medium c) Fine

Figure 12. Mesh sequence for turbulent airfoil test case of Nakayama [68].

Figure 13b through Figure 13d present the results from the error transport equation for each of the meshes employed. The streamwise (U) component of velocity is compared with experimental data for each of the three wake stations shown in Figure 13a. As one might expect, the magnitude of the errors reduces as mesh resolution is increased and the solution accuracy is improved. Examining the errors for the last station, it is observed that the predicted error increases from coarse to the medium mesh, and then reduces from the medium to the fine mesh. This is an instance where the initial mesh is too coarse to accurately capture the error. The ETE is solved on

the same grid used to solve the Navier-Stokes equations, and thus is subject to the same numerical discretization issues. Figure 14 compares the predicted lift and drag coefficients for the three-grid sequence. The error bars derived from the inviscid ETE solution provide confirmation of the level of accuracy achieved, at least for C_L. In contrast, the predicted errors in C_D are considerably larger than those provided by Richardson extrapolation. The primary contributor to this is the computed error in wall shear stress as discussed in prior work by the author [49, 50]. Using the viscous ETE formulation, the C_L error predictions are not affected, however the C_D error bars are greatly improved and more in line with the results of Richardson extrapolation.



a) Velocity contours and wake

b) Wake profiles on coarse grid

c) Wake profiles on medium grid

d) Wake profiles on fine grid

Figure 13. ETE predictions for low speed airfoil.

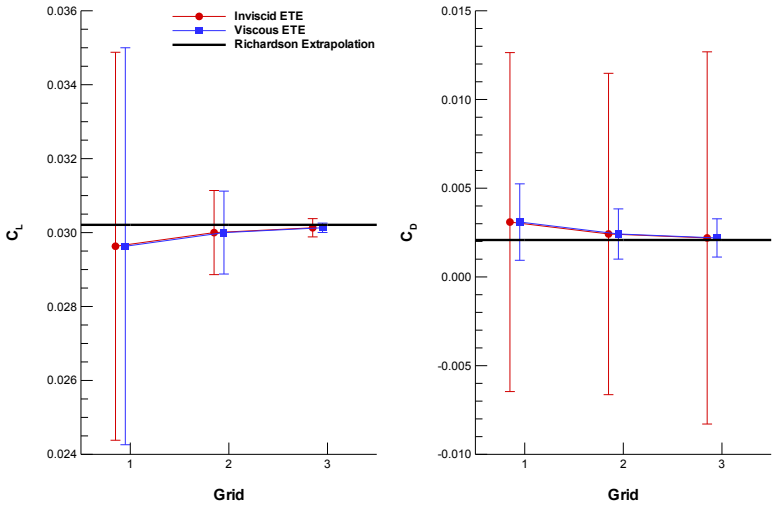


Figure 14. Error predictions for aerodynamic coefficients for a low speed airfoil.

7.2. Backward Facing Step

As an example of error quantification for a separated turbulent flow, we consider a simple backward facing step, which has been a standard benchmark in solution verification and validation studies [25]. Data for this case is well validated and comparisons are made here to those of Driver and Seegmiller [69]. Details of the geometry are shown in Figure 15. The step height H = 1.27 cm is given as a reference length. The tunnel span has a length of 12H, and the tunnel height prior to the step is 8H. The boundary layer is well developed prior to the step with a thickness of approximately 1.5H. The inlet was modeled far upstream to achieve this boundary layer growth. The approach flow itself has a centerline velocity of 44.2 m/s, at standard atmospheric pressure and temperature.

A series of four grids were employed, as summarized in Table 5, with the medium grid shown in Figure 16 for reference. All grids were hexahedral, with the medium grid in the region of interest containing 27,200 cells. The cell spacing for the coarse grid was halved in both the x and y directions, while the cell spacing was doubled successively in both directions for the fine and very fine grids. Contours of velocity magnitude are presented in Figure 17, in which the boundary layer thickness and reattachment point are clearly evident. We present error predictions for two sets of simulations. The node-centered solver CRUNCH CFD was used along with the standard k-ε turbulence model. We also present results for this case using the cell-centered solver AVUS [71] and the k-ω turbulence model.

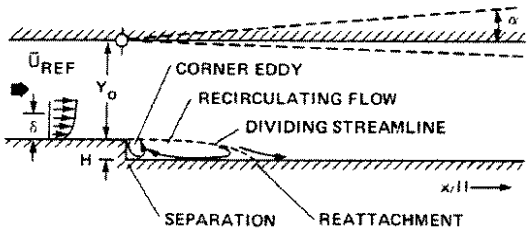


Figure 15. Geometry of backward facing step.

Table 5. Resolution in the region of interest for backward facing step study.

Grid	Resolution	Number of Cells
1	Coarse	6,800
2	Medium	27,200
3	Fine	108,800
4	Very Fine	435,200

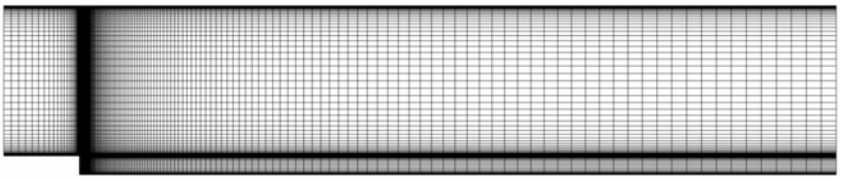


Figure 16. Computational domain showing cell spacing for the medium grid.

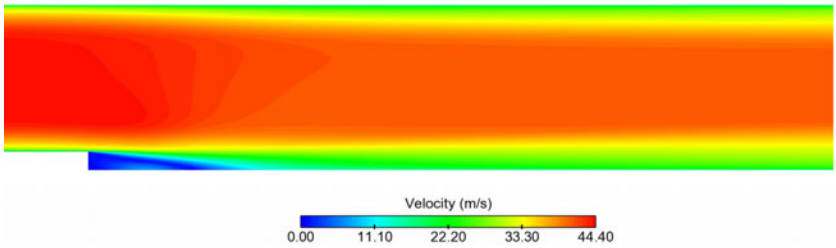


Figure 17. Contours of velocity magnitude for unit turbulent flow study.

Results for the node-centered CRUNCH CFD code are presented in Figure 18 through Figure 20. Figure 18 depicts streamwise velocity profiles at two stations along with experimental data and numerical error bars provided by the ETE solution. Results are shown at X/H = 2 and X/H = 8, two and eight step heights away from the lip, respectively. For the sake of clarity, each station is shown twice with error bars on Grid 1 and Grid 2 included in the left image and error bars on Grid 3 and Grid 4 in the right image. The CFD results for *U* velocity are nearly unchanged for all four grids, and show good agreement with the experimental data. The error bars for the coarse grid (Grid 1) are the largest of the four grids. The magnitude of the discretization error is the largest at the edge of boundary layer near the topside wall and in the recirculation/shear layer toward the lower wall. As the grids are refined, the magnitude of this error decreases. The medium grid (Grid 2) exhibits errors on the same order of magnitude as Grid 1. The fine grid (Grid 3) results show that the grid error has significantly diminished, and on the very fine grid (Grid 4) the error bars have nearly disappeared. The results for Grid 4 are thus grid converged.

The transverse velocity profiles are shown at the same locations in Figure 19. Grid refinement produces a more pronounced change in *V* velocity, unlike the U velocity profiles. The error bars are also significantly larger, more noticeably in the recirculation region at X/H = 2, where even Grid 3 has a notable amount error. Again, the errors decrease with each successive refinement, and the absence of error at Grid 4 proves that the solution is grid converged. When comparing the profiles for all four cases, the Grid 1 error bars contain the solution for the Grid 4 results, indicating the grid sequence is in the asymptotic range. Finally, Figure 20 presents turbulent kinetic energy. The results demonstrate the same trends as seen in the *U* and *V* velocity plots, and further confirm those conclusions. The error bars are largest for Grid 1, decrease incrementally for Grid 2, decrease substantially for Grid 3, and nearly vanish for Grid 4 when grid convergence is achieved. The magnitude of the error is still largest in the recirculation region, and also relatively large at the edge of the boundary layer. It is worth noting that turbulent kinetic energy is significantly over predicted downstream of the reattachment point, at X/H = 8. Since the discretization errors have been quantified using the ETE, and the error bars on the finer meshes do not contain the test data, we may conclude a physical modeling error is present. The turbulence modeling would need to be improved to better compare with the data.

a) X/H = 2

b) X/H = 8

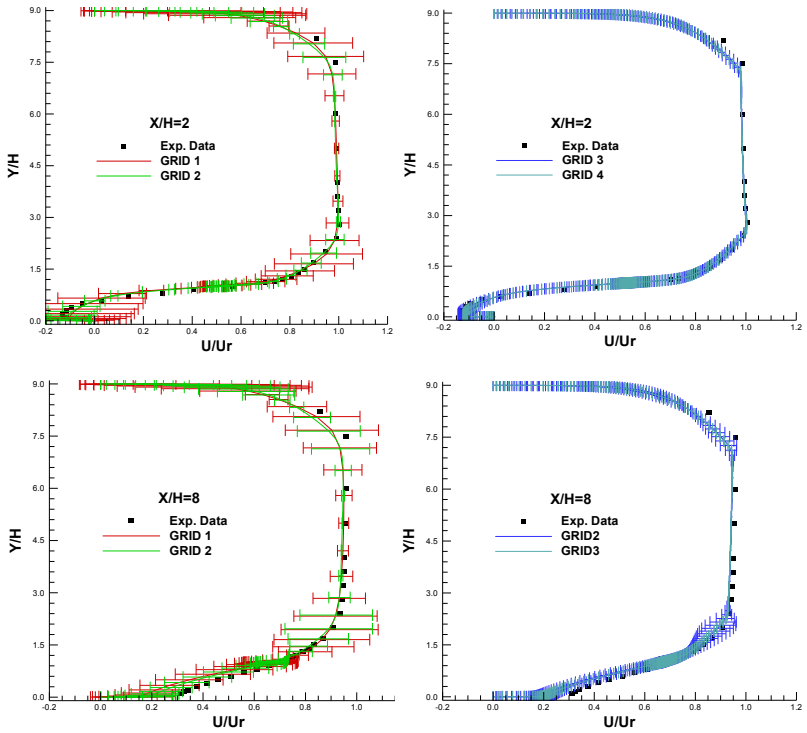


Figure 18. Error in streamwise velocity for *k-ε* solutions of backward facing step.

a) X/H = 2

b) X/H = 8

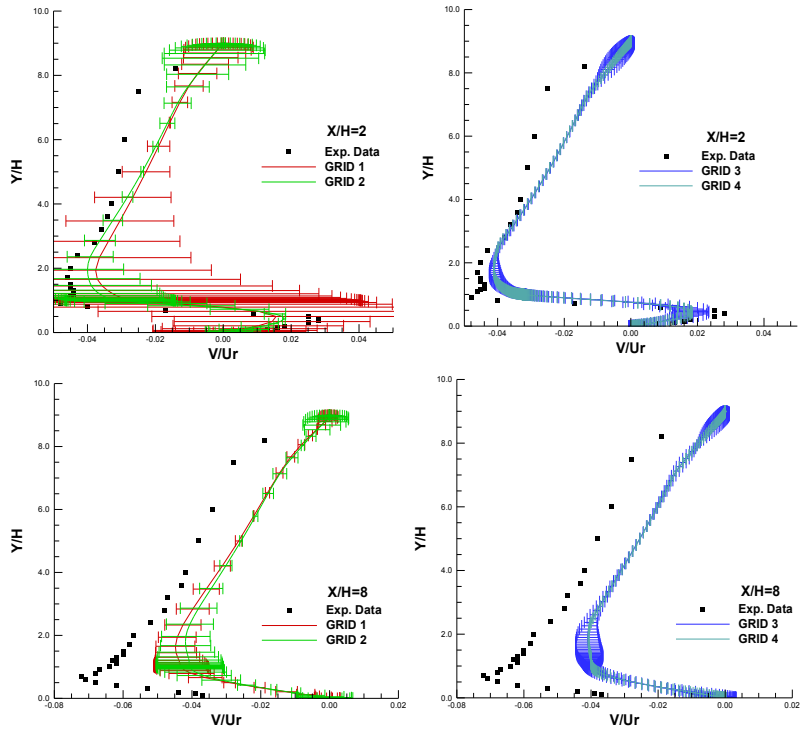


Figure 19. Error in transverse velocity for *k-ε* solutions of backward facing step.

a) X/H = 2

b) X/H = 8

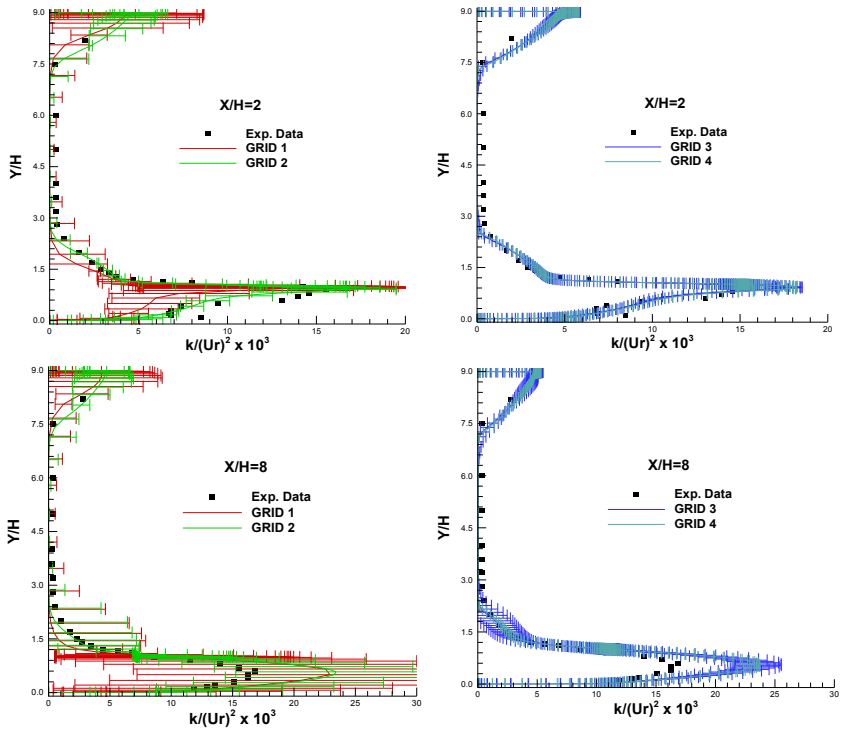
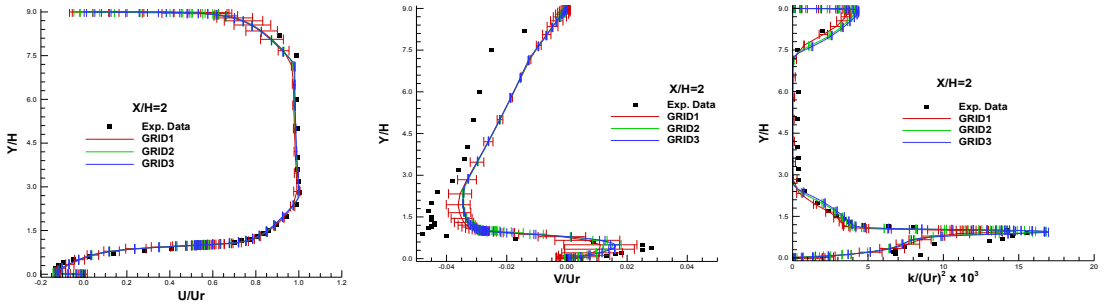


Figure 20. Error in turbulent kinetic energy for *k-ε* solutions of backward facing step.

Analysis of the backward facing step is repeated using the AVUS flow solver and the *k-ω* turbulence model. Simulations are presented on the first three meshes, as there was no discernible difference between Grid 3 and Grid 4. ETE solutions are considered using both NC and CC formulations. We again consider station cuts perpendicular to the mean flow direction at X/H values of 2 and 8. The former station cuts through the separation/recirculation region while the latter is located downstream of the reattachment point. Examining the streamwise and transverse velocity components and turbulent kinetic energy, there is little difference between the profiles extracted from the three solutions. This is particularly true for the medium and fine grids, Grid 2 and Grid 3. The ETE solver is able to predict this small increment between the three solutions. As seen in Figure 21 and Figure 22, as the mesh is refined, the error decreases appropriately with the medium grid, whose solution is much closer to the fine grid results. In addition, the error generated by the coarse and medium grids largely contains the value provided by the fine mesh. The largest error occurs in the recirculation region adjacent to the step in Figure 21 and at the edge of the boundary layer from the upstream walls. This represents the greatest amount of error in the flow solution. The error quickly diminishes to an appropriate value where there is practically no difference in the flow field quantities.

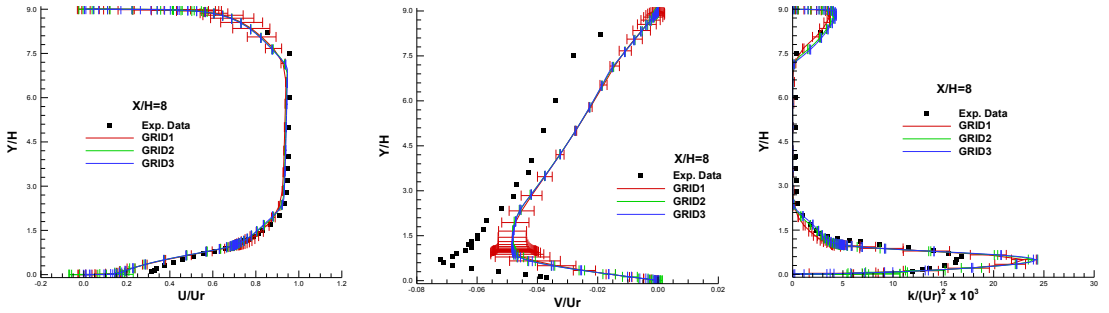
Figure 23 and Figure 24 provide a comparison between the results obtained by using CC and NC discretizations within the ETE solver. For the NC error analysis, the solution obtained on the CC grid is transferred onto a NC grid and solved in a NC fashion. The error predicted for the coarsest grid using both methods is presented along with the fine grid solution for the same quantities and profiles as Figure 21 and Figure 22. The most immediate observation is the noticeable decrease in error predicted by the NC formulation compared with the CC formulation. While the CC results contain the fine grid result, the NC error is not able to capture the difference in the two solutions. This deficiency is clearly present in the transverse velocity component near the Y/H location of

1.0 in Figure 23b and Figure 24b, but is present for all three quantities. By under predicting the error and not containing the fine grid result, this absence of error can lead to misplaced confidence in the solution provided by this coarse grid. While it would seem intuitive that one should use a cell-centered ETE to analyze a cell-centered CFD solution, this case provides solid evidence.



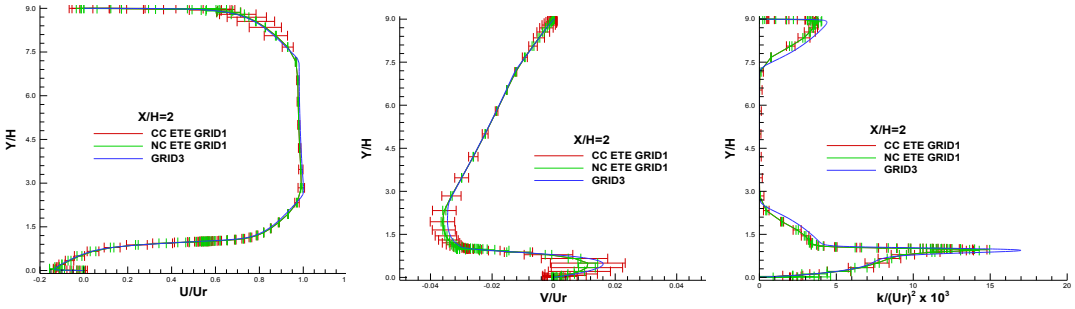
a) Streamwise velocity b) Transverse velocity c) Transverse kinetic energy

Figure 21. Error predictions for AVUS *k-w* solutions of backward facing step at X/H=2.



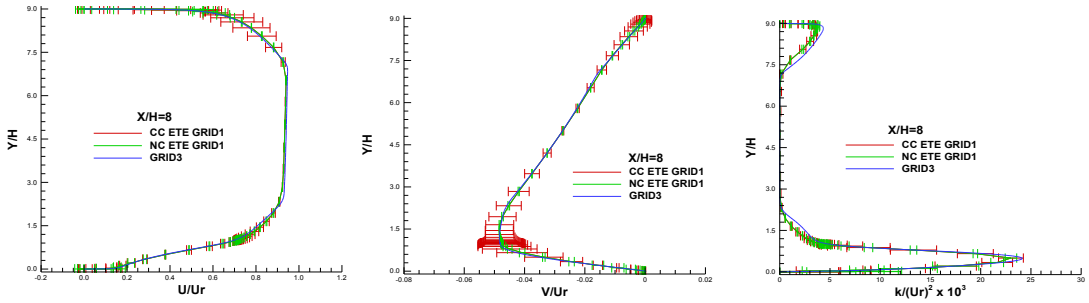
a) Streamwise velocity b) Transverse velocity c) Turbulent kinetic energy

Figure 22. Error predictions for AVUS *k-w* solutions of backward facing step at X/H=8.



a) Streamwise velocity b) Transverse velocity c) Turbulent kinetic energy

Figure 23. Comparison of NC and CC ETE predictions for AVUS solutions at X/H=2.



a) Streamwise b) Transverse velocity c) Turbulent kinetic

Figure 24. Comparison of NC and CC ETE predictions for AVUS solutions at X/H=8.

7.3. High Lift Prediction Workshop

Our error prediction methodology was demonstrated at the 1st AIAA High Lift Prediction Workshop (HiLiftPW-1) in June 2010 [6, 72]. ETE solutions were analyzed for CFD solutions obtained using the well-established unstructured Navier-Stokes solvers FUN3D [73] and USM3D [74] for two angles of attack where formal grid refinement studies were to be conducted. Figure 25 depicts the NASA High Lift Trap Wing considered in this study and Figure 26 illustrates a representative pressure coefficient distribution with various stations on the slat, main element, and flap, denoted where experimental data are available for comparison. In this discussion, we focus solely on the outboard stations near the wing tip, at 85% and 95% span, where grid refinement effects are more pronounced. Both the FUN3D and the USM3D solutions employ the Spalart-Allmaras turbulence model.

Figure 25. NASA High Lift Trap Wing.

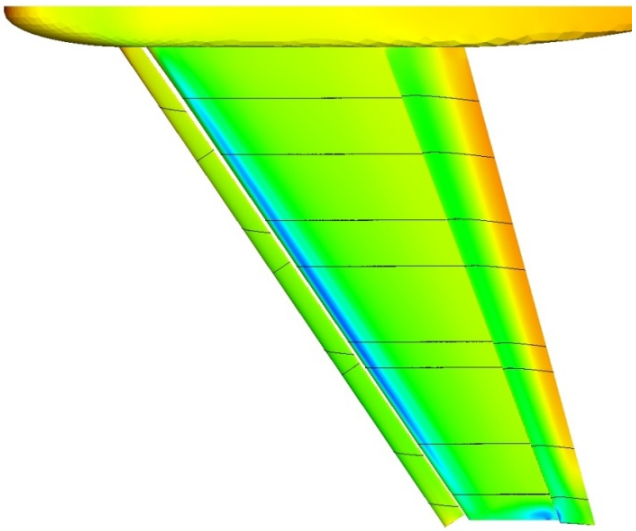
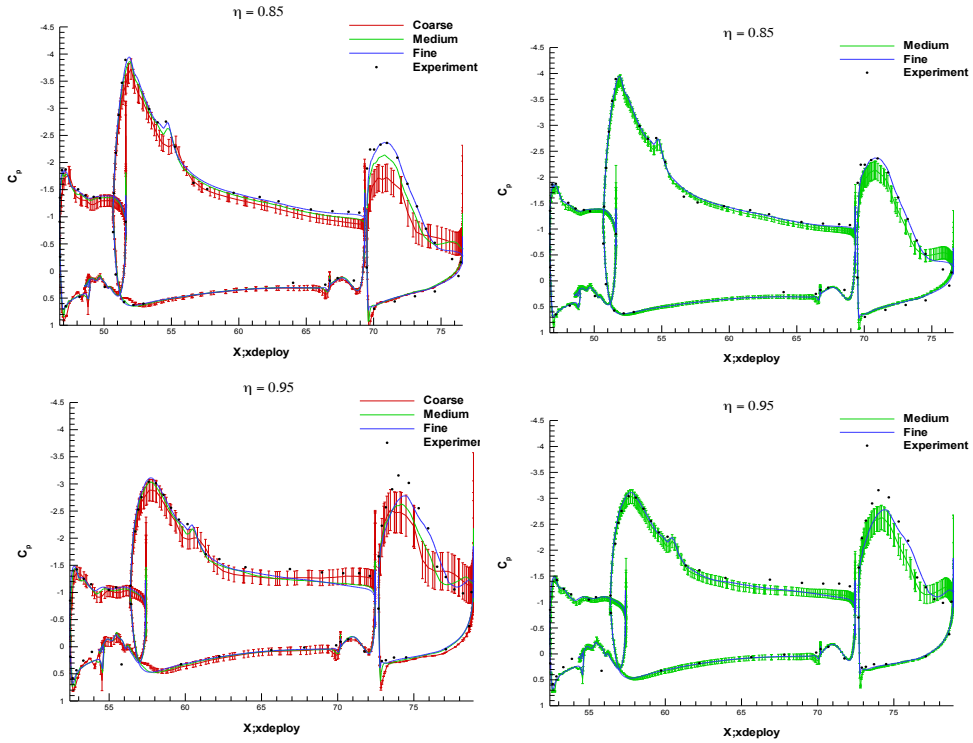


Figure 26. Station cuts on slat, main element, and flap.

Figure 27 presents predicted errors in surface pressure coefficient from node-centered FUN3D solutions at 85% and 95% span for each grid in the mesh sequence at an angle-of-attack of 13 degrees. The grid level is denoted by color; red indicates a coarse grid solution, green for the medium grid, and blue the finest grid. In plots of coarse and medium grid solutions, finer grid solutions are included for comparison, where ideally the coarse and medium grid error bars should contain these results. In cases where test data lay outside both the solution and the

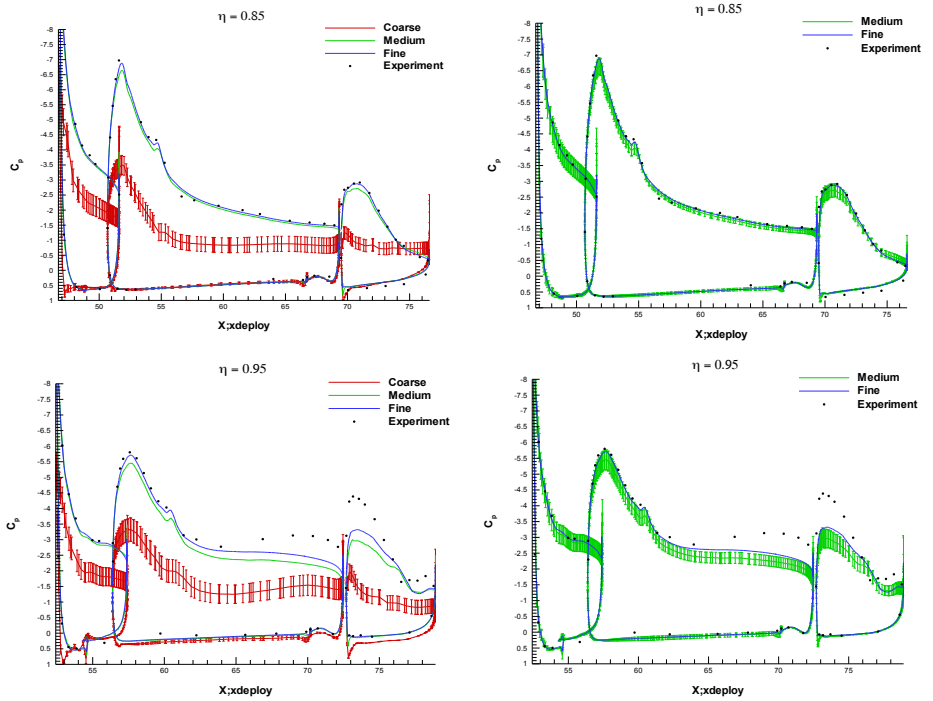
predicted error bars, this indicates a possible deficiency in physical modeling, as the error bars should provide solution verification to isolate resolution errors from other sources of error. Such discrepancy is most visible at the outboard stations of 85% and 95% span at higher angles of attack, where local suction peaks are present in the test data but absent in the CFD results.

Predicted errors in surface pressure coefficient were generally quite good, consistently capturing the increment between coarse and fine grid solutions, and in many instances capturing the increment between the CFD solution and experimental data. This is quite promising, as it lends credibility to the ETE method as a reliable error quantification technique. However, note that at the higher angle-of-attack of 28 degrees, presented in Figure 28, the coarse grid errors do not contain results from the medium and fine grids. This is due to the fact that the coarse grid predicts stall, while the medium and fine grids predict an attached flow over the wing. It is important to recognize that the error transport model cannot account for the absence or presence of a flow feature that does not exist on finer meshes. However given a set of similar solutions, the ETE can predict the increments. Predicted errors on the medium grid at 28 degrees, shown in Figure 28b, are well in agreement with the fine grid result. Since HiLiftPW-1, the coarse grid solution was found to be attributed to a hysteresis effect, and an attached flow could be obtained by ramping up the angle-of-attack.



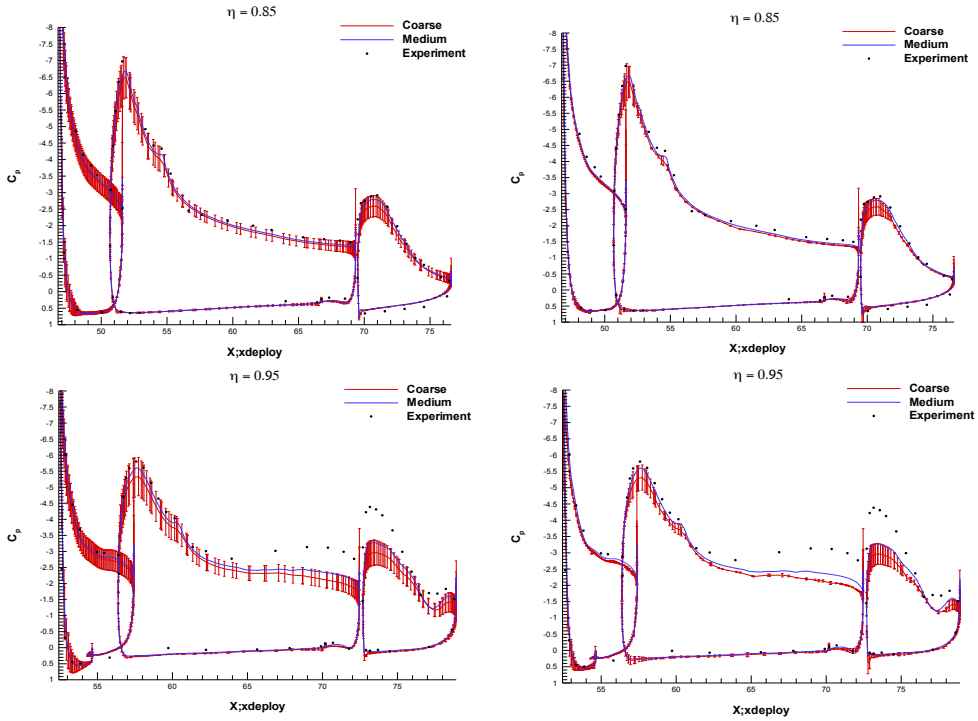
a) Coarse grid errors b) Medium grid errors

Figure 27. Error predictions for FUN3D solutions, $\alpha = 13°$.



a) Coarse grid errors b) Medium grid errors

Figure 28. Error predictions for FUN3D solutions, $\alpha = 28°$.



a) Node-Centered b) Cell-Centered ETE

Figure 29. Error predictions for USM3D solutions, $\alpha = 28°$.

Cell-centered solutions from USM3D were also analyzed in the original workshop, a sample of which is depicted in Figure 29a. The original ETE predictions from 2010 using the node-centered discretization overpredict the magnitude of the differences between the coarse and medium resolution grids, most notably on the slat and flap upper surfaces. In Figure 29b, we present revised error predictions employing the new cell-centered ETE formulation [53, 57]. The coarse grid errors are now in good agreement with the actual increments, with the exception of the main element plateau at 95% span. Recall that for tetrahedral meshes such as these, the number of cells is typically 5-6 times the number of vertices. Hence a node-centered discretization solves the ETE on a coarser mesh and predicts larger errors that are not consistent with the cell-centered CFD solutions. Figure 29 supports this conclusion.

7.4. Missile at Angle of Attack

An ogive missile at Mach 2.5 and an angle of attack of 14 degrees is presented as a demonstration of how the CRISP CFD error quantification and reduction package can be used to combine both aspects of V&V. The primary flow feature of interest is a vortex, which forms from flow separation and grows along the length of the missile body, as depicted in Figure 30. Experimental data available from Sturek, et al. [76] consists of pitot pressures through the vortex core. The original unstructured mesh is purposely coarse and does not sufficiently resolve the vortex. This case is particularly useful as a solution verification exercise since the k-ε turbulence model employed over predicts the turbulent viscosity and pitot pressure, allowing separate study of grid-induced errors and deficiencies in physical modeling. More advanced turbulence models such as explicit algebraic stress models (EASM) are able to match the test data, as depicted in Figure 31, which compares pitot pressures through the vortex in the spanwise (z) direction. This is well known from previous studies [77].

Three mesh refinements were performed using both conventional and ETE source mesh adaptation strategies, with changes in mesh size reported in Table 6. Both approaches effectively capture the vortex, as seen in Figure 32, which presents pitot pressure contours and the resulting meshes at an axial station of $x/D = 11.5$. Note that while the conventional AMR strategy concentrates refinement on the key flow features, namely the separation line and vortex core, the ETE source strategy introduces resolution in the field as well. Also note that the ETE source approach resulted in smaller grids, as indicated in Table 6.

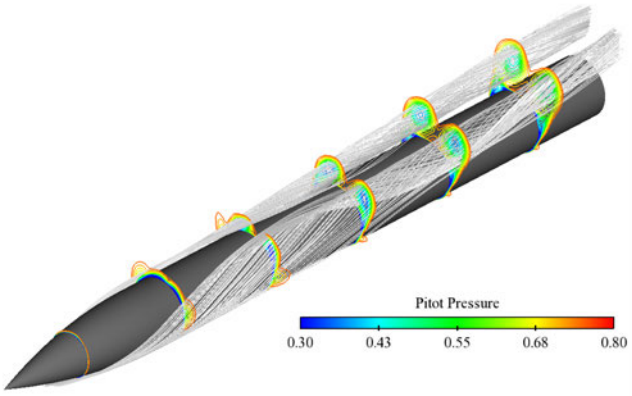


Figure 30. Isometric view of missile with visualization of vortex streamlines and contours of pitot pressure.

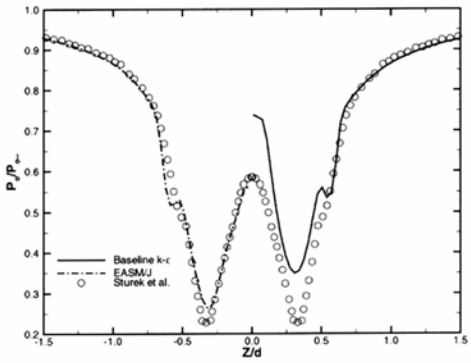
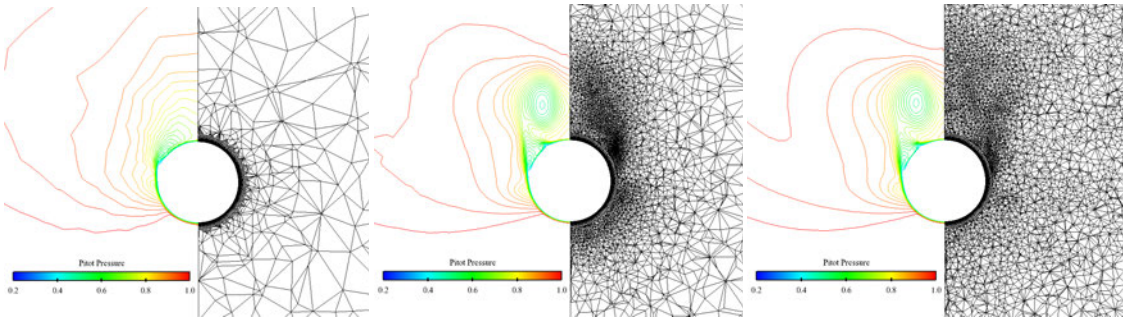


Figure 31. Effect of turbulence model on pitot pressure prediction (from Ref. 77).

Table 6. Mesh sizes, missile adaption study.

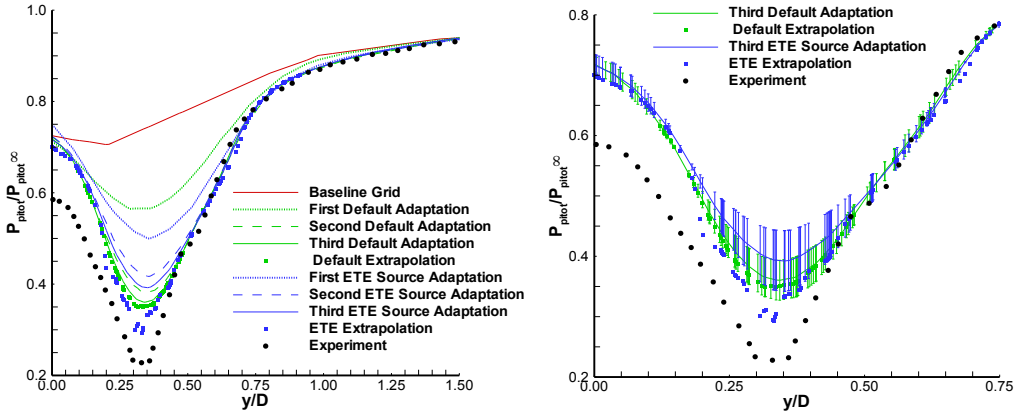
Mesh Level	Default Adaptation	ETE Source Adaptation
Baseline	231,486	231,486
First Refinement	389,583	374,361
Second Refinement	844,749	761,629
Third Refinement	1,314,918	1,091,633



a) Baseline grid b) Default adaptation c) ETE source adaptation

Figure 32. Improvement of missile flow field prediction with adaptive mesh refinement.

Figure 33 presents comparisons of predicted pitot pressures and experimental data through the vortex core in the spanwise (y) direction at $x/D = 11.5$, along with the results of Richardson extrapolation and predicted error bars from the ETE solver and Error Function Library. It should be stressed that using the k-ε model, a grid converged result will not match the test data, as discussed previously. Figure 33 illustrates several points. First, the solution is approaching a grid-converged result using either method, despite a poorly resolved initial grid. Second, the predicted error bars from the ETE solver generally contain the results of conventional Richardson extrapolation, with the exception of the minimum pressures predicted using the ETE source approach. Finally, the extrapolated values are different between the default and the ETE source grid sequences. This raises the question of whether the traditional, feature-based error indicator is converging to the correct value that would be obtained through uniform refinement. Conventional mesh adaptation strategies such as this aim to reduce local errors, thereby equidistributing the error in the domain [78]. In contrast, the goal of the ETE source method is to eliminate where the error originates, and this approach is more effective in reducing error globally.



a) Convergence of pitot pressure profiles b) Fine grid error bars and extrapolated values

Figure 33. Pitot pressure comparisons with test data and Richardson extrapolation.

7.5. Aircraft Engine Exhaust

Exhaust plumes are an important class of turbulent flows dominated by shear layer turbulence. The following demonstration presents error quantification for a supersonic turbojet exhaust into still air. Figure 34 illustrates the engine geometry and the meshes employed. The engine nozzle features a region of liner cooling approximately 2.5 diameters upstream of the nozzle exit, and additional cooling is provided by a slot along the inside circumference of the nozzle exit. The engine core and liner flow conditions for the turbojet are given in Table 7, while Table 8 lists the number of vertices in each grid. The grids are axisymmetric. Figure 35 shows the fine mesh solution, nondimensionalized by freestream conditions. The temperature and velocity contours depict the shock cells, which are characteristic of a supersonic jet. The turbulent kinetic energy (TKE) contours illustrate the development of the shear layer as the exhaust mixes with the ambient air.

Figure 36 compares centerline temperature and velocity distributions for the coarse and fine grids. Numerical error bars are predicted using the viscous ETE. Where possible, Richardson extrapolation was performed and is presented as the "exact" solution. Note that for instances where the solution sequence did not exhibit monotonic behavior, data is absent from the "exact" solution. In general, the fine grid results compare well with Richardson extrapolation, and small errors are observed. The viscous ETE properly captures the difference between the coarse grid and extrapolated results.

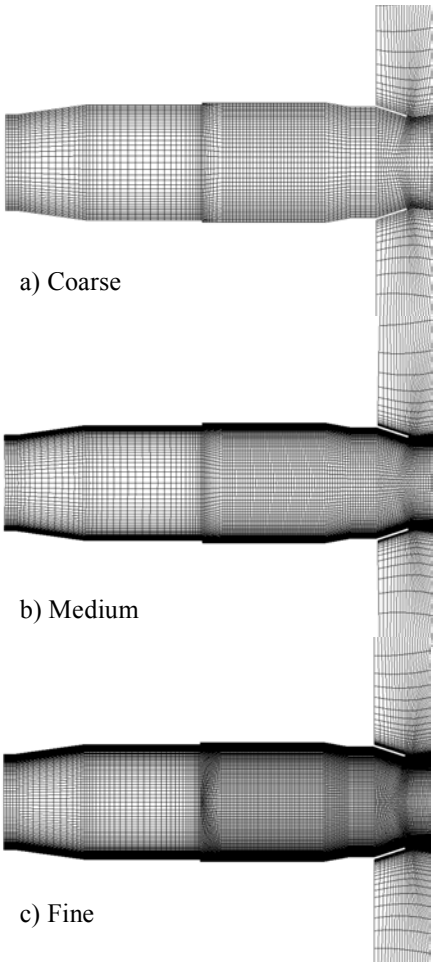


Figure 34. Aircraft exhaust grids, near field.

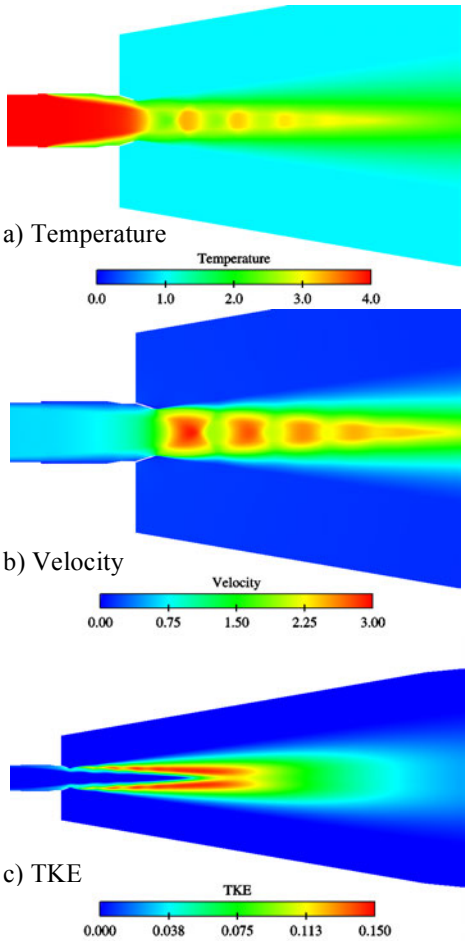


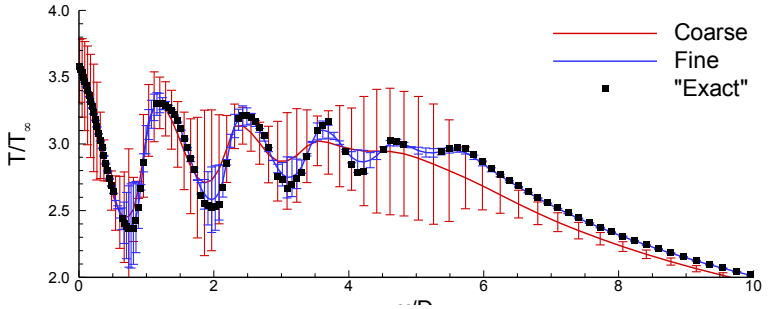
Figure 35. Fine grid solution, engine exhaust case.

Table 7. Engine conditions.

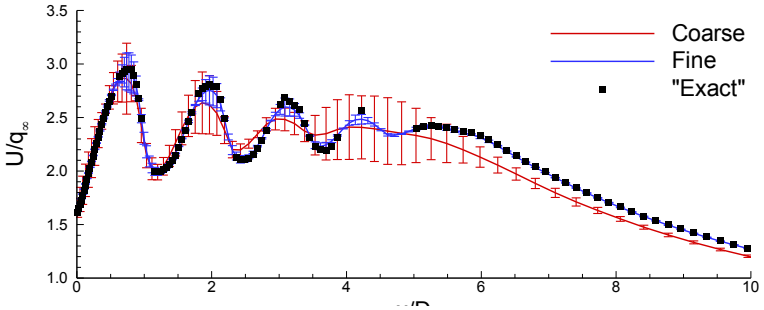
	Pressure (Pa)	Temperature (K)
Core	318456	1225
Liner	146941	505
Slot	146941	505

Table 8. Mesh statistics, engine exhaust case.

Mesh Size	Number of Vertices
Coarse Grid	13301
Medium Grid	36308
Fine Grid	80590



a) Temperature



b) Velocity

Figure 36. Predicted errors in centerline profiles, engine exhaust case.

7.6. Buoyant Plume in Crossflow

The final case we selected to demonstrate the utility of the CRISP CFD system for solution verification is the heated jet in crossflow experiment of Kamotani and Greber [79]. Air at 400°F is injected normal to a freestream of 30 ft/sec and room temperature. Our study considers a jet momentum ratio $J = \rho_j U_j^2 / \rho_0 U_0^2$ of 15.3. The baseline unstructured mesh is predominantly tetrahedral, with prismatic elements adjacent to the wall. In this study, the ETE source strategy is applied when adapting the grid. Table 9 provides the change in mesh size over the three-grid sequence. Figure 37 illustrates the jet flowfield through streamribbons colored by temperature. The subsequent vortical rollup of the jet as it penetrates the crossflow is evident in the temperature contours provided at several stations downstream of the jet.

The predicted penetration and mixing characteristics change significantly with mesh adaptation, even though the degree of refinement is not very aggressive. This is evident in figures 38 and 39, where the change in the temperature contours as the mesh is adapted is quite apparent. The coarse baseline grid artificially diffuses the jet, whereas after two refinements, the hot core is seen to penetrate farther into the crossflow, and a more well-defined vortex rollup develops.

Table 9. Mesh sizes used in buoyant plume study.

Mesh Level	Number of Vertices
Baseline	100,052
First refinement	157,105
Second refinement	212,237

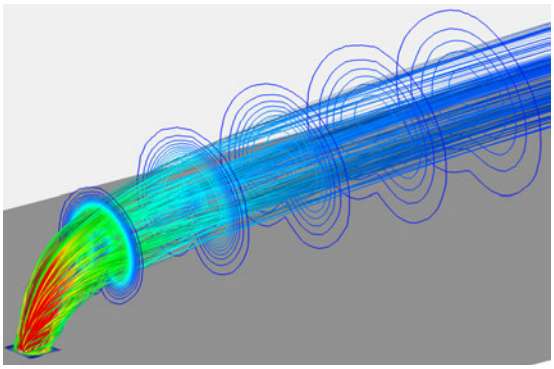


Figure 37. Jet flowfield, in crossflow, with streamribbons colored by temperature.

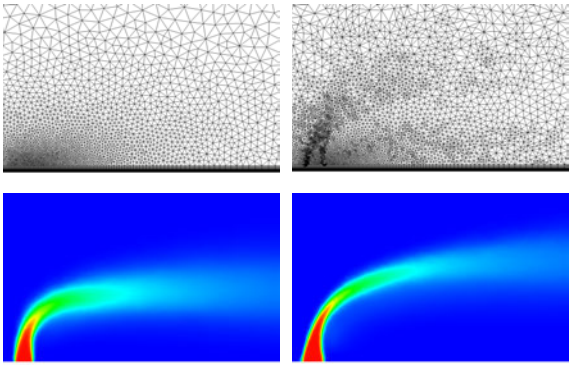
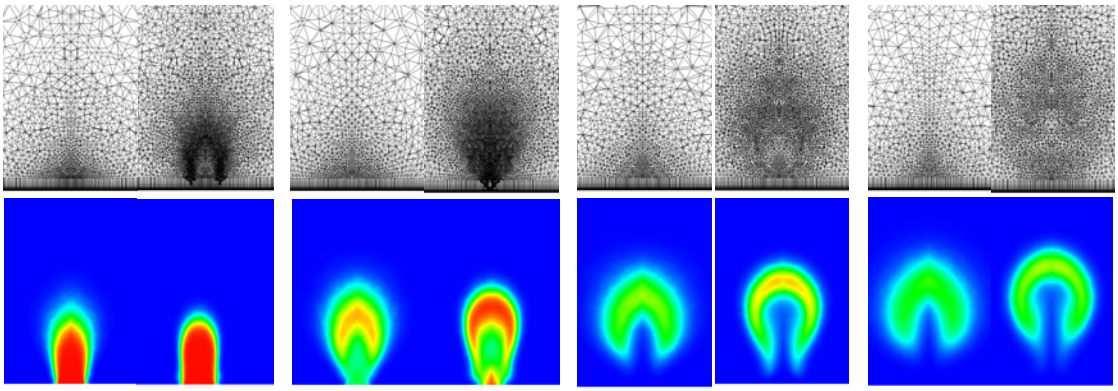


Figure 38. Effects of mesh adaptation on plume penetration, symmetry plane.



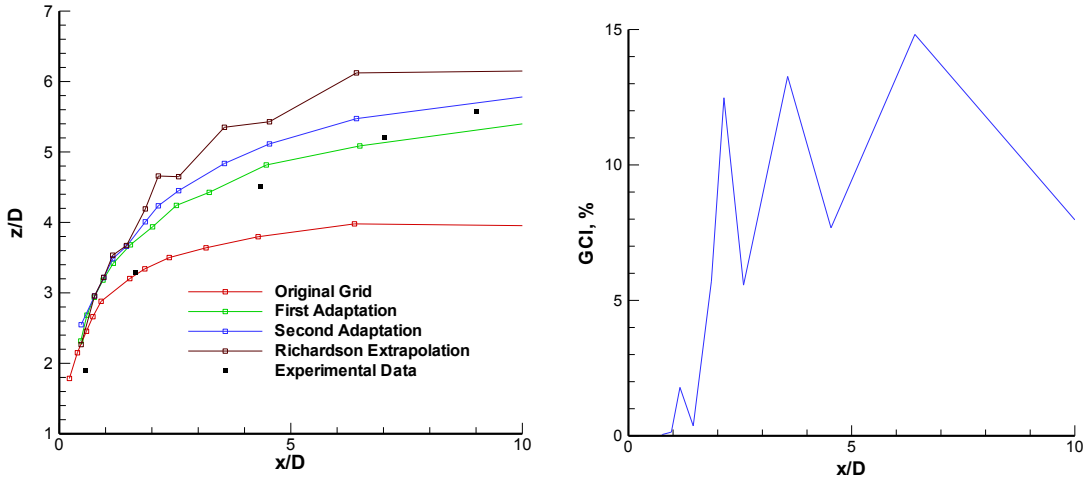
a) $x/D = 0.0$ b) $x/D = 0.5$ c) $x/D = 1.0$ d) $x/D = 1.5$

Figure 39. Effects of mesh adaptation on plume penetration, original (left) and adapted (right) temperature contours at selected crossflow planes.

Figure 40 compares the predicted temperature trajectory for each mesh in the grid sequence, and provides a measure of convergence. While it is clear that the adaptive refinements have improved the correlation with the experiment, the results of Richardson extrapolation indicate that a third refinement is likely necessary. The GCI exceeds 5% for most of the trajectory, and the level of convergence is somewhat erratic. Yet this is exactly what the CFD analyst needs: an accurate, quantifiable assessment of solution error, and a means to correct it.

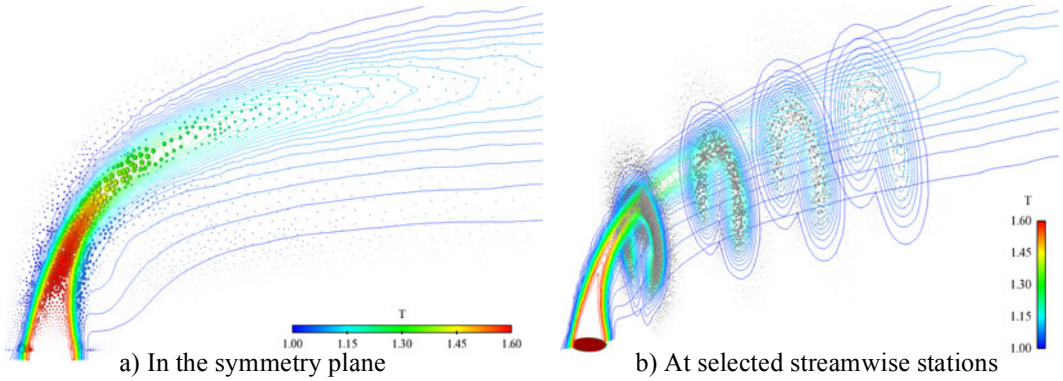
A novel technique for simultaneous rendering of a variable and its associated error, called "error bubbles," has been developed [52] to assist the user in visualizing and interpreting errors predicted by the ETE solver. Figure 41 presents error bubbles and contours of temperature in both the symmetry plane and at selected stations downstream of the jet. The error bubbles are colored by temperature in Figure 41a, and by a constant color in Figure 41b. The majority of the predicted error lies at the end of the jet's core, which suggests that further refinement would continue to increase the jet penetration and reduce the radial spread, both of which are affected by grid resolution. The error bubble visualization method is able to present the analyst with useful,

quantitative information on the level of error without detracting from the contours themselves, since the bubbles vanish in the limit of zero error. Other rendering methods such as texturing [52] remain viable options and may be pursued in future studies.



a) Extrapolation of temperature trajectory b) Grid Convergence Index (GCI)

Figure 40. Convergence of temperature trajectory with mesh adaptation.



a) In the symmetry plane b) At selected streamwise stations

Figure 41. Temperature error visualization using error bubbles.

8. Current Research Needs

While considerable research, development, and validation has been invested in the CRISP CFD code and it has had notable success for steady state problems, further work is needed to bring the error quantification tools into other problem classes. The temporal evolution of spatial errors remains an area of challenging research. Transient flows, particularly large eddy simulations (LES), may involve assessment of mean and fluctuating quantities. A natural extension of this problem class is to account for problems with moving or deforming boundaries. CFD is increasingly being applied to problems involving more complex physical phenomena as well. The prediction of sprays, particle-laden flows and free surface flows require additional submodels and/or transport equations, coupled to the conservation equations for the carrier phase in the case of dispersed multiphase flows. Error prediction for such problems would be beneficial in assessing the validity of new physical models by isolating discretization errors. With these advances, solution verification could be applied with even more complex test cases in validation databases such as CRAVE [67].

8.1. Errors in Chemical Kinetics

It can readily be seen from prior discussion that the accuracy of the chemical source term ultimately depends on the accuracy of the local species mass fractions and mixture temperature predictions. Numerical diffusion will adversely affect species production/destruction, as the local species concentrations and/or temperatures on a given grid may be lower than what is obtained in the fine grid limit. In the extreme, a coarse grid may prevent the formation of a flame entirely. Consider the source term defined in Eqn. 8.

$$\bar{D}\left(\bar{Q}\right) - \bar{D}\left(\bar{Q}^h\right) \tag{33}$$

The source term $\bar{D}\left(\bar{Q}^h\right)$ is evaluated on the given grid of mesh width h, using the actual solution obtained on that grid. The term $\bar{D}\left(\bar{Q}\right)$ represents the exact source term that would reflect the exact solution of the governing equations for the chemically reacting system. In essence, $\bar{D}\left(\bar{Q}\right)$ would be the source term for a grid-converged solution. Expressed in terms of the dependent variables, it becomes:

$$\bar{D}\left(\bar{Q}\right) = f\left(y_{i_{exact}}, T_{exact}\right) \quad \bar{D}\left(\bar{Q}^h\right) = f\left(y_i^h, T^h\right) \tag{34}$$

While the exact solution is not typically known, it may be approximated by applying the error predicted by the current solution of the ETE.

$$\bar{D}\left(\bar{Q}\right) = f\left(y_i^h + \varepsilon_{y_i}, T^h + \varepsilon_T\right) \tag{35}$$

Thus by evaluating the chemical source term using the solution and the predicted errors, we obtain a new residual term for the ETE, which represents error in the chemical source terms.

$$\vec{R}_{CHEM} = \iiint \left(\bar{D}\left(\bar{Q}\right) - \bar{D}\left(\bar{Q}^h\right)\right) dV \tag{36}$$

$$\boxed{\frac{\partial}{\partial t}\iiint \vec{\varepsilon}\, dV + \iint \left(\left(A\left(\bar{Q}^h\right)\vec{\varepsilon}\right)\cdot\hat{n}\right) dA = \iint \left(\vec{G}\left(\vec{\varepsilon}\right)\cdot\hat{n}\right) dA - \vec{R}_{INV} - \vec{R}_{TURB} - \vec{R}_{CHEM}} \tag{37}$$

8.2. Errors In Transport (Scalar Fluctuations)

In our current implementation of the error transport method, turbulent Prandtl and Schmidt numbers are assumed to be constant, and as such the predicted errors in turbulent thermal conductivity and mass diffusivity, seen in Eqns. 18 and 19, are solely a function of the predicted error in eddy viscosity, as in Eqn. 23. A natural extension of this formulation is to expand the errors in these quantities to reflect a variable Pr_t and Sc_t field, as provided by a scalar fluctuation model (SFM). To accomplish this, the ETE would be modified to include error in the energy and mixture fraction variance equations of the SFM, from which one would obtain error in the conserved variables ρk_e, $\rho \varepsilon_e$, ρk_f, and $\rho \varepsilon_f$.

Following the error propagation approach discussed previously, the error in turbulent Prandtl and Schmidt numbers become:

$$\varepsilon_{\mathrm{Pr}_t} = \frac{1}{2}\mathrm{Pr}_t\left(\frac{\varepsilon_k}{k} - \frac{\varepsilon_k}{\varepsilon} - \frac{\varepsilon_{k_e}}{k_e} + \frac{\varepsilon_{\varepsilon_e}}{\varepsilon_e}\right) \tag{38}$$

$$\varepsilon_{Sc_t} = \frac{1}{2}Sc_t\left(\frac{\varepsilon_k}{k} - \frac{\varepsilon_k}{\varepsilon} - \frac{\varepsilon_{k_f}}{k_f} + \frac{\varepsilon_{\varepsilon_f}}{\varepsilon_f}\right) \tag{39}$$

It follows that the errors in thermal conductivity and mass diffusivity originally defined in Eqn. 23 will now include additional terms:

$$\varepsilon_{k_t} = k_t\left(\frac{\varepsilon_{\mu_t}}{\mu_t} - \frac{\varepsilon_{\mathrm{Pr}_t}}{\mathrm{Pr}_t}\right) \qquad \varepsilon_{\mathcal{D}_t} = \mathcal{D}_t\left(\frac{\varepsilon_{\mu_t}}{\mu_t} - \frac{\varepsilon_{Sc_t}}{Sc_t}\right) \tag{40}$$

These modifications would impact the turbulent residual of the ETE and provide an additional error generation mechanism not present in the current formulation. The net effect of this extension would be to produce a more accurate prediction of the error in temperature and species distribution in the domain.

8.3. Multiphase Flows

CFD is increasingly being applied to problems involving more complex physical phenomena such as gas/liquid or gas/solid flows. The prediction of sprays, particle-laden flows, and free surface flows require additional submodels and/or transport equations, coupled to the conservation equations for the carrier phase in the case of dispersed multiphase flows. Error prediction for such problems would be beneficial in assessing the validity of new physical models by isolating mesh-induced errors. Potential extensions to the error transport method include treatments for the source terms that provide for mass, momentum, and energy transfer between the carrier and dispersed phases, or treatments for assessing errors at a gas/liquid interface.

8.4. Transient Applications

The temporal evolution of spatial errors remains an area of challenging research. Research to date has not attempted to address temporal discretization errors, which may have a notable effect. At issue is how to separate errors due to the time-stepping scheme from errors caused by the mesh discretization when examining the spatial variation in the CFD solution. Transient flows, particularly LES, may involve an assessment of mean and fluctuating quantities, such as root mean square (RMS) pressure or velocity. In the future, the function library could include options for transient cases.

A natural extension of this problem class is to account for problems with moving or deforming boundaries. The additional aspect from a numerical perspective is the geometric conservation law (GCL). In problems involving moving meshes, the inviscid flux contains an additional term to account for the volume swept out by control volume faces in a given time step. Errors in this flux would certainly be an important element of error quantification for store release predictions, control surface deflections, missile staging, and similar applications.

9. Summary and Concluding Remarks

Computational fluid dynamics (CFD) has become an essential element of modern design in numerous disciplines. Increasingly, engineers and scientists using CFD solvers are being asked not merely the question, "What is the answer?" but "How good is your answer?" Answering the latter question with confidence is not easy. In the area of error quantification, little has been done outside of conventional Richardson extrapolation on structured meshes. Yet unstructured grid solvers are prevalent in most organizations using CFD in the design and analysis cycle, and furthermore, the requirement of monotonic behavior often renders Richardson extrapolation inapplicable or impractical. Over the years, many research efforts have attempted to address the issue of error reduction through the use of adaptive mesh refinement (AMR) codes. However, in most cases, the selection of an appropriate error estimate was *ad hoc*, and rarely demonstrated certifiable error reduction, but rather a loose interpretation of grid convergence by inspection.

The research summarized in this chapter has yielded significant advances towards practical analysis tools for automating error quantification and reduction for CFD. Through the development and validation of an error transport equation (ETE) solver and the Error Function Library, CFD analysts now have an ability to quantify mesh-induced errors in their simulations. Error source information was demonstrated as a more reliable means of performing AMR to certify grid convergence and error reduction on a sequence of meshes. Finally, novel rendering techniques permit greater flexibility in interpreting and understanding these error predictions. While there are many facets of error quantification in CFD that remain to be explored, the CRISP CFD code represents a unique and powerful package for solution verification.

Although Richardson extrapolation is the commonly accepted approach for solution verification, it is restrictive in its applicability, and rarely presented in CFD papers. Monotonic behavior may be difficult to achieve, or some solution metrics may behave monotonically though others do not. The examples provided in this work exhibit these difficulties. Creating three, four, or even five grids to verify grid convergence may be acceptable in 2D studies, however for complex 3D flows its practicality is quickly lost. The ETE offers a viable alternative approach to solution verification, particularly for cases where Richardson extrapolation cannot be applied. While the ultimate objective is a single grid error estimation tool, current studies on sequences of meshes quantitatively verified a reduction in the predicted error with grid refinement, whether manual or adaptive. Using the ETE, meaningful trends may even be extracted where only two grids are employed.

Acknowledgments

The Air Force Research Laboratory, Computational Sciences Branch at Wright-Patterson AFB funded portions of this research under contract FA8650-07-C-3703, Phase II SBIR, with Dr. Gregory P. Brooks as technical monitor. This work is also funded by NASA Langley Research Center under contract NNX11CB67C, Phase II SBIR, with Jeremy Hanke and Alaa Elmiligui as technical monitors.

References

1. Barth, T. J., "Recent Developments in High Order K-Exact Reconstruction on Unstructured Meshes," 31st Aerospace Sciences Meeting, AIAA Paper 93-0668, Jan. 1993.
2. Mehta, U. B., "Guide to Credible Computer Simulations of Fluid Flows," *Journal of Propulsion and Power*, Vol. 12, No. 5, pp. 940-948, 1996.
3. Mehta, U. B., "Some Aspects of Uncertainty in Computational Fluid Dynamics Results," *Journal of Fluids Engineering*, Vol. 113, No. 4, pp. 538-543, 1991.
4. American Institute of Aeronautics and Astronautics, "Guide for the Verification and Validation of Computational Fluid Dynamics Simulations," AIAA Guide G-077-1998, 1998.
5. Frietas, C. J., "The Issue of Numerical Uncertainty," *Applied Mathematical Modeling*, Vol. 26, pp. 237-248, 2002.
6. 1st AIAA High Lift Prediction Workshop: http://hiliftpw.larc.nasa.gov/index-workshop1.html
7. 4th AIAA Drag Prediction Workshop: http://aiaa-dpw.larc.nasa.gov/Workshop4/workshop4.html
8. Vassberg, J. C., Tinoco, E. N., Mani, M., Brodersen, O. P., Eisfeld, B., Wahls, R. A., Morrison, J. H., Zickuhr, T., Laflin, K. R., and Mavriplis, D. J., "Summary of the Third AIAA CFD Drag Prediction Workshop," 45th Aerospace Sciences Meeting, AIAA Paper 2007-0260, Jan. 2007.
9. Laflin, K. R., Klausmeyer, S. M., Zickuhr, T., Vassberg, J. C., Wahls, R. A., Morrison, J. H., Brodersen, O. P., Rakowitz, M. E., Tinoco, E. N., and Godard, J.-L., "Data Summary from Second AIAA Computational Fluid Dynamics Drag Prediction Workshop," *Journal of Aircraft*, Vol. 42, No. 5, pp. 1165-1178, 2005.
10. Cavallo, P. A., "Error Transport Methods in Solution Verification – Review, Successes, and Current Challenges," *JANNAF Simulation Credibility Workshop*, Arlington, VA, Apr. 2011.
11. Mattick, S., Brinckman, K. W., Dash, S. M., and Liu, Z., "Improvements in Analyzing High-Speed Fuel/Air Mixing Problems Using Scalar Fluctuation Modeling," 46th Aerospace Sciences Meeting, AIAA Paper 2008-0768, Jan. 2008.
12. Brinckman, K. W., Calhoon, W. H., Jr., and Dash, S. M., "Scalar Fluctuation Modeling for High-Speed Aeropropulsive Flows," *AIAA Journal*, Vol. 45, No. 5, pp. 1036-1046, 2007.
13. Brinckman, K. W., Calhoon, W. H., Jr., Mattick, S., Tomes, J., and Dash, S. M., "Scalar Variance Model Validation for High-Speed Variable Composition Flows," 44th Aerospace Sciences Meeting, AIAA Paper 2006-0715, Jan. 2006.
14. Calhoon, W. H., Jr., Brinckman, K. W., Tomes, J., Mattick, S., and Dash, S. M., "Scalar Fluctuation and Transport Modeling for Application to High Speed Reacting Flows," 44th Aerospace Sciences Meeting, AIAA Paper 2006-1452, Jan. 2006.
15. Zambon, A. C., Ott, J. D., Dash, S. M., and Kennedy, K. D., "Accuracy and Efficiency Enhancements for High-Speed Aero-Propulsive Flows with Combustion," DoD HPC Users Group Meeting, Portland, OR, June 2011.
16. Zambon, A. C., Brinckman, K., Feldman, G., Kenzakowski, D. C., Dash, S. M., and Kennedy, K. D., "Accuracy and Efficiency Enhancements for High-Speed Aero-Propulsive Flows with Combustion," DoD HPC Users Group Meeting, 2012.

17. Danehy, P. M., O'Byrne, S., Cutler, A. D., and Rodriguez, C. G., "Coherent Anti-Stokes Raman Scattering (CARS) as a Probe for Supersonic Hydrogen-Fuel/Air Mixing," JANNAF APS/CS/PSHS/MSS Joint Meeting, Dec. 2003.

18. O'Byrne, S., Danehy, P. M., and Cutler, A. D., "Dual-Pump CARS Thermometry and Species Concentration Measurements in a Supersonic Combustor," 42nd Aerospace Sciences Meeting, AIAA-2004-0710, Jan. 2004.

19. Rodriguez, C., and Cutler, A., "Computational Simulation of a Supersonic-Combustion Benchmark Experiment," 41st Joint Propulsion Conference, AIAA Paper 2005-4424, July 2005.

20. Keistler, P. G., Xiao, X., Hassan, H. A., and Cutler, A. D., "Simulation of the SCHOLAR Supersonic Combustion Experiments," 45th Aerospace Sciences Meeting, AIAA Paper 2007-0835, Jan. 2007.

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

22. Celik, I., Ghia, U., and Roache, P. J., "Procedure for Estimation and Reporting of Uncertainty due to Discretization in CFD Applications," *Journal of Fluids Engineering*, Vol. 130, No. 7, 2008.

23. Baker, T. J., "On the Relationship between Mesh Refinement and Solution Accuracy," 17th Computational Fluid Dynamics Conference, AIAA Paper 2005-4875, June 2005.

24. Coleman, H. W., Stern, F., DiMascio, A., and Campana, E., "The Problem with Oscillatory Behavior in Grid Convergence Studies," *Journal of Fluids Engineering*, Vol. 123, No. 2, pp. 438-439, 2001.

25. Eca, L., Hoekstra, M., Roache, P. J., and Coleman, H. W., "Code Verification, Solution Verification and Validation: an Overview of the 3rd Lisbon Workshop," 19th Computational Fluid Dynamics Conference, AIAA Paper 2009-3647, June 2009.

26. Celik, I., Li, J., Hu, G., and Shaffer, C., "Limitations of Richardson Extrapolation and Some Possible Remedies," *Journal of Fluids Engineering*, Vol. 127, No. 4, pp. 795-805, 2005.

27. Roache, P. J., "Perspective: A Method for Uniform Reporting of Grid Refinement Studies," *Journal of Fluids Engineering*, Vol. 116, No. 3, pp. 405-413, 1994.

28. Roache, P. J., "Quantification of Uncertainty in Computational Fluid Dynamics," *Annual Review of Fluid Mechanics*, Vol. 29, pp. 123-160, 1997.

29. Oberkampf, W. L., and Trucano, T. G., "Verification and Validation in Computational Fluid Dynamics," *Progress in Aerospace Sciences*, Vol. 38, No. 3, pp. 209-272, 2002.

30. Roy, C. J., "Review of Code and Solution Verification Procedures for Computational Simulation," *Journal of Computational Physics*, Vol. 205, No. 1, pp. 131-156, 2005.

31. Cosner, R. R., Oberkampf, W. L., Rahaim, C. P., and Shih, T. I.-P., "AIAA Committee on Standards for Computational Fluid Dynamics – Status and Plans," 42nd Aerospace Sciences Meeting, AIAA Paper 2004-0654, Jan. 2004.

32. Qin, Y., Chi, X., and Shih, T. I.-P., "Estimating Grid-Induced Errors in Navier-Stokes Solutions by Euler Discrete-Error-Transport Equations," 43rd Aerospace Sciences Meeting, AIAA Paper 2005-0567, Jan. 2005.

33. Qin, Y., Keller, P. S., Sun, R. L., Hernandez, E. C., Perng, C.-Y., Trigui, N., Han, Z., Shen, F. Z., Shieh, T., and Shih, T. I.-P., "Estimating Grid-Induced Errors in CFD by Discrete-Error-Transport Equations," 42nd Aerospace Sciences Meeting, AIAA Paper 2004-0656, Jan. 2004.

34. Qin, Y., and Shih, T. I.-P., "Analysis and Modeling of the Residual in the Discrete-Error-Transport Equation," 16th Computational Fluid Dynamics Conference, AIAA Paper 2003-3850, June 2003.

35. Qin, Y., and Shih, T. I.-P., "A Method for Estimating Grid-Induced Errors in Finite-Difference and Finite-Volume Methods," 41st Aerospace Sciences Meeting, AIAA Paper 2003-0845, Jan. 2003.

36. Qin, Y., and Shih, T. I.-P., "A Discrete Transport Equation for Error Estimation in CFD," 40th Aerospace Sciences Meeting, AIAA Paper 2002-0906, Jan. 2002.

37. Celik, I., and Hu, G., "Single Grid Error Estimation Using Error Transport Equation," *Journal of Fluids Engineering*, Vol. 126, No. 3, pp. 778-790, 2004.

38. Celik, I., Hu, G., and Badeau, A., "Further Refinement and Bench Marking of a Single-Grid Error Estimation Technique," 41st Aerospace Sciences Meeting, AIAA Paper 2003-0628, Jan. 2003.

39. Celik, I., Karaismail, E., Blancas, F. E., Parsons, D., and Sezer, H., "Error Estimation Using Hybrid Methods," American Society of Mechanical Engineers (ASME) Fluids Engineering Division Summer Meeting, July 2012.

40. Zhang, X. D., Trepanier, J.-Y., and Camarero, R., "An A Posteriori Error Estimation Method Based on Error Equations," 13th Computational Fluid Dynamics Conference, AIAA Paper 97-1889, June-July 1997.

41. Zhang, X. D., Trepanier, J.-Y., and Camarero, R., "A Posteriori Error Estimation for Finite-Volume Solutions of Hyperbolic Conservation Laws," *Computer Methods in Applied Mechanics and Engineering*, Vol. 185, pp. 1-19, 2000.

42. Ilinca, C., Zhang, X. D., Trepanier, J.-Y., and Camarero, R., "A Comparison of Three Error Estimation Techniques for Finite-Volume Solutions of Compressible Flows," *Computer Methods in Applied Mechanics and Engineering*, Vol. 189, pp. 1277-1294, 2000.

43. Zhang, X. D., Vallet, M.-G., Dompierre, J., Labbe, P., Pelletier, D., Trepanier, J.-Y., Camarero, R., Lassaline, J. V., Manzano, L. M., and Zingg, D. W., "Mesh Adaptation using Different Error Indicators for the Euler Equations," 15th Computational Fluid Dynamics Conference, AIAA Paper 2001-2549, June 2001.

44. Van Straalen, B. P., Simpson, R. B., and Stubley, G. D., "A Posteriori Error Estimation for Finite Volume Simulations of Fluid Flow Transport," *Proceedings of the 3rd Annual Conference of the CFD Society of Canada*, June 1995.

45. Hay, A., and Visonneau, M., "Error Estimation Using the Error Transport Equation for Finite-Volume Methods and Arbitrary Meshes," *International Journal of Computational Fluid Dynamics*, Vol. 20, No. 7, pp. 463-479, 2006.

46. Wilson, R. V., and Stern, F., "Verification and Validation for RANS Simulation of a Naval Surface Combatant," 38th Aerospace Sciences Meeting, AIAA Paper 2000-0904, Jan. 2000.

47. Tong, X.-L., and Luke, E. A., "An Error Transport Equation in Primitive Variable Formulation," 49th Aerospace Sciences Meeting, AIAA Paper 2011-1295, Jan. 2011.

48. Anderson, D. A., Tannehill, J. C., and Pletcher, R. H. *Computational Fluid Mechanics and Heat Transfer*, New York: Hemisphere Publishing Corp., pp. 89-91, 1984.

49. Cavallo, P. A., and Sinha, N., "An Error Transport Equation with Practical Applications," 18th Computational Fluid Dynamics Conference, AIAA Paper 2007-4092, June 2007.

50. Cavallo, P. A., and Sinha, N., "Error Quantification for Computational Aerodynamics Using an Error Transport Equation," *Journal of Aircraft*, Vol. 44, No. 6, pp. 1954-1963, 2007.

51. Cavallo, P. A., Sinha, N., and O'Gara, M. R., "Viscous Error Transport Equation for Error Quantification of Turbulent Flows," 38th AIAA Fluid Dynamics Conference, AIAA Paper 2008-3851, June 2008.

52. Cavallo, P. A., O'Gara, M. R., and Schikore, D. R., "Software System for Prediction, Visualization, Analysis, and Reduction of Errors In CFD Simulations," 19th AIAA Computational Fluid Dynamics Conference, AIAA Paper 2009-3649, June 2009.

53. Cavallo, P. A., O'Gara, M. R., Feldman, G. M., and Liu, Z., "Unified Error Transport Solver for Solution Verification on Unstructured Grids," 42nd AIAA Fluid Dynamics Conference, AIAA Paper 2012-3345, June 2012.

54. CGNS, CFD General Notation System: http://cgns.github.io

55. Fox, R. W., and McDonald, A. T. *Introduction to Fluid Mechanics*, 3rd ed. New York: John Wiley and Sons, pp. 718-719, 1985.

56. American Institute of Aeronautics and Astronautics, "Assessing Experimental Uncertainty – Supplement to AIAA S-071A-1999," AIAA Guide G-045-2003, 2003.

57. Walatka, P. P., Buning, P. G., Pierce, L., and Elson, P. A., "PLOT3D User's Manual," NASA TM-101067, Mar. 1990.

58. Pang., A.T., Wittenbrink, C.M., and Lodha, S.K., "Approaches to Uncertainty Visualization," *The Visual Computer*, Vol. 13, No. 8, pp. 370-390, 1997.

59. Wittenbrink, C. M., Saxon, E., Furman, J. J., Pang., A., and Lohda, S., "Glyphs for Visualizing Uncertainty in Environmental Vector Fields," *IEEE Trans. Visualization and Computer Graphics*, Vol. 2, No. 3, pp. 266-279, 1996.

60. Lodha, S. K., Pang, A., Sheehan, R.E., and Wittenbrink, C. M., "UFLOW: Visualizing Uncertainty in Fluid Flow," *Proceedings of the 7th Conference on Visualization '96*, IEEE CS Press, pp. 249-254, 1996.

61. EnSight Software: http://www.ensight.com

62. Cavallo, P. A., and Grismer, M. J., "A Parallel Adaptation Package for Three-Dimensional Mixed-Element Unstructured Meshes," *Journal of Aerospace Computing, Information, and Communication,* Vol. 2, No. 11, pp. 433-451, 2005.

63. Cavallo, P. A., Sinha, N., and Feldman, G. M., "Parallel Mesh Adaptation Method for Moving Body Applications," *AIAA Journal,* Vol. 43, No. 9, pp. 1937-1945, 2005.

64. Schloegel, K., Karypis, G., and Kumar, V., "A Unified Algorithm for Load-Balancing Adaptive Scientific Simulations," Technical Report, 00-033, University of Minnesota, Department of Computer Science and Engineering, 2000.

65. Baker, T. J., "Interpolation from a Cloud of Points," *Proceedings of the 12th International Meshing Roundtable*, Sept. 2003.

66. Zhang, X. D., Vallet, M.-G., Dompierre, J., Labbe, P., Pelletier, D., Trepanier, J.-Y., Camarero, R., Lassaline, J. V., Manzano, L. M., and Zingg, D. W., "Mesh Adaptation using Different Error Indicators for the Euler Equations," 15th Computational Fluid Dynamics Conference, AIAA Paper 2001-2549, June 2001.

67. Liu, Z., Brinckman, K., and Dash, S. M., "Automated Validation of CFD Codes for Analysis of Scramjet Propulsive Flows Using CRAVE," 45th Joint Propulsion Conference, AIAA Paper 2009-4846, Aug. 2009.

68. Nakayama, A., "Characteristics of the Flow Around Conventional and Supercritical Airfoils," *Journal of Fluid Mechanics*, Vol. 160, pp. 155-179, Nov. 1985.

69. Driver, D. M., and Seegmiller, H .L., "Features of a Reattaching Turbulent Shear Layer in Divergent Channel Flow," *AIAA Journal*, Vol. 23, No. 2, pp. 163-171, Feb. 1985.

70. Hosangadi, A., Lee, R. A., Cavallo, P. A., Sinha, N., and York, B. J., "Hybrid, Viscous, Unstructured Mesh Solver for Propulsive Applications," 34th Joint Propulsion Conference, AIAA Paper 98-3153, July 1998.

71. Strang, W. Z., Tomaro, R. F., and Grismer, M. J., "The Defining Methods of $Cobalt_{60}$: A Parallel, Implicit, Unstructured Euler/Navier-Stokes Flow Solver," 37th Aerospace Sciences Meeting, AIAA Paper 99-0786, Jan. 1999.

72. Cavallo, P. A., O'Gara, M. R., and Kenzakowski, D. C., "Assessment of Grid-Induced Errors in HiLiftPW-1 Predictions Using Error Transport Equations," presented at the 1st AIAA High Lift Prediction Workshop, Chicago, IL, June 27, 2010. http://hiliftpw.larc.nasa.gov/Workshop1/ParticipantTalks/cavallo-craft.pdf

73. FUN3D Manual, National Aeronautics and Space Administration: http://fun3d.larc.nasa.gov

74. Frink, N. T., "Tetrahedral Unstructured Navier-Stokes Method for Turbulent Flows," *AIAA Journal*, Vol. 36, No. 11, pp. 1975-1982, 1998.

75. Cavallo, P. A., and Feldman, G. M., "Quantification of Grid Refinement Effects for NASA High Lift Trap Wing Using Error Transport Model," 30th Applied Aerodynamics Conference, AIAA Paper 2012-2847, June 2012.

76. Sturek, W. B., Birch. T., Lauzon, M., Housh, C., Manter, J., Josyula, E., and Soni, B., "The Application of CFD to the Prediction of Missile Body Vortices," 35th Aerospace Sciences Meeting and Exhibit, AIAA Paper 97-0637, Jan. 1997.

77. Papp, J. L., Kenzakowski, D. C., and Dash, S. M., "Calibration and Validation of EASM Turbulence Model for Jet Flowfields," 40th Aerospace Sciences Meeting, AIAA Paper 2002-0855, Jan. 2002.

78. Baker, T. J., "Mesh Adaptation Strategies for Problems in Fluid Dynamics," *Finite Elements in Analysis and Design*, Vol. 25, No. 3-4, pp. 243-273, 1997.

79. Kamotani, Y., and Greber, I., "Experiments on a Turbulent Jet in a Cross Flow," *AIAA Journal*, Vol. 10, No. 11, pp. 1425-1429, 1972.

Toward Automatic Verification of Goal-Oriented Flow Simulations

Marian Nemec[*] and **Michael J. Aftosmis**[†]

[*] *Science & Technology Corp., Moffett Field, CA 94035*

[†] *NASA Ames Research Center, Moffett Field, CA 94035*

We demonstrate the power of adaptive mesh refinement with adjoint-based error estimates in veri-fication of simulations governed by the steady Euler equations. The flow equations are discretized using a finite volume scheme on a Cartesian mesh with cut cells at the wall boundaries. The dis-cretization error in selected simulation outputs is estimated using the method of adjoint-weighted residuals. Practical aspects of the implementation are emphasized, particularly in the formulation of the refinement criterion and the mesh adaptation strategy. Following a thorough code verification example, we demonstrate simulation verification of two- and three-dimensional problems. These involve an airfoil performance database, a pressure signature of a body in supersonic flow, and a launch abort with strong jet interactions. The results show reliable estimates and automatic control of discretization error in all simulations at an affordable computational cost. Moreover, the approach remains effective even when theoretical assumptions, e.g., steady-state and solution smoothness, are relaxed.

Nomenclature

A	Face area
C	Constant used in estimating discretization error, $C = 2$ or $\frac{4}{3}$
$\mathbf{d}$	Distance vector from cell centroid to face centroid
$\mathscr{E}$	Discretization error
E	Total energy per unit mass
e	Discretization error with respect to a uniformly refined mesh
$\mathbf{F}$	Inviscid flux tensor
$\mathbf{G}$	Numerical flux function
J	Scalar functional or output, e.g., lift coefficient
$\hat{\mathbf{n}}$	Outward pointing unit normal
N	Number of cells
$\mathbf{P}$	Prolongation operator or matrix

p	Pressure
$\mathbf{Q}$	Flow solution vector of conservative variables $[\rho, \rho u, \rho v, \rho w, \rho E]^{\mathrm{T}}$
$\mathbf{R}$	Vector of residuals
t	Time
$\mathbf{U}$	Flow solution vector of primitive variables $[\rho, u, v, w, p]^{\mathrm{T}}$
u, v, w	Cartesian components of velocity
V	Volume

Greek letters

η	Error indicator
ρ	Density
ψ	Adjoint vector

Superscripts

a	Adjoint
H	Data reconstruction from mesh with characteristic cell size H

Subscripts

H	Discretization on mesh with characteristic cell size H
h	Discretization on mesh with characteristic cell size $h = \frac{1}{2}H$
c	Adjoint correction
L	Linear interpolant
TL	Trilinear interpolant
TQ	Triquadratic interpolant
w	Wall
∞	Freestream

1. Introduction

> *There is no doubt that mesh adaptation can lead to significant improvement in solution accuracy. ... What remains in doubt is whether the current methods of mesh adaptation can be brought to a sufficient level of reliability and robustness for routine use as a predictive tool.*
>
> T. Baker, 1997 [1]

Despite significant progress in solution-adaptive mesh refinement [1, 2, 3, 4, 5], verification of flow simulations remains largely a manual procedure that requires expert guidance [6, 7, 8]. Most of the time is consumed by crafting a computational mesh, checking the results, and refining the mesh to assess and control discretization error[1]. As the number and complexity of simulations increase (for example, consider flight vehicle performance databases involving 10^3–10^5 cases), manual simulation verification becomes impractical. Instead, simulations are verified for only a subset of cases and involve "best-practice" guidelines, which do not guarantee that the results comply with the expected standards of accuracy [9, 10].

[1] The subject of validation (modeling error) is not considered and in our experience, discretization error dominates other aspects of simulation verification, such as iterative convergence.

The numerical accuracy of flow simulations depends inextricably on discretization error. In other words, numerical inaccuracy is a consequence of the mesh. Establishing credibility of complex simulations, therefore, requires that mesh generation and error estimation be integral parts of each simulation. As the mesh and the flow solution evolve, a systematic reduction in discretization error is achieved through use of error estimates derived from the current flow solution. The benefit is a straightforward quantitative assessment of convergence because a mesh refinement study is intrinsic to every case.

The goal of most engineering simulations is to predict a handful of outputs — for example, aerodynamic forces and moments. In such goal-oriented simulations, it is most efficient to focus on discretization error directly affecting the outputs of interest. For example, even the simple problem of predicting the span efficiency factor of an isolated wing in subsonic inviscid flow becomes prohibitively expensive if the mesh is refined to follow the tip vortex far downstream. As the influence of the vortex on span efficiency decreases with downstream distance, so should the cell refinement. Experience with error estimates that do not target outputs — in particular, direct residual or truncation-error estimates — shows that the adapted meshes are frequently inferior to those crafted by experts who understand the goals of the simulation.

Remarkably, a relatively straightforward modification to residual error estimates allows the prediction of error in outputs: the cell-wise residuals are weighted by their influence on the output. This is the idea behind the method of adjoint-weighted residuals, where the weights are obtained from the solution of an adjoint equation. The result is not only an estimate of error in the outputs due to discretization (for example the error in lift), but also a cell-wise error indicator to guide mesh refinement. The adjoint-weighted approach was first developed within the framework of the finite element method [11, 12] and extended to finite volume methods by Giles and Pierce [13], Barth [14], and Venditti and Darmofal [15]. The approach has been steadily refined to improve its accuracy and efficiency [16, 17, 18, 19, 20, 21, 22], and has been used successfully to establish the credibility of goal-oriented simulations [23, 24, 25, 26, 27].

The routine use of adjoint-based error estimates for automatic simulation verification, however, is predicated on robust mesh generation. This is because failures in mesh generation often require expert intervention to resolve. If a typical simulation requires five to ten adaptation cycles to attain sufficient output accuracy, then the construction of an aerodynamic performance database for a moderate range of operating conditions may invoke the mesh generator ten thousand times. Therefore, the mesh generator must be fast and failsafe for automatic verification to be viable in an engineering environment. One such robust approach is the embedded-boundary (cut-cell) method, where the mesh is constructed by embedding the geometry in a regular lattice of hexahedral (Cartesian) or tetrahedral elements [28, 29, 30, 31, 23, 32, 33].

The purpose of this work is to demonstrate that the combination of adjoint-based error estimates with a Cartesian cut-cell method is a practical approach for automatic verification of steady, goal-oriented simulations. We cover the development and implementation of a simulation verification framework previously described in [34, 35, 36] with additional details and improvements. The framework uses the approach of Venditti and Darmofal [15] to formulate reliable error estimates and the approach of Aftosmis and Berger [37] for incremental refinement of nested Cartesian cut-cell meshes. The framework emphasizes robustness and efficiency, in terms of both execution speed and memory requirements, because both are central considerations in engineering and decision-making.

We begin with a brief review of discretization error in Section 2. An estimate of the error in user-selected outputs is expressed in terms of adjoint-weighted residuals using the algebraic formulation

of [15]. Section 3 presents the salient features of the flow solver, the formulation of the discrete adjoint equation, and the implementation of the adjoint solver. Section 4 explains the details of the error estimation procedure and the formulation of a robust refinement criterion. The simulation verification framework is presented in Section 5, including a discussion of the adaptation mechanics, error control, and practical aspects of the implementation. The results are organized in two parts. Section 6 presents a code verification example that establishes the accuracy of the error estimate for guiding mesh refinement. Section 7 demonstrates examples of automatic simulation verification on a sequence of three problems of increasing difficulty. Additional examples of simulation verification through use of this framework can be found in [38, 39, 40, 41, 42, 43, 44, 45].

2. Error Estimates

2.1. Discretization Error

Our goal is to compute a reliable approximation of a scalar output functional $J(\mathbf{Q})$, for example lift or drag, derived from a flow solution $\mathbf{Q}$ that satisfies the flow equations:

$$\mathbf{R}(\mathbf{Q}) = \mathbf{0} \tag{1}$$

such as the Euler or Navier–Stokes equations. To compute a discrete approximation of the functional $J_H(\mathbf{Q}_H)$, the domain is tessellated into N control volumes with characteristic cell-size H, which we call the "working" mesh. The flow equations are discretized and solved to satisfy a system of modified partial differential equations:

$$\mathbf{R}_H(\mathbf{Q}_H) = \mathbf{0} \tag{2}$$

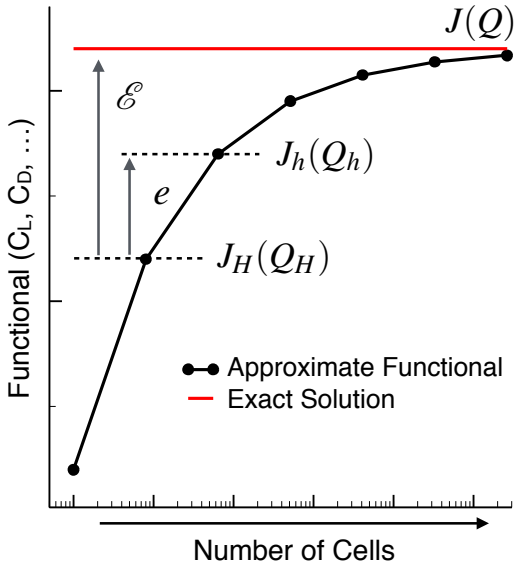


Figure 1. Example of a uniform mesh refinement study showing the definition of discretization error, $\mathscr{E}$, and the error relative to the next mesh, e.

where $\mathbf{Q}_H = [\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_N]^{\mathrm{T}}$ is the discrete flow solution vector, e.g., an algebraic vector of cell-average values, and the discrete operator $\mathbf{R}_H$ represents the residual vector. Similarly, J_H represents the discrete operator used to evaluate scalar functionals, e.g., the integration of pressure to obtain lift given the flow solution on the working mesh $\mathbf{Q}_H$.

The error in the functional due to discretization is:

$$\mathscr{E} = |J(\mathbf{Q}) - J_H(\mathbf{Q}_H)| \tag{3}$$

This is illustrated in Figure 1, which shows a notional mesh refinement study on a sequence of nested, uniformly refined meshes. We assume that errors not related to H, such as distance to farfield, are negligible. If the discretization is consistent, then the approximation $J_H(\mathbf{Q}_H)$ converges to the exact solution $J(\mathbf{Q})$ as the number of cells in the computational domain is increased and the characteristic cell-size H shrinks. In practice, direct computation of the discretization error is difficult because it involves the analytic partial differential equations, Eq. 1.

An alternate approach is to compute the error relative to the functional on the next nested mesh:

$$e = |J_h(\mathbf{Q}_h) - J_H(\mathbf{Q}_H)| \tag{4}$$

as shown in Figure 1. We refer to the next uniformly refined mesh as the "embedded" mesh with characteristic cell-size h. Assuming that the problem is smooth, the discretization error $\mathscr{E}$ can be expressed as a geometric series in e. For example, the error expression for a functional with second-order convergence is:

$$\mathscr{E} = \sum_{i=0}^{\infty} \frac{1}{4^i} e = \frac{4}{3} e \tag{5}$$

and $\mathscr{E} = 2e$ for first-order functionals[2]. This trades the need for the exact solution $J(\mathbf{Q})$ for the requirement that the functional must be in the asymptotic range with a known convergence rate. Put another way, the starting mesh should be sufficiently fine. The key step becomes approximating $J_h(\mathbf{Q}_h)$ without solving on the embedded mesh.

2.2. Method of Adjoint-Weighted Residuals

To derive a reliable approximation of the functional $J_h(\mathbf{Q}_h)$, consider its truncated Taylor series expansion about the working-mesh solution:

$$J_h(\mathbf{Q}_h) \approx J_h(\mathbf{Q}_h^H) + \frac{\partial J_h(\mathbf{Q}_h^H)}{\partial \mathbf{Q}_h} \left(\mathbf{Q}_h - \mathbf{Q}_h^H\right) \tag{6}$$

The algebraic vector $\mathbf{Q}_h^H$ denotes a reconstruction of the flow solution from the working mesh to the embedded mesh via a prolongation operator, $\mathbf{Q}_h^H = \mathbf{P}\mathbf{Q}_H$. The term $J_h(\mathbf{Q}_h^H)$ is the evaluation of the functional using the reconstructed flow solution on the embedded mesh, e.g., lift computation using the reconstructed state and finer boundary resolution. This is usually straightforward. The challenge is the explicit dependence on $\mathbf{Q}_h$ in the inner-product term of Eq. 6.

To eliminate $\mathbf{Q}_h$, expand the residual equation to obtain:

$$\mathbf{R}_h(\mathbf{Q}_h) = \mathbf{0} \approx \mathbf{R}_h(\mathbf{Q}_h^H) + \frac{\partial \mathbf{R}_h(\mathbf{Q}_h^H)}{\partial \mathbf{Q}_h} \left(\mathbf{Q}_h - \mathbf{Q}_h^H\right) \tag{7}$$

Note that Eqs. 6 and 7 are approximate. The derivation can be made exact through use of the mean-value linearization; however, this only defers the use of similar approximations to obtain computable error estimates. Combining Eqs. 6 and 7 gives:

$$J_h(\mathbf{Q}_h) \approx J_h(\mathbf{Q}_h^H) - \frac{\partial J_h(\mathbf{Q}_h^H)}{\partial \mathbf{Q}_h} \left[\frac{\partial \mathbf{R}_h(\mathbf{Q}_h^H)}{\partial \mathbf{Q}_h}\right]^{-1} \mathbf{R}_h(\mathbf{Q}_h^H) \tag{8}$$

which is independent of $\mathbf{Q}_h$. The adjoint equation is obtained from Eq. 8 by defining the following intermediate product:

$$\left[\frac{\partial \mathbf{R}_h(\mathbf{Q}_h^H)}{\partial \mathbf{Q}_h}\right]^{\mathrm{T}} \psi_h = \frac{\partial J_h(\mathbf{Q}_h^H)}{\partial \mathbf{Q}_h}^{\mathrm{T}} \tag{9}$$

where the vector ψ denotes the adjoint variables. Rewriting Eq. 8 with the adjoint variables:

$$J_h(\mathbf{Q}_h) \approx J_h(\mathbf{Q}_h^H) - \psi_h^{\mathrm{T}}\, \mathbf{R}_h(\mathbf{Q}_h^H) \tag{10}$$

[2]The sum of a geometric series $1 + r + r^2 + \ldots$ is $\frac{1}{1-r}$ if $|r| < 1$. The common ratio r is $1/4$ for second-order functionals and $1/2$ for first-order functionals.

reveals that the adjoints weight the residual errors to form a correction term to approximate the functional on the embedded mesh. Substituting Eq. 10 into Eq. 4, the error expression (Eq. 5) becomes:

$$\mathscr{E} \approx C|J_h(\mathbf{Q}_h^H) - \psi_h^{\mathrm{T}}\, \mathbf{R}_h(\mathbf{Q}_h^H) - J_H(\mathbf{Q}_H)| \tag{11}$$

where the constant $C = \frac{4}{3}$ for second-order functionals and $C = 2$ for first-order functionals.

While Eq. 11 is independent of $\mathbf{Q}_h$, it does require the solution of the adjoint equation on the embedded mesh (ψ_h of Eq. 9). This is impractical because a solution of the large, linear adjoint system can be nearly as expensive as a nonlinear flow solution. Various strategies exist to circumvent this difficulty. The initial step involves solving the adjoint system on the working mesh:

$$\left[\frac{\partial \mathbf{R}_H(\mathbf{Q}_H)}{\partial \mathbf{Q}_H}\right]^{\mathrm{T}} \psi_H = \frac{\partial J_H(\mathbf{Q}_H)}{\partial \mathbf{Q}_H}^{\mathrm{T}} \tag{12}$$

This solution is then prolonged to the embedded mesh to estimate ψ_h. The estimate can be sharpened by additional (implementation-specific) procedures, such as relaxation. To explain the salient features of our approach, we first introduce the governing equations and the numerical method, and then return to evaluation of Eq. 11 in Section 4.

3. Governing Equations and Numerical Method

3.1. Flow Equations

We solve the three-dimensional Euler equations governing compressible flow of a perfect gas. For a finite region of space with volume V and surface area A, the integral form of the Euler equations is given by:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_V \mathbf{Q}\,\mathrm{d}V + \oint_A \mathbf{F}\cdot\hat{\mathbf{n}}\,\mathrm{d}A = \mathbf{0} \tag{13}$$

where $\mathbf{Q} = [\rho, \rho u, \rho v, \rho w, \rho E]^{\mathrm{T}}$, $\mathbf{F}$ is the inviscid flux tensor, and $\hat{\mathbf{n}}$ is the outward-pointing unit normal vector.

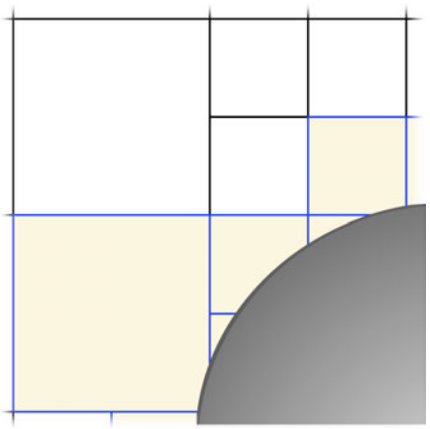


Figure 2. A multilevel Cartesian mesh with a cut-cell boundary. Adjacent cells cannot exceed 2:1 ratio.

The Euler equations are solved with a finite-volume method on a regular Cartesian mesh with embedded boundaries. The body geometry is specified by a watertight surface triangulation. The volume mesh consists of hexahedral cells, except for a layer of body-intersecting cells, or cut cells, which are arbitrary polyhedra adjacent to the boundaries, as illustrated in Figure 2. The mesh is viewed as an unstructured collection of control volumes to facilitate solution-adaptive refinement.

Spatial discretization uses a cell-centered approach, where the control volumes V correspond to the mesh cells and the cell-averaged value of $\mathbf{Q}$, denoted by $\mathbf{Q}_H$, is located at the centroid of each cell. The control volumes are fixed in time. The semi-discrete form is given by:

$$\mathbf{V}_H \frac{\mathrm{d}\mathbf{Q}_H}{\mathrm{d}t} + \mathbf{R}_H(\mathbf{Q}_H) = \mathbf{0} \tag{14}$$

where $\mathbf{V}_H$ is a diagonal matrix containing the cell volumes. The residual in each cell i is expressed as:

$$\mathbf{R}_i = \sum_{\mathrm{j}\in\mathrm{V_i}} \mathbf{G}_j \cdot \hat{\mathbf{n}}_j A_j \tag{15}$$

where j denotes the jth face of volume V_i with area A, and $\mathbf{G}$ represents the numerical flux function.

Residual evaluation for a second-order accurate discretization proceeds by linearly reconstructing the solution to the face centroid. This is illustrated in Figure 3, for two neighboring Cartesian cells l, r sharing a common face. Primitive variables, $\mathbf{U} = [\rho, u, v, w, p]^{\mathrm{T}}$, are used for the reconstruction, and the left and right states are given by:

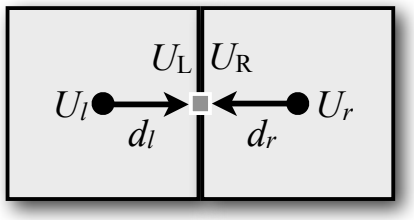


Figure 3. Reconstruction of face midpoint value from cell centroids.

$$\begin{aligned} \mathbf{U}_{\mathrm{L}} &= \mathbf{U}_l + \mathbf{d}_{\mathrm{l}}\, \phi_l \nabla \mathbf{U}_l \\ \mathbf{U}_{\mathrm{R}} &= \mathbf{U}_r - \mathbf{d}_{\mathrm{r}}\, \phi_r \nabla \mathbf{U}_r \end{aligned} \tag{16}$$

Here $\mathbf{d}_l$ and $\mathbf{d}_r$ are the distance vectors from the cell centroids to the face centroid, $\nabla\mathbf{U}$ is the solution gradient determined via a linear least-squares procedure, and ϕ is a vector of slope limiter values used to directionally enforce monotonic solutions [46]. The flux value at the face centroid is obtained via the flux-vector splitting approach of van Leer [47]:

$$\mathbf{G}(\mathbf{U}_{\mathrm{L}}, \mathbf{U}_{\mathrm{R}}) = \mathbf{f}^{+}(\mathbf{U}_{\mathrm{L}}) + \mathbf{f}^{-}(\mathbf{U}_{\mathrm{R}}) \tag{17}$$

At the implementation level, the assembly of the residual vector is accomplished by a loop over the faces of the mesh. The flux contributions are scattered from the face and accumulated in the cells:

$$\mathbf{R}_l = \mathbf{R}_l + \mathbf{G}A \qquad \mathbf{R}_r = \mathbf{R}_r - \mathbf{G}A \tag{18}$$

where the sign reflects the change in the direction of the outward-pointing normal.

All boundary conditions are enforced weakly. At the wall, zero normal velocity is enforced by specifying a wall flux. This flux is non-zero only for the momentum components and uses the pressure at the wall centroid of each cut cell (p_w):

$$\mathbf{G}_{\mathrm{w}} = (0, p_{\mathrm{w}}, p_{\mathrm{w}}, p_{\mathrm{w}}, 0)^{\mathrm{T}} \tag{19}$$

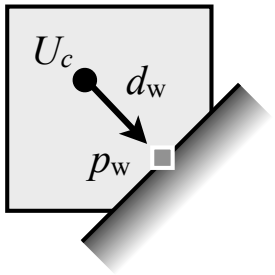


Figure 4. Linear reconstruction of pressure to face centroid at the wall.

as shown in Figure 4. Linear reconstruction, Eq. 16, is used to compute p_w. In the farfield, the flux function, Eq. 17, is used to compute the flux across faces on the boundary. The boundary state (either $\mathbf{U}_{\mathrm{L}}$ or $\mathbf{U}_{\mathrm{R}}$) is set via Riemann invariants, and linear reconstruction (Eq. 16) is used for the interior state.

Steady-state solutions are obtained with a five-stage Runge–Kutta scheme accelerated by local time stepping and full approximation storage multigrid. The multigrid residual restriction operator is the sum of the residuals of the fine mesh cells enclosed by the coarse cell, while the prolongation operator is direct injection. Time-to-solution is further reduced by parallel computing using a highly-scalable

domain decomposition scheme. For details on mesh generation and the flow solution algorithm, see Aftosmis *et al.* [48, 28, 49, 50] and Berger *et al.* [51].

We consider two classes of functionals as primary outputs of interest. The first is aerodynamic performance coefficients, such as coefficients of lift and drag, given by:

$$J = \frac{1}{q_\infty A_{\text{ref}}} \int_{\text{w}} (\hat{\mathbf{n}} \cdot \boldsymbol{\xi})(p_w - p_\infty)\, dA \tag{20}$$

where q_∞ is the freestream dynamic pressure, A_{ref} is the reference area, p_∞ is the freestream pressure, and $\boldsymbol{\xi}$ is the appropriate projection for the coefficient of interest, e.g., $\boldsymbol{\xi} \perp \mathbf{V}_\infty$ for lift. The discretization of Eq. 20 uses midpoint quadrature and is summed over all cut cells. The second class is field functionals that can be specified anywhere in the computational domain. An example is a line sensor for pressure:

$$J_l = \int_0^L \left(\frac{p - p_\infty}{p_\infty} \right)^n d\ell \tag{21}$$

where L is the length of the line and n is a user-specified exponent (usually 1 or 2). The evaluation of line sensors involves finding the set of cells intersected by the line and integrating with midpoint quadrature that uses linear reconstruction of pressure to the line-segment midpoint inside each intersected cell.

3.2. Discrete Adjoint Equation

The adjoint equation as derived in Eqs. 9 and 12 is in discrete form, i.e., the discretized residual and functional operators are linearized. This is a consequence of computing the error relative to the embedded mesh, Eq. 4, instead of the exact solution, Eq. 3. For a reliable estimate of error via Eq. 11, the discrete adjoint solution must converge as the mesh is refined:

$$\lim_{h \to 0} \psi_h \to \psi \tag{22}$$

where ψ is the solution of the analytic adjoint equation obtained by linearizing the flow equations, Eq. 13, and the functional before discretization. In other words, the discretization of the flow equations and the functional must yield an asymptotically consistent adjoint discretization[3].

Adjoint consistency for functionals that involve wall-boundary integrals, such as Eq. 20, is prescribed by the form of the wall flux. Referring to Eq. 19, the transpose of the wall-flux Jacobian is column-rank deficient because the flux is zero for both the continuity and energy equations. For example, the adjoint system at the wall is given by:

$$\begin{bmatrix} 0 & \hat{n}_x \partial p_\rho & \hat{n}_y \partial p_\rho & 0 \\ 0 & \hat{n}_x \partial p_u & \hat{n}_y \partial p_u & 0 \\ 0 & \hat{n}_x \partial p_v & \hat{n}_y \partial p_v & 0 \\ 0 & \hat{n}_x \partial p_p & \hat{n}_y \partial p_p & 0 \end{bmatrix} \begin{bmatrix} \psi^1 \\ \psi^2 \\ \psi^3 \\ \psi^4 \end{bmatrix} = \begin{bmatrix} \partial J_\rho \\ \partial J_u \\ \partial J_v \\ \partial J_p \end{bmatrix} \tag{23}$$

where for clarity we assumed two dimensions and first-order discretization ($p_w = p$), and linearized with respect to primitive variables. This reduces to:

$$\hat{n}_x \psi^2 + \hat{n}_y \psi^3 = \partial J_p \tag{24}$$

[3]For a definition of adjoint consistency see [52, 53]; here we use this term only in the sense of Eq. 22.

which is a well-known analytic adjoint boundary condition derived by recognizing that any variation in wall-normal velocity is zero. Moreover, Eq. 23 shows that the boundary functional should be a function of only pressure, $J = J(p)$. There are no similar restrictions on field functionals; these may be a function of any flow variable.

At the farfield boundary, linearization of the flux function (Eq. 17) in conjunction with the Riemann invariants yields an inconsistent adjoint discretization for subsonic freestream conditions [52]. Since this should have little impact on functional accuracy if the distance to the farfield is sufficiently large, we leave the residuals unchanged. For an example of a duality-preserving formulation, see [53]. To avoid pollution of the error estimates, we omit values from cells adjacent to the farfield boundary and their face neighbors (two layers of cells).

3.3. Adjoint Solver

To solve the adjoint equation, we introduce an unsteady term in Eq. 12 to obtain the following semi-discrete form:

$$\mathbf{V}_H \frac{\mathrm{d}\psi_H}{\mathrm{d}t} + \mathbf{R}^{\mathrm{a}} = \mathbf{0} \tag{25}$$

where the adjoint residual vector is given by:

$$\mathbf{R}^{\mathrm{a}} = \frac{\partial \mathbf{R}_H}{\partial \mathbf{Q}_H}^{\mathrm{T}} \psi_H - \frac{\partial J_H}{\partial \mathbf{Q}_H}^{\mathrm{T}} \tag{26}$$

An important consideration for adjoint solvers is memory usage relative to the flow solver. In general, since the adjoint solver is required to run on the working mesh H, the large matrix-vector product $\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}^{\mathrm{T}} \psi$ introduces a complication. The flow Jacobian $\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}$ is a sparse matrix that is constant during the adjoint solution procedure. Its non-zero entries can be precomputed and stored, thereby minimizing the time needed to compute the matrix-vector product at each iteration. This is an effective strategy when dealing with implicit flow solvers that already store the flow Jacobian [54]. Alternatively, when dealing with explicit solvers, some or all entries of the flow Jacobian can be recomputed when forming the matrix-vector product; see Barth [55], Giles *et al.* [56], Nielsen *et al.* [57] and Mavriplis [58].

We adopt the explicit flow-solution method outlined in Sec. 3.1 for the solution of the adjoint system. The matrix-vector product is recomputed on the fly at each evaluation of the residual by reusing the face-based approach of the flow solver. To demonstrate, consider first-order discretization and examine the update of the flow-solver residual vector from an arbitrary face of the mesh as shown in Figure 3 and Eq. 18:

$$\mathbf{R}_H = \begin{bmatrix} \vdots \\ +\mathbf{G}A \\ -\mathbf{G}A \\ \vdots \end{bmatrix} \begin{matrix} \\ \leftarrow \text{ cell } l \\ \leftarrow \text{ cell } r \\ \\ \end{matrix} \tag{27}$$

Linearize Eq. 27 and apply the transpose operator to obtain:

$$\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}^{\mathrm{T}} \psi = \begin{bmatrix} \ddots & & & \\ & A\frac{\partial \mathbf{G}}{\partial \mathbf{Q}_l}^{\mathrm{T}} & -A\frac{\partial \mathbf{G}}{\partial \mathbf{Q}_l}^{\mathrm{T}} & \\ & A\frac{\partial \mathbf{G}}{\partial \mathbf{Q}_r}^{\mathrm{T}} & -A\frac{\partial \mathbf{G}}{\partial \mathbf{Q}_r}^{\mathrm{T}} & \\ & & & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ \psi_l \\ \psi_r \\ \vdots \end{bmatrix} \tag{28}$$

Therefore, the adjoint residual update when sweeping over the faces of the mesh in a fashion analogous to the flow solver is:

$$\mathbf{R}^{\mathrm{a}}_l = \mathbf{R}^{\mathrm{a}}_l + \frac{\partial \mathbf{f}^+}{\partial \mathbf{Q}_l}^{\mathrm{T}} A\,(\psi_l - \psi_r) \tag{29}$$

$$\mathbf{R}^{\mathrm{a}}_r = \mathbf{R}^{\mathrm{a}}_r + \frac{\partial \mathbf{f}^-}{\partial \mathbf{Q}_r}^{\mathrm{T}} A\,(\psi_l - \psi_r) \tag{30}$$

where we include the linearization of the split fluxes of Eq. 17. Note that the transpose reverses the operator order of the flow solution procedure. For example, linearizing one of the split fluxes:

$$\frac{\partial \mathbf{f}^+}{\partial \mathbf{Q}_l}^{\mathrm{T}} = \left(\frac{\partial \mathbf{U}_l}{\partial \mathbf{Q}_l}\right)^{\mathrm{T}} \left(\frac{\partial \mathbf{f}^+}{\partial \mathbf{U}_l}\right)^{\mathrm{T}} \tag{31}$$

shows that the flux-function linearization is evaluated before the adjoint of the transformation from conservative to primitive variables.

For second-order spatial discretization, the adjoint of the reconstruction procedure (Eq. 16) requires an additional pass over the faces of the mesh. Linearization of the flow gradient involves only the geometry-dependent least-squares weights, which are already computed and stored by the flow solver. Gradient linearization is omitted in cut cells with volume fractions less than single-precision machine epsilon to avoid spurious adjoint values similar to those observed in [59]. Furthermore, the linearization assumes that the limiter values, ϕ in Eq. 16, are independent of the flow solution. In other words, the limiter is treated as a constant. Although this is mostly a pragmatic choice, in [34] we showed that the impact of this simplification on the accuracy of the linearization is relatively small. The right-hand side of Eq. 12 is the linearization of the functional, i.e. Eq. 20 or 21. This linearization involves the pressure reconstruction procedure of Eq. 16 and is exact except for the treatment of the limiter.

Since the eigenvalues of the flow Jacobian matrix are not changed by the transpose operator, we expect Eq. 25 to have similar stability properties as the flow equations (Eq. 14). Convergence to steady-state is accomplished using the same five-stage Runge–Kutta time marching and multigrid schemes of the flow solver. Giles [60, 61] derived conditions for Runge–Kutta time marching and multigrid schemes that ensure the same asymptotic convergence rate of the flow and adjoint solvers. This duality-preserving algorithm is implemented almost automatically, since the flow solver's residual prolongation operator is a transpose of the restriction operator.

Overall, the CPU time per iteration of the adjoint solver is roughly equivalent to the flow solver. This is because the additional cost of re-evaluating the matrix-vector product at each adjoint iteration is offset by reusing the local time step and limiter values directly from the flow solver. The implementation results in only a slight increase in memory usage over the flow solver due to the storage of the converged flow solution and its gradient. Moreover, the face-based data structures and the domain decomposition scheme of the flow solver are reused with only minor modifications; see [34] for details.

4. Error Estimation

With flow and adjoint solutions in hand on the working mesh H, we return to Eq. 10 to estimate the functional on the embedded mesh h. The embedded mesh is constructed explicitly and contains about

$8N$ cells[4]. Computation of the residual $\mathbf{R}_h(\mathbf{Q}_h^H)$ involves reconstructing the flow solution on the embedded mesh from the working mesh data $\mathbf{Q}_h^H = \mathbf{P}\mathbf{Q}_H$ (Eq. 6). The value in each embedded cell is obtained from its parent cell by linear reconstruction (Eq. 16). This is denoted by $\mathbf{Q}_\mathrm{L} = \mathbf{P}_\mathrm{L}\mathbf{Q}_H$, where $\mathbf{P}_\mathrm{L}$ represents the linear prolongation operator. No special treatment is performed at mesh refinement boundaries and cut cells, where irregular stencils pollute the residual on the embedded mesh. Instead, we rely on the adjoint weights to attenuate the residuals together with filters that compensate for these numerical artifacts when tagging cells for refinement (described in Section 5.3).

Several studies [62, 15] show that higher-order reconstruction is more accurate for evaluating the residual on the embedded mesh. Nevertheless, we find that it is difficult to match the robustness of linear reconstruction in practical applications with shocks and other strong nonlinearities. Moreover, the implementation is straightforward since we can reuse functions from the existing flow-solver code.

An estimate for ψ_h in Eq. 10 is obtained through use of the adjoint solution from the working mesh. Similar to $\mathbf{Q}_h^H$, let ψ_h^H represent a reconstructed adjoint solution of Eq. 12 and rewrite Eq. 10 to obtain:

$$J_h(\mathbf{Q}_h) \approx J_h(\mathbf{Q}_h^H) - \underbrace{(\psi_h^H)^\mathrm{T}\mathbf{R}_h(\mathbf{Q}_\mathrm{L})}_{\text{Adjoint Correction}} - \underbrace{(\psi_h - \psi_h^H)^\mathrm{T}\mathbf{R}_h(\mathbf{Q}_\mathrm{L})}_{\text{Remaining Error}} \tag{32}$$

This manipulation yields a computable *adjoint correction* term that is usually non-zero in finite-volume methods. It can be used directly to obtain a better estimate of the functional or to obtain an error estimate via Eq. 11, as long as the last term, the *remaining error*, is small. This is likely, because the remaining error is a higher-order term. Moreover, its magnitude can be controlled with adaptive mesh refinement. Note that we tacitly assume that all higher derivatives of the functional and residual equations are also small. The crux becomes finding a reliable estimate of $\psi_h - \psi_h^H$ to formulate a robust adaptation criterion.

We use a trilinear interpolant for constructing ψ_h^H and a triquadratic interpolant for approximating ψ_h. These interpolants are based on shape functions for "brick" elements commonly used in finite-element methods. The use of interpolation for ψ_h is a compromise between accuracy, cost, and factors related to implementation and maintenance. On one hand, interpolation reduces the quality of the remaining-error estimate because it models the solution error $\psi_h - \psi_H$ with an interpolation error that primarily detects solution nonlinearity. On the other hand, this approach maximizes speed since the number of arithmetic operations on the embedded mesh is relatively small. Furthermore, in contrast to the flow reconstruction, the adjoint reconstruction is not followed by residual evaluation, which relaxes robustness constraints in the implementation.

Before constructing the interpolants, the adjoint solution is linearly reconstructed from the centroid to the vertices of each cell on the working mesh, including cut cells. We use Eq. 16, and each vertex receives contributions from all its coincident cells. The average of all contributions determines the vertex solution value. Put another way, this is a data-smoothing step. Special logic is implemented at mesh refinement boundaries, where the hanging vertices of small cells need additional updates from their big-cell neighbors. The solution at the eight vertices of each working-mesh hexahedron is used to form a unique trilinear polynomial:

$$\psi_\mathrm{TL} = c_0 + c_1 x + c_2 y + c_3 z + c_4 xy + c_5 xz + c_6 yz + c_7 xyz \tag{33}$$

[4]Nested subdivision of a hexahedron creates eight embedded hexahedra, but at cut cells some children may be inside the geometry.

The triquadratic reconstruction operator is given by:

$$\begin{aligned}\psi_{\mathrm{TQ}} = c_0 + c_1 x + c_2 y + c_3 z + c_4 xy + c_5 xz + c_6 yz + c_7 xyz + c_8 x^2 \\ + c_9 y^2 + c_{10} z^2 + c_{11} x^2 y + c_{12} x^2 z + c_{13} xy^2 + c_{14} xz^2 + c_{15} y^2 z \\ + c_{16} yz^2 + c_{17} x^2 yz + c_{18} xy^2 z + c_{19} xyz^2 \end{aligned} \quad (34)$$

To determine the 20 unknown coefficients, we use the eight solution values from the trilinear case in conjunction with the solution gradient at the vertices. The gradient value at a vertex is determined by the arithmetic average of all gradients from cells common to the vertex. The Barth-Jespersen limiter [63] is used to prevent oscillatory reconstruction. The resulting over determined system of 32 equations is solved in a least-squares sense. A well-behaved triquadratic interpolant is ensured by the addition of safeguards. These involve monitoring solution differences between the triquadratic, trilinear and cell-centroid values, and using the lower-order values when large differences are detected.

We split Eq. 32 into an estimate for the corrected functional:

$$J_c = J_h(\mathbf{Q}_{\mathrm{L}}) - \psi_{\mathrm{TQ}}^{\mathrm{T}} \mathbf{R}_h(\mathbf{Q}_{\mathrm{L}}) \quad (35)$$

and a cell-wise estimate of the remaining error in each cell of the working mesh:

$$\eta_H = \sum_{j \in \mathrm{V}_i} (\psi_{\mathrm{TQ}} - \psi_{\mathrm{TL}})^{\mathrm{T}} \mathbf{R}_h(\mathbf{Q}_{\mathrm{L}})_j \quad (36)$$

where j denotes the j^{th} child of parent cell V_i and $\eta_H = [\eta_1, \eta_2, \ldots, \eta_N]^{\mathrm{T}}$. Note that triquadratic reconstruction is used in both the functional correction and remaining error (Eqs. 35 and 36), which is a slight departure from Eq. 32. This is based on the assumption that the triquadratic interpolant is our best estimate of the embedded-mesh adjoint. Hence, it is used not only to compute the remaining error, but also to get an improved functional estimate.

Substituting Eq. 35 into Eq. 11 gives a computable estimate of discretization error on the working mesh:

$$\mathscr{E} \approx C|J_c - J_H(\mathbf{Q}_H)| \quad (37)$$

To define a local quantity suitable for driving adaptive mesh refinement, the remaining error (Eq. 36) is localized to form an error indicator $|\eta|_H = [|\eta_1|, |\eta_2|, \ldots, |\eta_N|]^{\mathrm{T}}$. The sum of the error-indicator values over the cells of the working mesh gives a bound on the estimate of the remaining error:

$$\eta = \sum_{i=1}^{N} |\eta_i| \quad (38)$$

Since the absolute value operator prevents cell-wise error cancellation, this bound is quite conservative. In fact, when dealing with difficult simulations containing non-smooth flow and arbitrary geometry, η is typically more conservative than the value given by Eq. 37.[5] We return to this topic in the examples provided in Section 7.

An alternate approach involves the use of the adjoint correction term, $|(\psi_h^H)^{\mathrm{T}} \mathbf{R}_h(\mathbf{Q}_h^H)|$, as an error indicator [64, 18]. However, the remaining error term converges at approximately double the rate of the correction and is more conservative at sonic and stagnation lines, where adjoint variables vanish

[5] Since η estimates the bound on the remaining error with respect to the embedded mesh, $C\eta$ can be used to extrapolate toward the exact value, similar to Eq. 37.

but their derivatives do not. Moreover, since the flow and adjoint equations are solved on the same mesh, there is an open question regarding whether the error indicator maintains consistency of the adjoint solution as the mesh is refined. Although $|\eta|_H$ is sensitive to nonlinearities in the adjoint solution, other implementations [15, 5] use a complementary remaining-error term that involves an inner product of the flow solution with the adjoint residual. This term makes the mesh more suitable for the adjoint solution, but our numerical experiments did not show significant benefits.

There are several extensions of the present approach for handling multiple outputs. In principle, each output requires its own adjoint, which significantly increases simulation cost. One way to reduce the cost is to form a discrete error equation that is solved in conjunction with a modified adjoint system [16]. In this work, a simpler approach is used, where the solution of only one adjoint system is required. Multiple outputs are combined using a weighted-sum formula:

$$J = \sum_{i=1}^{K} w_i J_i \tag{39}$$

where K is the number of outputs and w is an array of user-specified constants. In practice, outputs are frequently projections of wall pressure, for example aerodynamic forces and moments, for which the weighted-sum of axial, normal and side forces works well.

5. Mesh Adaptation

5.1. Equidistribution of Remaining Error

While error estimates are critical for assessing solution quality, automatic error control requires additional procedures that identify regions of high error and modify the mesh to drive the error below desired tolerances. Unlike traditional error indicators, such as feature detection and estimates of local truncation error, $|\eta|_H$ is a direct (point-wise) estimate so there is no ambiguity regarding the selection of an indicator variable, its relation to the functional error, and its convergence rate.

To introduce the concept of error equidistribution, consider an example error map shown in Figure 5. The values of $|\eta|_H$ for C_d in the near-body region of a Joukowski airfoil in subsonic flow ($M_\infty = 0.4$ and $\alpha = 1^\circ$) are shown in the left frame. A logarithmic scale is used to emphasize the rapid variation of the error indicator near the airfoil, with highest errors at the leading and trailing edges. The corresponding error histogram for the entire domain is shown on the right. The horizontal axis contains bins of the error-indicator values. High-error cells lie to the right. The vertical axis is the percentage of cells in each bin.

The histogram provides insight into how well the mesh fits the simulation. In this case, the histogram is skewed, with most of the cells contributing little error. This indicates that the mesh is inefficient, which is expected because the near-body mesh is uniform. The high-error cells are close to the airfoil and, in particular, near regions of high curvature. The highest errors (above 10^{-5}) dominate η. This is reflected by the location of the mean-error value, η/N, shown as a dashed line in Figure 5. The majority of cells have error of several orders below the mean, thus the mean is well to the right of the peak (the mode of the histogram). Hence, most of the computational work associated with this mesh is unnecessary.

The basic strategy for controlling cell size in order to minimize error is to equidistribute the error as the adaptation advances. The principle of error equidistribution has been demonstrated in [1, 3],

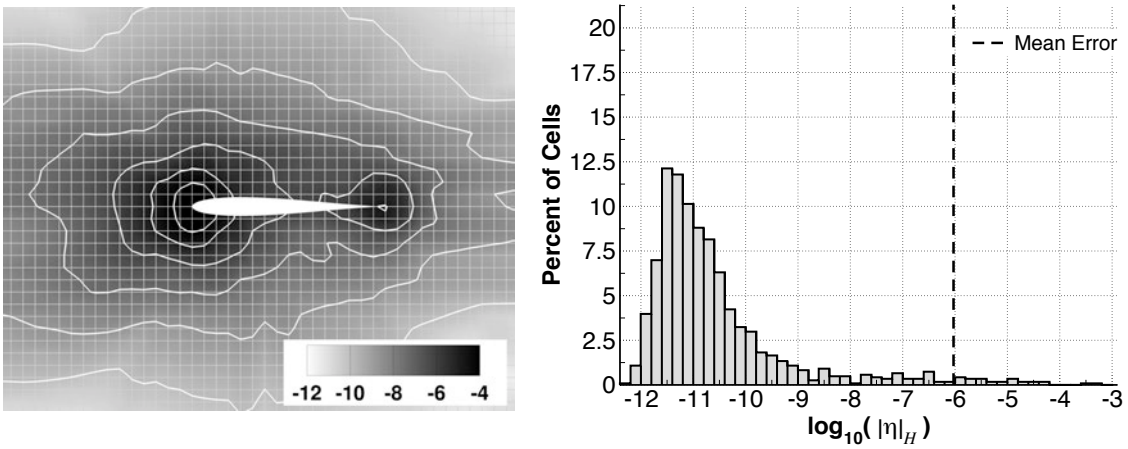


Figure 5. Near-body cell-wise error indicator for C_d ($\log_{10}|\eta|_H$) of airfoil in subsonic flow (left) and corresponding error histogram (right).

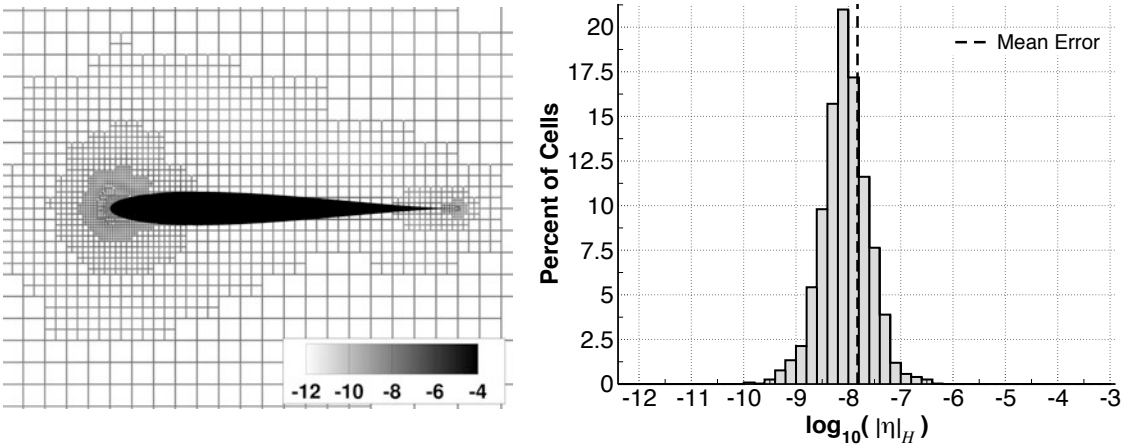


Figure 6. Near-body adapted mesh colored by $\log_{10}|\eta|_H$ in C_d for airfoil in subsonic flow (left) and corresponding error histogram (right).

among others. The goal is for each cell to contribute equally toward improving the accuracy of the simulation. Intuitively, any departure from a uniform error distribution implies that the mesh points could be redistributed to obtain a lower average error. Historically, error equidistribution has been sought in global error estimates, such as truncation errors or energy norms. These error estimates, however, tend to significantly increase computational cost because they may trigger refinement of all flow features everywhere in the domain. In contrast, adjoint error estimates seek equidistribution of the functional error and consequently focus only on regions (cells) important for predicting the functional. In other words, the adaptation seeks to refine cells that make $|\eta|_H$ uniform and η small.

Figure 6 illustrates the main idea, where an adapted mesh is generated for the airfoil example of Figure 5. The adapted mesh is shown on the left and its histogram on the right. The histogram is nearly symmetric, with little spread. The value of the error indicator varies less than an order of magnitude in over 80% of the cells. Compared to the error map shown in Figure 5, the adapted mesh reduces the error variation by seven orders of magnitude. All high-error cells have migrated to the left and are clustered close to the mode of the histogram. Moreover, the mean is also close to the

mode, suggesting that little computational work is wasted on low-error cells. The grading of the mesh is achieved through a series of adaptations and is a direct reflection of the error map in each cycle, such as the one shown in Figure 5. In the limit, the ideal histogram is a delta function. In practice, the discrete subdivision of Cartesian cells and cell-wise nonuniform error convergence, as well as overall computational cost, limit the tightness of the histogram.

5.2. Error Control

Efficiency of the adaptation procedure is driven by the amount of computational work needed to transform the error histogram of the initial mesh to a delta function and position it sufficiently far left in the region of low error. This can be controlled by carefully selecting refinement and coarsening thresholds above and below which cells are marked for refinement and coarsening, respectively. Choosing refinement thresholds where only the highest-error cells are refined yields tight histograms but requires many adaptation cycles to shift the mode into a region of low error. Therefore, the procedure should also minimize the number of cycles and, in particular, avoid solving on similar meshes when close to the final mesh.

For simulation verification, perhaps the most common approach is to prescribe a tolerance TOL on the remaining error or directly on $\mathscr{E}$. For example, the goal may be to construct a mesh with $\eta < 0.0001$ in C_D. This results in thresholds proportional to $\mathtt{TOL}/N$ (or $\mathtt{TOL}/N_{max}$) that drive the mesh toward error equidistribution. In earlier work [35, 36], we evaluated this approach and modified it to accommodate a"worst-errors-first" strategy. This reduces computational cost by avoiding the problem of generating too many cells early in the adaptive process, before the error map is accurate, and then paying for these cells on every intermediate mesh until the highest-error cells are finally addressed in the closing cycles. The threshold is set to $\lambda \cdot \mathtt{TOL}/N$, where $\lambda \geq 1$ is a user-specified array of constants that typically decrease as the adaptation advances.

Use of the tolerance-driven approach in practice reveals several problems. Specifying a meaningful TOL is awkward for functionals such as line sensors, where there is little intuition guiding reasonable choices of desirable error level. In addition, λ is problem dependent and the resulting mesh growth is hard to control in difficult simulations, where occasional poor convergence of the flow or adjoint solver may occur and cause spurious error estimates in some region of the domain. An alternate approach is to minimize the error for a given cell budget, e.g., constructing a mesh with one million cells that predicts C_D most accurately. Cells are sorted on their level of error and a threshold is determined such that a fraction of the highest error cells is refined. The threshold in each cycle can be determined from statistics of the error distribution [65, 66, 37], such as the mean and standard deviation, or set to meet a user-specified mesh growth.

5.3. Implementation

The main steps of the simulation are given by Procedure 1. While the steps are standard, we note several places where we specialized the procedure:

1. The surface triangulation T is held fixed. Consequently, it is important that the surface triangulation is sufficiently fine to support the final volume mesh, particularly in regions of high surface curvature.

Procedure 1: Adaptive Mesh Refinement

Input: Surface triangulation T and refinement parameter array τ

Termination Criteria: `TOL`, or maximum number of cells N_{max}, or number of cycles M, or maximum level of refinement R_{max}

$H_0 \longleftarrow InitialMesh(T)$ `// Generate initial mesh`

for $i \leftarrow 1$ **to** M **do**

  $\mathbf{Q}_H \longleftarrow FlowSolve(H_{i-1})$

  $\psi_H \longleftarrow AdjointSolve(H_{i-1}, \mathbf{Q}_H)$

  $h \longleftarrow EmbedMesh(H_{i-1})$ `// Uniform refinement`

  $\eta_H \longleftarrow CellwiseErrorEstimate(h, H_{i-1}, \mathbf{Q}_H, \psi_H)$

  $\eta \longleftarrow \sum_{H_{i-1}} |\eta|_H$

  $H_i \longleftarrow AdaptMesh(\tau_i, |\eta|_H, H_{i-1}, \mathtt{TOL})$

  break if $(\eta < \mathtt{TOL} \;||\; N > N_{max} || \; R > R_{max})$

end

$\mathbf{Q}_H \longleftarrow FlowSolve(H_M)$ `// Optional solve`

Result: $\{J_H, J_c, \eta\}_i, \;\; i = 0, \ldots, M$

2. The exit criteria of the adaptation loop are positioned such that only the flow solution is computed on the final mesh. This is optional but effective in practice, because the primary outputs of the simulation come from the flow solver and it is sensible to use the already computed error map. In terms of simulation verification, the procedure is conservative because we expect η to decrease on the final mesh.

3. Except for the final adaptation, only one level of refinement is added per adaptation cycle. We allow more levels on the final cycle because the final error map should be close to asymptotic. Coarsening is not considered, which improves robustness without significantly impacting the efficiency of steady simulations.

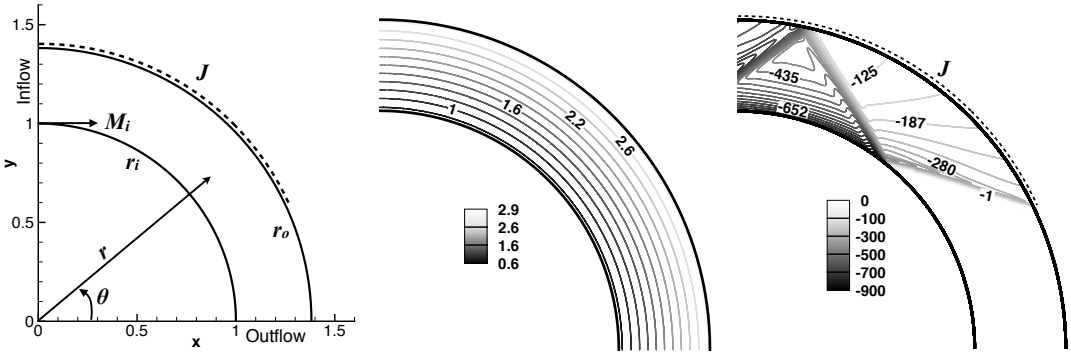
4. Once the cells are marked for refinement, the tags are processed to enhance mesh smoothness. This includes filtering out refinement islands and voids, buffering tags (usually by one layer of cells) and enforcing $2:1$ cell ratios.

The main results of Procedure 1 are the functional value, its correction, and remaining error for the sequence of meshes generated during the simulation. Convergence analysis of these quantities is used to verify the simulation.

6. Code Verification

An analytic supersonic vortex provides a model problem with a known solution [67] to verify the performance of the error estimation framework. We also verify the accuracy of the adaptive mesh refinement procedure to establish a benchmark for automatic simulation verification. The problem involves isentropic flow between concentric circular arcs at supersonic conditions, as shown in Figure 7(a). The exact solution is given by:

$$\rho = \rho_i \left\{ 1 + \frac{\gamma - 1}{2} M_i^2 \left[1 - \left(\frac{r_i}{r} \right)^2 \right] \right\}^{\frac{1}{\gamma - 1}}$$



(a) Setup (b) Pressure (c) Density adjoint

Figure 7. Supersonic-vortex model problem ($M_i = 2.25$, $r_i = 1$, and $r_o = 1.382$).

$$u = a_i M_i \left(\frac{r_i}{r}\right) \sin\theta, \quad v = -a_i M_i \left(\frac{r_i}{r}\right) \cos\theta, \quad p = \frac{p_i}{\rho_i^\gamma} \rho^\gamma \tag{40}$$

where $a_i = \rho_i = 1$, $p_i = 1/\gamma$, $M_i = 2.25$, $r_i = 1$, and $r_o = 1.382$. The output of interest is the integral of pressure over a portion of the outer arc:

$$J = \int_0^{\frac{5\pi r_o}{14}} p_{r_o} \, \mathrm{d}l = \frac{5\pi r_o}{14} p_{r_o} \approx 4.39262683 \tag{41}$$

as sketched in Figure 7(a). This choice permits validation of the quadratic adjoint interpolant described in Section 2, which models the adjoint solution on the embedded mesh. Figure 7 shows that while the flow is smooth, the adjoint is not.

All simulations are initialized with the analytic solution and solved to steady state without limiter. Dirichlet boundary conditions at the inlet are prescribed from the analytic solution; the wall boundary conditions are specified via Eq. 19.

6.1. Uniform Mesh Refinement

A uniform mesh refinement study is performed that involves a sequence of six nested meshes. Figure 8 shows the initial mesh, which contains 47 cells. The final mesh contains 33,069 cells. Note that the boundary discretization changes non-smoothly with refinement, e.g., on each mesh the cut-cell area fractions varied by at least five orders of magnitude. Consequently, convergence rates are obtained via linear regression of the finest four solutions.

Figure 8 shows that the solution error (point-wise L_1 norm of $\rho - \rho_H$) is $\mathscr{O}(h^2)$ when measured over the entire domain. This is consistent with the second-order accurate discretization and the smooth analytic solution. The second line in Figure 8 shows that the convergence rate is reduced to slightly below $\mathscr{O}(h^{3/2})$ when confined to include only the boundary due to the irregular discretization stencils in the cut cells.

Despite the slower convergence of the solution along the boundary, the first line in Figure 9(a) shows that the error in J_H is $\mathscr{O}(h^2)$. The second line shows the convergence rate of error in the corrected functional J_c. As expected, its convergence parallels J_H, but the error improves by about a half-order of magnitude. The third line in Figure 9(a), labeled "J_c Exact ψ_h", is obtained by solving the adjoint equation on the embedded mesh and using these values in Eq. 35 when computing the corrected

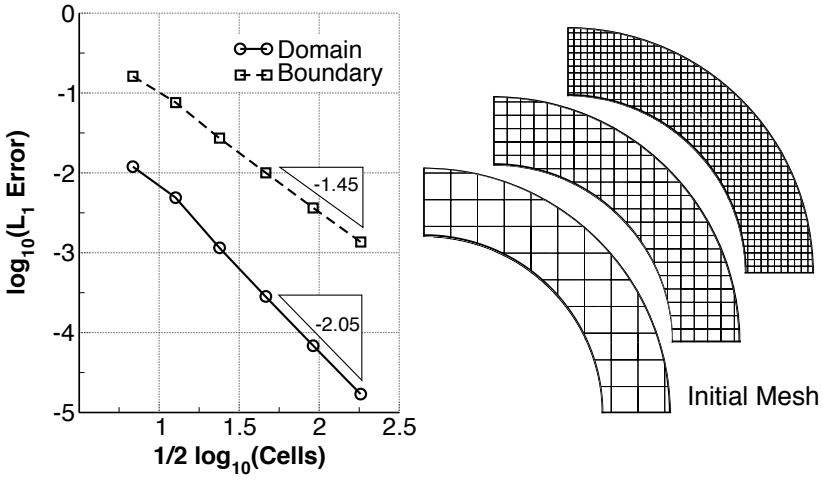
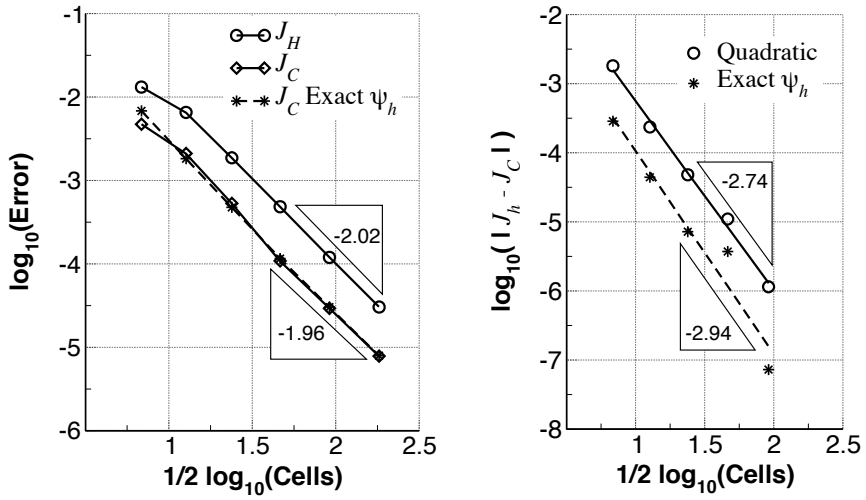


Figure 8. Error in density versus number of cells (left), and initial mesh and meshes after first two uniform refinements (right).



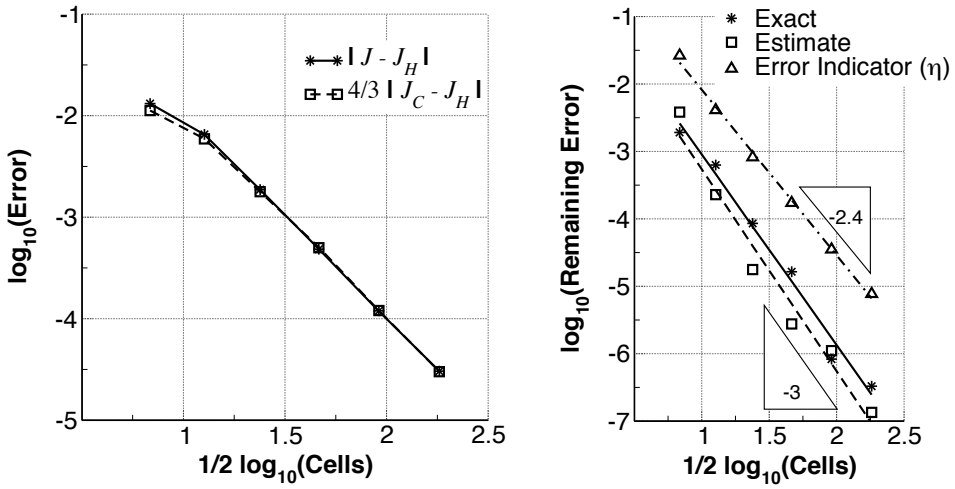
(a) Error (b) Relative error

Figure 9. Convergence of functional (J_H) and its correction (J_c) for the uniform mesh refinement study of the supersonic vortex. Error is measured with respect to the exact value (left) and the value on the embedded mesh (right).

functional. This is referred to as an exact correction, which we use to validate the triquadratic interpolant[6]. Except on the initial mesh, the quadratic correction performs as well as the exact correction.

Figure 9(b) examines the accuracy of the corrected functional J_c in detail. The correction should capture the dominant part of the functional's value on the embedded mesh J_h relative to the working mesh value J_H, as sketched in Figure 1 and expressed by Eq. 4. We examine the accuracy of the corrected functional using both the quadratic adjoint interpolant and the embedded-mesh adjoint, labeled as "Exact ψ_h" in Figure 9(b). Both corrections are superconvergent; the exact correction attains almost $\mathcal{O}(h^3)$ when measured in this relative-error metric. The quadratic mimics the exact correction at a slightly slower rate and with about a half-order offset starting from the coarsest mesh. These results show that the adjoint field on the embedded mesh is accurately predicted with the limited triquadratic interpolant (described in Section 4).

The ability to compute relative error accurately suggests that the true error should also be predicted reliably as H tends to zero. Since the convergence rate of the functional is second-order, Eq. 37 with $C = \frac{4}{3}$ estimates the true error. The first line in Figure 10(a) shows convergence of the true error $|J - J_H|$, which is copied from Figure 9(a) for reference. The second line (squares) is the error estimate $\mathscr{E}$ using Eq. 37. The estimate is sharp; the ratio of the estimate to the true error (effectivity) is close to 1 on all but the coarsest meshes, i.e., the lines essentially over-plot.



(a) Error estimate

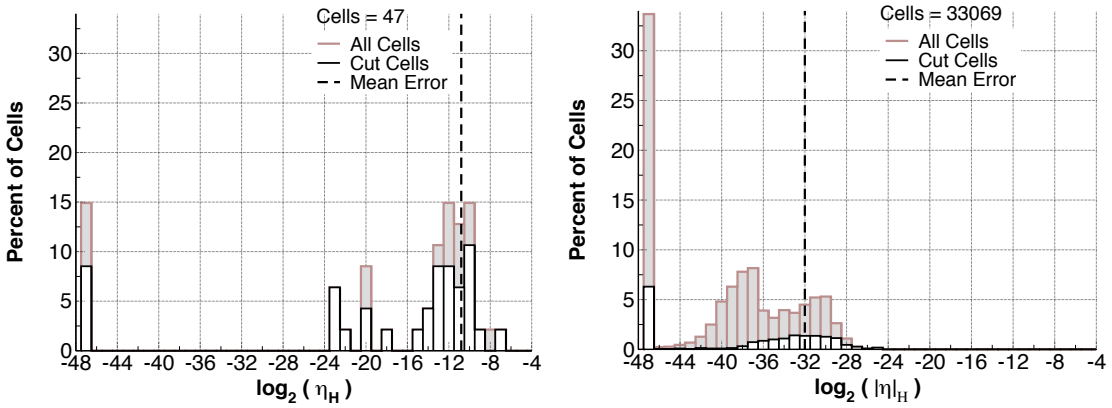
(b) Remaining error

Figure 10. Accuracy of the error estimate using Eq. 11 (left). Remaining error and bound estimate (right) for the uniform mesh refinement study of the supersonic-vortex problem.

The vanishing gap between the estimated and true error is quantified by the first line in Figure 10(b), labeled "Exact". We use this error gap to test the accuracy of the remaining error term of Eq. 32. This is done by evaluating Eqs. 36 and 38 without localization, i.e., by omitting the absolute value operator in Eq. 38. The second line in Figure 10(b), labeled "Estimate", shows the value of $4/3\eta$ without localization[7]. The estimate is within a half-order of magnitude of the measured gap and has a similar convergence rate. The agreement is excellent considering that the quadratic interpolant

[6]Recall that the cost of an embedded-mesh adjoint solution is prohibitive in practice. It is roughly equal to that of a flow solution on the embedded mesh.

[7]The factor of $4/3$ is applied to extrapolate the remaining error to its analytic value based on the observed second-order functional convergence.



(a) Initial mesh

(b) Final mesh (cut cells magnified 6× for clarity)

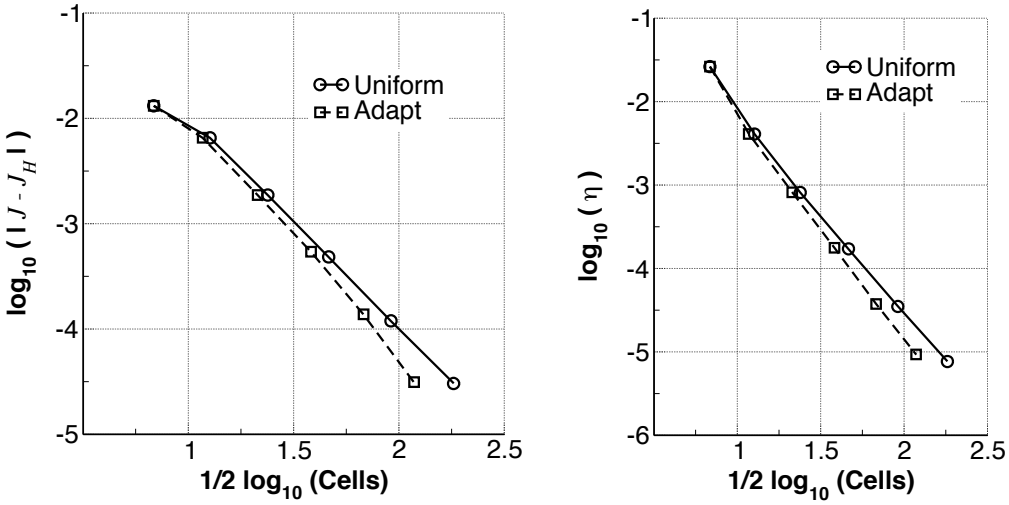
Figure 11. Histograms of the cell-wise error indicator $|\eta|_H$ for the uniform mesh refinement study of the supersonic vortex. Heights are normalized by the total number of cells. Error indicator values in cut cells are shown as rectangles with white fill.

is used in both the correction and remaining error terms, and only linear prolongation of the flow solution is used when evaluating the embedded-mesh residuals. The third line in Figure 10(b), labeled "Error Indicator", shows the effect of localization when forming the error-indicator bound η using Eq. 38. Recall that the absolute value operator prevents cell-wise error cancellation. As a result, the bound is roughly one order of magnitude larger than the remaining error on the coarsest mesh and its convergence rate is slower, indicating that the bound is a conservative estimate of the remaining error.

Having established the accuracy of the error indicator, Figure 11 shows histograms of the error indicator $|\eta|_H$ on the initial and final meshes. The histograms are similar to the ones presented in figures 5 and 6 except we use a $\log_2$ scaling of the cell-wise error. This scaling is intuitive for predicting how far a bin moves to the left once its cells are refined because the subdivision of Cartesian cells is discrete. For example, we show in Figure 10(b) that the convergence rate of the error-indicator bound η is roughly $\mathcal{O}(h^2)$, which implies that the mean error should shift four units to the left after each refinement[8]. Figure 11 shows that the mean error is close to 2^{-12} on the initial mesh and shifts to 2^{-32} after five refinements.

Figure 11(b) shows that the uniformly refined mesh is not particularly well suited for the simulation at hand. The error distribution is far from a delta function and more than a third of the 33,069 cells contribute no error. The adjoint field shown in Figure 7(c) provides some insight. The large variations reveal the influence of point-source mass perturbations, which include interactions with the inner arc, on J (Eq. 41). Recall that J is defined over about $2/3$ of the outer arc, at which point the adjoint variable vanishes because any perturbation downstream of this location cannot influence J in this hyperbolic problem. Hence, cell refinement outside the functional's zone of dependence yields many cells with zero-error contributions. These cells increase cost without improving accuracy of the simulation.

[8]The mean shifts $p+d$ units to the left, where p is the order of the error indicator and d is the spatial dimension.



(a) Functional error (b) Error-indicator bound

Figure 12. Convergence of the functional error (left) and error indicator η (right) on adapted meshes.

6.2. Adaptive Mesh Refinement

Accuracy of the adaptive procedure is demonstrated by performing five adaptive refinements starting from the initial mesh of the uniform study. The refinement threshold is set to the mean error value, except in the first two adaptations where it is shifted two $\log_2$ bins to the left of the mean due to the coarse initial mesh. Figure 12 shows the error convergence of the functional and η, and compares them with those of the uniform meshes. We observe that after each refinement, the functional error and η nearly match the values obtained from the uniform meshes but use fewer cells. The final mesh contains 13,929 cells compared to the 33,069 cells of the finest uniformly refined mesh.

Figure 13 shows the final mesh and its error histogram. The refinement pattern is clearly driven by features of the adjoint solution; recall Figure 7(c). Essentially no mesh refinement occurs past the functional's zone of dependence. An error indicator based on local truncation errors would unnecessarily adapt cells in this region. The largest sensitivity to residual errors occurs along the inner arc, which may seem counter-intuitive. This is due to the nonuniform inlet Mach number and the number of local Mach wave reflections that can reach the functional from the inner arc. More importantly, the final mesh appears to strike a good compromise for accommodating both the flow and adjoint solutions. The inner product within the error indicator $|\eta|_H$ is dominated by the adjoint interpolation error $(\psi_{\mathrm{TQ}} - \psi_{\mathrm{TL}})$, which emphasizes regions of high adjoint curvature that are captured in the final refinement pattern.

The histogram in Figure 13 shows that the final adapted mesh fits the simulation well and contrasts markedly from that in Figure 11(b). The histogram is symmetric with little spread, the mean error is close to the mode, and the adapted mesh has almost no cells with zero error. Moreover, the histograms of figures 11 and 13 show not only the error indicator values in every cell, but also isolate the errors in the cut cells. Recall that Figure 8 shows slower convergence rate of the solution in the cut cells. To compensate, the error indicators in cut cells are inflated by a factor of 1.5 before selecting cells for refinement. This value is based on experience with many problems. Figure 13 shows that mode of the error distribution in cut cells roughly corresponds to the mode for all cells, indicating that the adaptation mechanics are not adversely affected by the nonuniform error convergence.

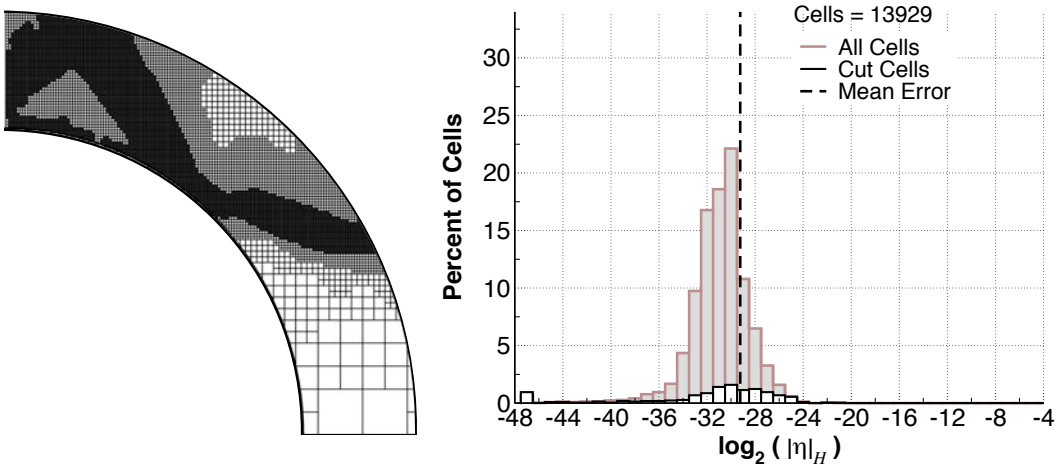


Figure 13. The final adapted mesh (left) and the corresponding error-indicator histogram (right) for the adaptive refinement of the supersonic vortex problem. The cut-cell percentage is inflated $3\times$ for clarity.

7. Examples of Simulation Verification

Three examples are presented to demonstrate the effectiveness of Procedure 1 in simulation verification. The examples progressively increase in complexity, from airfoil simulations to three-dimensional simulations of a launch abort vehicle. To varying degrees, the examples deliberately violate the assumptions made in the derivation of the error estimates — in particular, assumptions of smoothness, steady-state, and fineness of the initial mesh. These assumptions rarely apply in practical engineering simulations and our goal is to characterize the performance of Procedure 1 in such situations.

7.1. Airfoil Performance Database

The first example demonstrates verification of a model aerodynamic performance database. The goal is to predict the drag coefficient, $J = C_d$, of the familiar NACA 0012 airfoil over a range of freestream conditions. We consider a total of 60 cases involving six subsonic Mach numbers, $M_\infty = \{0.1, 0.2, 0.3, 0.5, 0.7, 0.9\}$ and four supersonic Mach numbers, $M_\infty = \{1.1, 1.3, 2, 3\}$, at six angles of attack, $\alpha = \{0, 0.5, 1, 2, 4, 8^\circ\}$.

Figure 14 shows the near-body region of the initial mesh, including a closeup of the NACA 0012 airfoil[9] with a unit chord. The distance to the outer boundary is 64 chords. The initial mesh contains 16×16 cells with characteristic length $H = 8$ chords. The airfoil is intersected by just

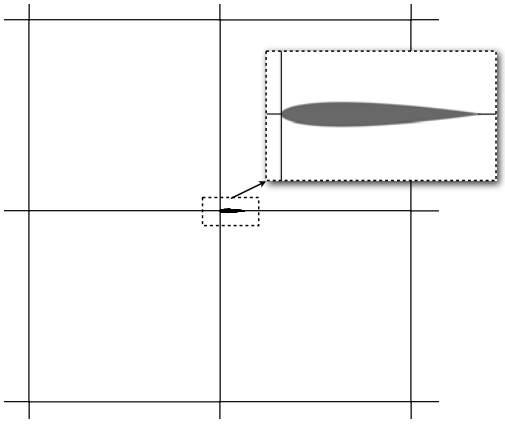


Figure 14. Near-body view of initial mesh with inset showing NACA 0012 airfoil.

[9] The trailing edge of the airfoil is modified to be sharp, which is accomplished by modifying the last coefficient of the equation defining its thickness distribution [68].

four cells. The spatial discretization does not involve a limiter when $M_\infty \leq 0.5$; cases with $M_\infty > 0.5$ use the Van Leer limiter.

To construct the database, we set the maximum level of refinement, R_{max}, to 18 and set the adaptation threshold to the mean error. This is a compromise of the various database strategies we examined in [39], which contrasted uniform-error and fixed-mesh databases. In general, specifying a uniform error tolerance, e.g., $\mathscr{E} < 0.0001$ in C_d, for all cases is not cost effective. This is because the cost of a constant-error database is frequently dominated by a few corner cases, for which a less accurate computation would suffice. Alternatively, specifying the number of adaptation cycles (M in Procedure 1) and the mesh growth for each cycle fixes the cost of the database. This strategy, however, restricts the degree of control over the variation of error across the database. The current approach achieves a balance by requiring that the final meshes for all cases contain the same smallest cell size ($H \approx 0.0002$ chords), but allows the total number of cells to vary.

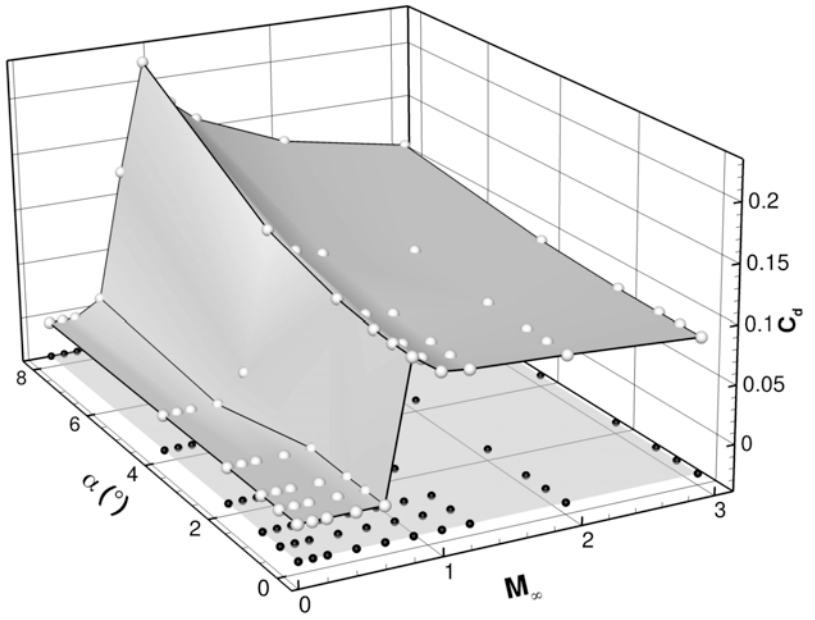
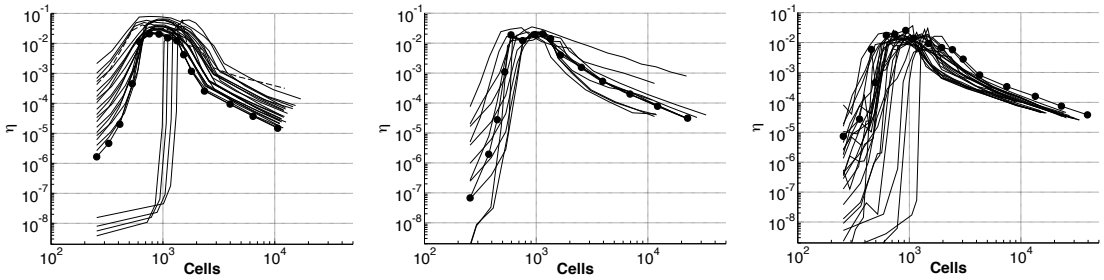


Figure 15. C_d for all freestream conditions of the airfoil performance database example.

Figure 15 shows the final value of C_d for all simulations in the database. Although there are no analytic solutions for the NACA 0012 airfoil in a finite computational domain, we expect C_d to approach zero in a shock-free two-dimensional inviscid flow. Reassuringly, Figure 15 shows that C_d is essentially zero when $M_\infty \leq 0.5$. Moreover, while the variation in C_d with angle of attack is generally mild at fixed M_∞, there is a rapid rise in C_d with respect to M_∞ in the transonic regime. The largest values of C_d are observed when $M_\infty = 0.9$, which is due to a strong expansion over the aft portion of the airfoil caused by shocks at the trailing edge. As the Mach number increases into the supersonic regime, there is a gradual decrease in C_d as the bow shock becomes more oblique.

While Figure 15 demonstrates that the simulation data follows expected trends, it offers little quantitative evidence for simulation verification. Since this example involves 60 simulations, we focus the current discussion only on the error estimate, Eq. 37, and the error indicator, Eq. 38, which are central to verification and adaptation. The single-point examples in subsections 7.2 and 7.3 study these quantities in greater detail.



(a) Subsonic cases (b) Transonic cases (c) Supersonic cases

Figure 16. Convergence of error indicator η for subsonic $M_\infty = \{0.1, 0.2, 0.3, 0.5\}$, transonic $M_\infty = \{0.7, 0.9\}$ and supersonic $M_\infty = \{1.1, 1.3, 2, 3\}$ cases of the database example.

7.1.1. Error-Indicator Convergence

Figure 16 shows convergence of the error-indicator bound η (Eq. 38). The cases are divided according to the freestream Mach number into groups of subsonic, transonic and supersonic flow. Overall, η converges well. This implies mesh convergence for all cases and demonstrates the ability of the adaptation to control the magnitude of the remaining error in Eq. 32. The circle symbols, plotted for an arbitrary case in each plot, mark adaptation cycles. The error indicator initially increases until the meshes reach around 1000 cells. This is typical when starting from such a coarse mesh; recall Figure 14. Increasing η indicates that new features of the solution are emerging, which are not yet captured by the error analysis. Nevertheless, the early error maps are sufficiently accurate to identify critical regions of the evolving flowfield. When combined with the incremental *h*-refinement strategy of Procedure 1, the error maps reliably handle coarse initial meshes. As the simulations approach R_{max}, in particular over the last four meshes of each case, η is decreasing linearly indicating that the output is asymptotic.

Figure 16(a) shows that η is reduced by three orders of magnitude for all but two of the 24 subsonic cases. The convergence pattern is very similar among the cases, especially once the meshes reach 1000 cells. All four cases with the sharp error rise at 1000 cells are at $\alpha = 0^\circ$. The case with the slowest convergence is when $M_\infty = 0.5$ and $\alpha = 8^\circ$, indicated by the dashed line, which corresponds to the onset of transonic flow. To reach the desired R_{max} of 18, the meshes for all subsonic cases require 14 adaptations[10].

Figure 16(b) shows convergence of η for the 12 transonic cases. The meshes, on average, are larger than those for the subsonic cases. The cases with the smallest values of η and also the smallest number of cells correspond to the shock-free cases, i.e., $M_\infty = 0.7$ and $\alpha \leq 1^\circ$. The slowest convergence (largest η) occurs when $M_\infty = 0.7$ and $\alpha = 8^\circ$; the next slowest is $M_\infty = 0.7$ and $\alpha = 4^\circ$. These two cases are actually the slowest to converge in the entire database. We obtained similar results in [39], where the largest sensitivity of aerodynamic performance to discretization error was also near $M_\infty = 0.7$. The primary culprit is the dependence of C_d on the location of the upper surface shock. Once M_∞ reaches 0.9, the shocks migrate to the trailing edge. Figure 16(c) shows convergence of the 24 supersonic cases. These cases converge well and are tightly clustered. The meshes require 15 adaptations to reach R_{max} of 18, which is one additional cycle when compared to the subsonic and transonic cases.

[10]The initial mesh contains four levels of refinement.

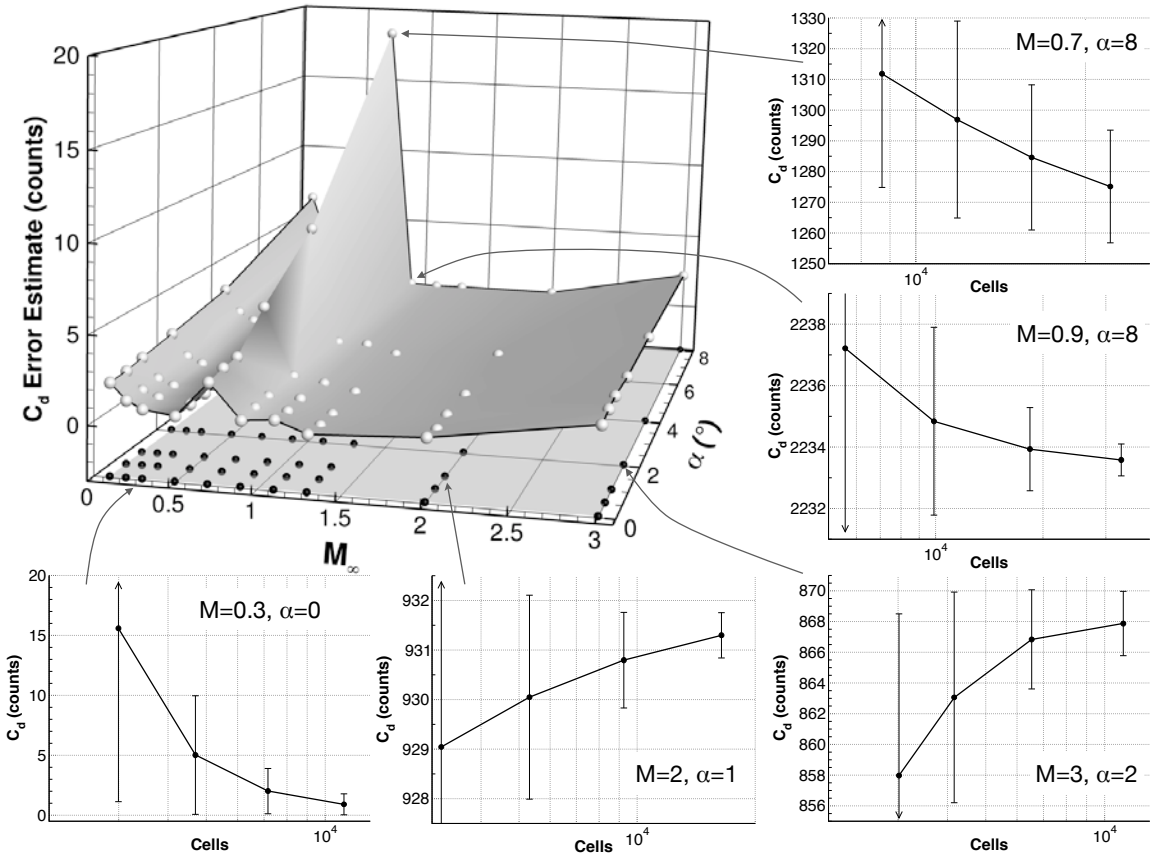


Figure 17. Final error estimate in C_d for all freestream conditions (upper left) and samples of C_d convergence on the last four meshes.

7.1.2. Level of Discretization Error

Having established convergence of the error indicator, we now examine the level of discretization error across the database. Figure 17 shows the value of the final error estimate $\mathscr{E}$ in the carpet plot at upper-left. The error estimates are computed with $C = \frac{4}{3}$ in Eq. 37 for shock-free cases that do not require a limiter ($M_\infty \leq 0.5$); otherwise, $C = 2$ is used. The carpet plot shows that $\mathscr{E}$ is small and essentially uniform ($\mathscr{E} \leq 2$ counts) over most of the subsonic and supersonic cases. Not surprisingly, the largest error estimates are obtained in the transonic regime, specifically when $M_\infty = 0.7$, which corroborates with the slow convergence of η in Figure 16(b). There is also an increase in $\mathscr{E}$ as $M_\infty \to 0$ that indicates some loss of accuracy in the incompressible regime due to the lack of low Mach-number preconditioning. This is consistent with results presented in [39].

Along the perimeter of the carpet plot in Figure 17, we show convergence of C_d for the last three adaptations of representative cases. The error bars denote the value of $\mathscr{E}$. In general, the insets show that C_d changes less than one count over the final adaptation, and $\mathscr{E}$ is decreasing by about a factor of two per cycle and brackets the final solution over at least the last two cycles. For example, the lower-left inset shows the classic subsonic case $M_\infty = 0.3$ and $\alpha = 0^\circ$, where C_d correctly approaches zero as $H \to 0$. The error estimate brackets the expected final solution ($C_d = 0$ to plotting accuracy) on the last three meshes. The case at $M_\infty = 0.7$ and $\alpha = 8^\circ$, shown at top-right, is the main exception. The value of C_d is still changing significantly and the error estimate is just starting to tighten. More adaptation cycles (smaller cells) would be required to reduce the error estimate further. Taken

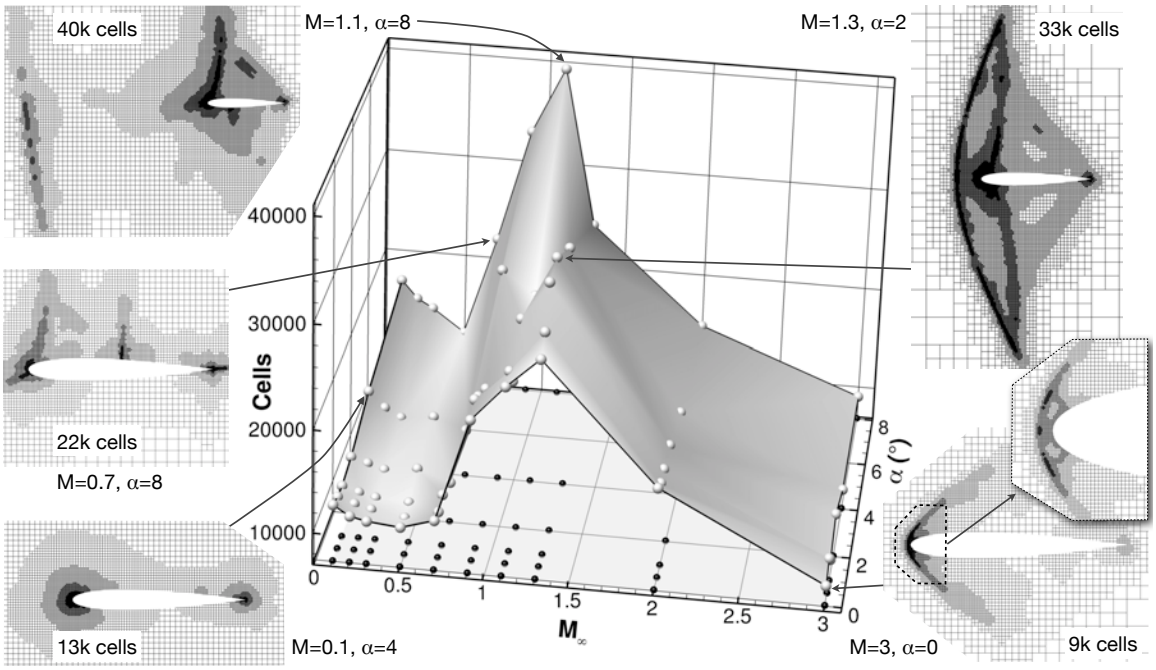


Figure 18. Number of cells for all freestream conditions with samples of final adapted meshes (near-body views).

together, figures 16 and 17 provide the quantitative evidence required for verification of every case in the database.

7.1.3. Meshing Requirements

Figure 18 shows the number of cells required to achieve the error levels presented in Figure 17, as well as snapshots of the final near-body meshes for selected cases. The carpet plot shows that the number of cells varies by about a factor of four over the database and peaks when $0.9 \leq M_\infty \leq 1.3$. To give an indication of the wall-clock time needed to construct this database, each case requires, on average, about four minutes on one core of a laptop computer[11].

The largest mesh is obtained when $M_\infty = 1.1$ and $\alpha = 8^\circ$ and contains 39,500 cells. The top-left inset shows that the high cell count is primarily due to a detached bow shock far upstream from the airfoil, which is indicated by the cluster of small cells along the inset's left edge. In contrast, Figure 18 shows that cases with $M_\infty \geq 2$ require many fewer cells. Recall that the refinement pattern is driven by the inner product of the adjoint interpolation error $(\psi_{\mathrm{TQ}} - \psi_{\mathrm{TL}})$ with the flow residual error, Eq. 36. Upstream of the bow shock, the flow residuals are zero because the spatial discretization preserves uniform freestream. Concurrently, the adjoint solution vanishes downstream of the limiting characteristic intersecting the airfoil's trailing edge. This is similar to Figure 7(c) of the supersonic vortex problem, where the adjoint solution is zero when outside the functional's zone of dependence. Consequently, $|\eta|_H$ vanishes upstream of the bow shock and outside the limiting characteristic, as indicated in the insets on the right side of Figure 18 for cases at $M_\infty = 1.3$ and $M_\infty = 3$. This confines the refinement pattern to a "diamond" that contracts in the cross-flow direction with increasing Mach

[11] MacBook Air (2013) with a 1.7 GHz Intel Core i7 processor and 8GB of memory.

number. As a result, the smallest mesh, containing only 8860 cells, is obtained at the highest Mach number, specifically at $M_\infty = 3$ and $\alpha = 4°$.

As $M_\infty \to 0$, the carpet plot of Figure 18 shows a moderate increase in the number of cells that is consistent with the increase in $\mathscr{E}$ shown in Figure 17. The inset at lower-left shows a typical mesh in the subsonic regime. At a given angle of attack, the meshes are self-similar in the sense that the refinement regions simply enlarge with decreasing Mach number to compensate for the weakening pressure variations.

Overall, figures 17 and 18 offer a convincing demonstration of the benefits of adaptive mesh refinement and automatic error control for simulation verification. Even in this simple 2D example, constructing a single mesh that accommodates all the flow regimes — from smooth subsonic flow to the disparate shock systems of transonic and supersonic flow — is a daunting task that requires orders of magnitude more cells. Moreover, uniform refinement of such a mesh to demonstrate verification is not practical.

7.2. Pressure Signature of a Body in Supersonic Flow

This example demonstrates verification of a simulation that predicts the near-field pressure signature of a lifting configuration in supersonic flow. Such simulations are used to determine loudness of sonic booms [69, 70]. The key to credible noise analysis is the accurate prediction of the weak shocks and expansions, and their interactions, that are generated by low-sonic-boom vehicles. These pressure disturbances are highly susceptible to attenuation by discretization error and consequently, these simulations hinge on the mesh [71, 38, 23, 72, 43, 73].

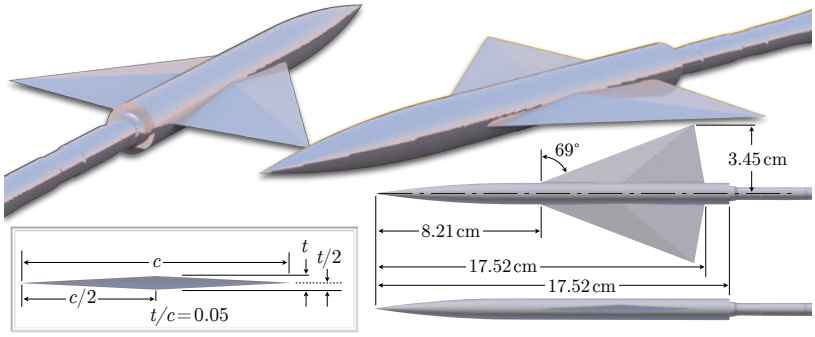


Figure 19. Delta-wing-body model with a partial view of the wind-tunnel sting.

The simulation involves a popular wing-body model, originally identified as "Model 4" in the experiments of Hunton *et al.* [74] and used subsequently in many studies; for example, [70, 38, 72, 75, 76]. Figure 19 shows the details of the geometry, including an axisymmetric fuselage, a delta wing with diamond-airfoil cross sections, and an approximate wind-tunnel sting from [38] that forms a step junction at the base and extends downstream. The surface tessellation contains 1.3 million triangles.

Our goal is to predict the pressure signature on a line located 3.6 model lengths below the model's centerline and parallel to freestream. The output functional is given by Eq. 21, where we set $n = 2$ to emphasize the importance of accurately capturing the peaks of the signature. The freestream Mach number is 1.68 and the angle of attack is set to match a C_L of 0.15. This corresponds to one of

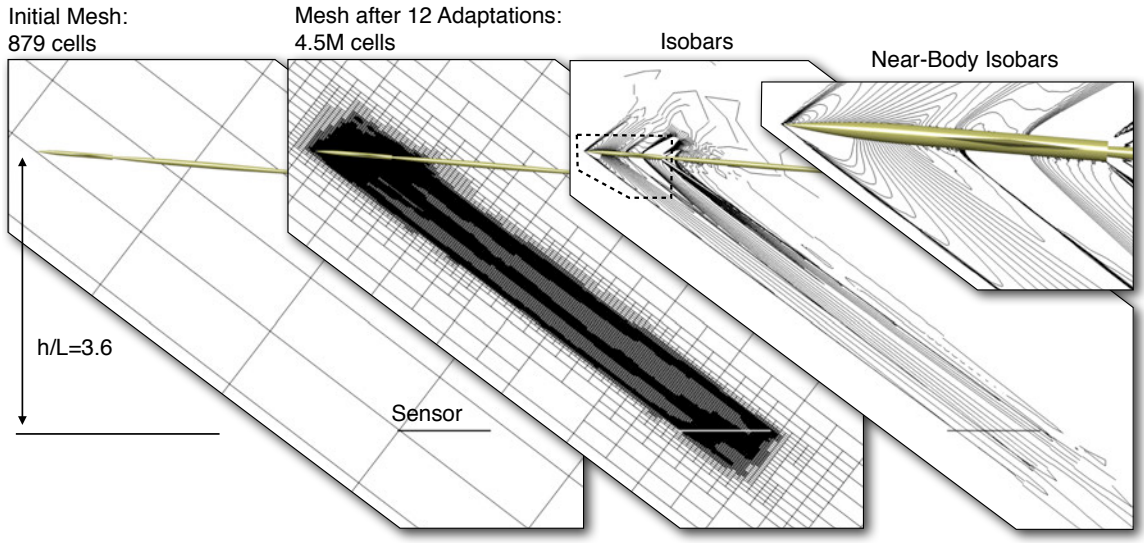


Figure 20. Near-field view of the initial mesh (left), mesh after 12 adaptations (middle), and isobars (right) with inset frame showing isobars close to the model. Pressure signature example (symmetry plane, isobar range 0.65–0.78 in 0.005 increments, $M_\infty = 1.68$).

the experiments performed in [74]. Monotonicity of the solution is maintained through use of the Barth-Jespersen limiter [63].

The left frame of Figure 20 shows the setup for a half-body simulation. The model and the sensor (solid horizontal line) are shown on the symmetry plane of the near-field region of the initial mesh. The mesh is rotated to approximately align the cells with the freestream Mach-wave angle and the cells are stretched, about $2:1:2$, in the direction of the wave angle and spanwise to improve computational efficiency. The initial mesh contains $12 \times 12 \times 8$ cells, and the geometry and sensor are intersected by just five cells. In other words, except for the rotation and cell stretching, the mesh is not biased to anticipate features of the solution and should be far from the asymptotic region of the functional. The refinement threshold for each adaptation cycle is set to the mean error value if the error indicator is decreasing, otherwise it is set to twice the mean value to limit over-refinement. The maximum number of cells, N_{max}, is set to 20 million cells.

The middle and right frames of Figure 20 show the mesh and flow solution after 12 adaptations. Before addressing verification, we briefly discuss the salient features of the refinement pattern. As expected, the refinement follows the shocks and expansions between the model and the sensor; however, note the refinement above the model and below the sensor. Similar to the vortex problem of Section 6, we visualize the functional's zone of dependence through the adjoint solution. Figure 21 shows the absolute value of the first adjoint variable. The grayscale map is tuned to indicate where point-source perturbations of the mass-conservation equation influence the pressure signature. In addition to the region between the body and the sensor, the plot confirms the dependence on regions above the model and below the sensor. This is consistent with the expected propagation of characteristics in this three-dimensional flowfield. Moreover, no refinement occurs upstream of the leading shock because the adjoint solution and the flow residual are zero[12], which makes $|\eta|_H$ vanish. There is also little refinement in the wake. This happens because the adjoint vanishes downstream of the sensor, since residual perturbations past the sensor in supersonic flow cannot affect the signature. Lastly, the sensitivity of the error indicator to changes in adjoint curvature triggers

[12]The adjoint is zero due to the quadratic form of J.

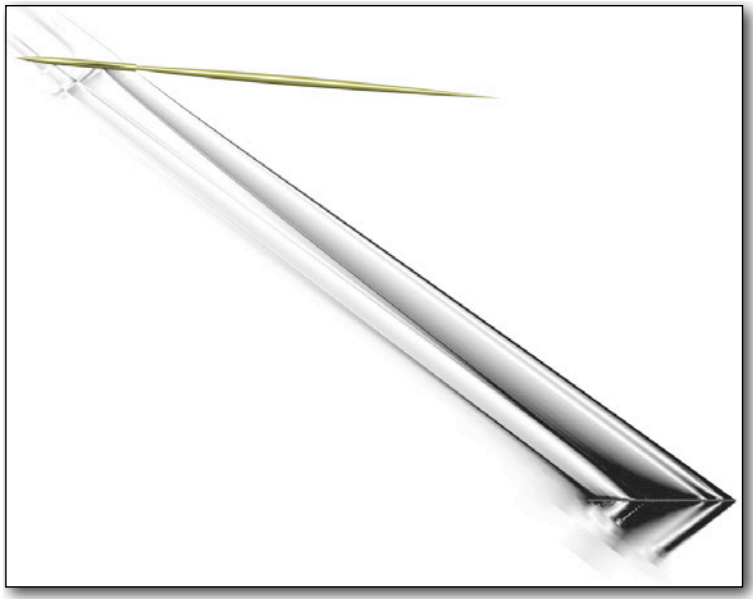


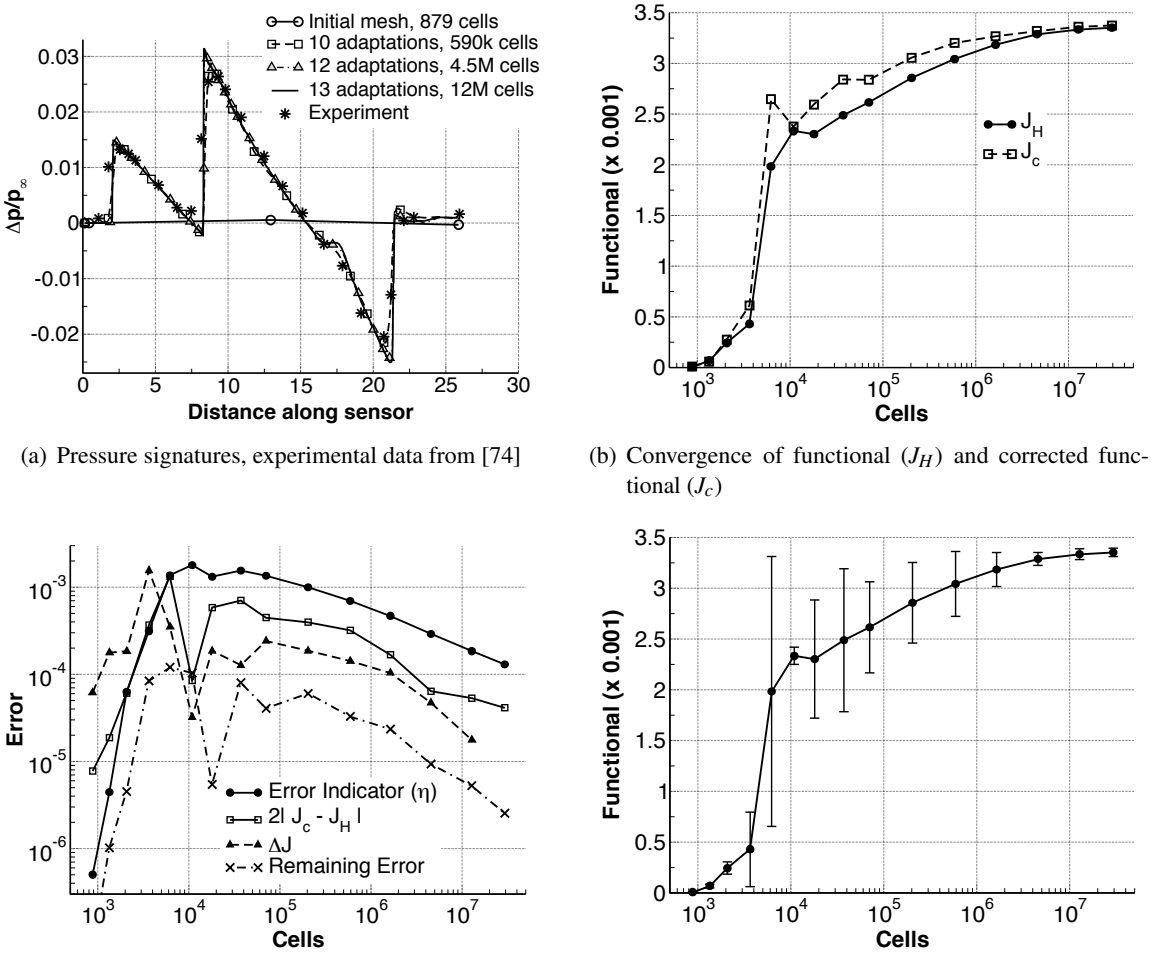
Figure 21. Contours of density-adjoint ($|\psi_0|$, white 0, black 0.012) on symmetry plane.

the refinement near the sensor (middle frame of Figure 20 and Figure 21) to accommodate both the flow and adjoint solutions on the same mesh. This is similar to the discussion around Figure 13 in Subsection 6.2.

To demonstrate verification, Figure 22 presents convergence of the signature, the functional, and its error estimates. Figure 22(a) shows the classical, qualitative evidence of mesh convergence by examining changes in the signature as the mesh is refined. The main features of the signature are captured within ten adaptations and the signal is converged to plotting accuracy in twelve adaptations. We also validate the signature using experimental data ($*$ symbols) from [74]. The agreement between the experiment and the simulation is excellent.

Figures 22(b)–22(d) show quantitative evidence of convergence as the mesh is refined. Figure 22(b) shows convergence of the functional J_H (solid circles) and the corrected functional J_c (open squares). Changes in these quantities begin to taper once the mesh reaches 200,000 cells, where the corrected functional begins to predict the value of the functional on the next mesh with increasing accuracy. Figure 22(c) shows convergence of the various error estimates. In general, the error increases over the first five cycles and thereafter decreases. As in the airfoil database example (recall Figure 16), the increasing error indicates new features emerging in the flowfield due to the coarse initial mesh.

The top line (solid circles) in Figure 22(c) shows convergence of the error-indicator bound η. The second line (squares) shows convergence of the error estimate $\mathscr{E}$ of Eq. 37 with $C = 2$. This assumes J_H is $\mathscr{O}(h)$, which is conservative and expected because both the flow and geometry are dominated by non-smooth features. The error-indicator bound is larger than the error estimate, but both show similar convergence behavior. The third line (solid triangles) monitors the magnitude of the functional update ($\Delta J = |J_H^i - J_H^{i-1}|$). Once the mesh reaches 70,000 cells, the update begins to converge and becomes smaller than the error estimate. The fourth line ($\times$ symbols) shows the magnitude of the remaining error term, Eqs. 36 and 38 without localization. Localization increases the magnitude of the error indicator η by about two orders. This is only slightly larger than what is shown in Figure 10(b) for the vortex problem despite the significantly more complex flow. Furthermore, this confirms that the remaining error term in Eq. 32 is small relative to the adjoint correction.



(a) Pressure signatures, experimental data from [74]

(b) Convergence of functional (J_H) and corrected functional (J_c)

(c) Convergence of error estimates

(d) Functional convergence with error bars showing estimate of discretization error

Figure 22. Mesh convergence for the pressure signature example.

Figure 22(d) combines information presented in figures 22(b) and 22(c) to concisely demonstrate simulation verification. The convergence of the functional is shown with error bars that represent the estimated discretization error $\mathscr{E}$, which is taken from the second line of Figure 22(c). The largest error occurs on the fourth adaptation and the error estimate sharpens considerably by the tenth adaptation (600,000 cells). Starting from this mesh, the error estimates reliably bracket the solution obtained on the finest mesh (29.4 million cells), indicating asymptotic convergence.

Figure 23 shows error histograms and the mean error value for the initial, fifth and final meshes. The left frame of Figure 23 shows that most of the cells of the initial mesh contribute no error, which is due to the uniform starting mesh. The middle frame of Figure 23 shows the histogram after five adaptations (sixth mesh), where the error indicator η reaches its maximum value, as shown in Figure 22(c). This mesh contains about 11,000 cells and the histogram shows the beginnings of error equidistribution, with the mode of the distribution already close to the mean. The right frame of Figure 23 presents the histogram of the final mesh. The mode moved to the left, the mean stayed close to the mode and the error distribution sharpened considerably.

In terms of performance, the total simulation (wall-clock) time was 1 hour 42 minutes on 96 Intel

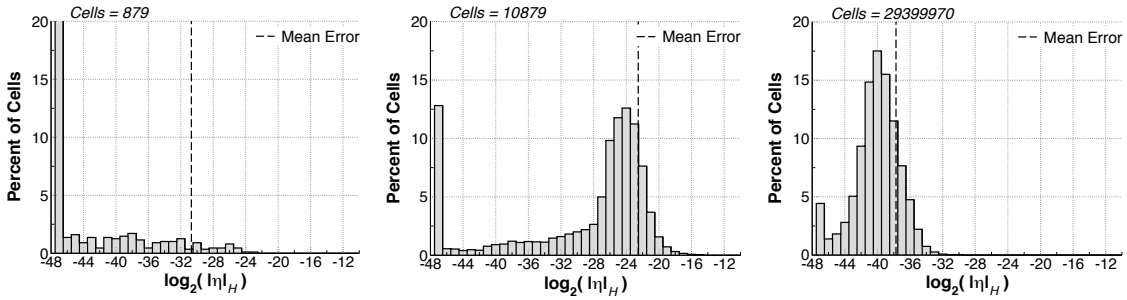


Figure 23. Histograms showing the distribution of error-indicator values for the initial mesh (left), mesh after five adaptations (middle) and final mesh (right) for the pressure signature example.

Xeon E5-460L cores [77]. The flow and adjoint solutions on the final mesh (29.4 million cells) required about 20 minutes each. Figure 22(d) indicates that in an engineering setting it would be sensible to terminate the simulation after twelve adaptations (omit the last two solutions). Convergence of the functional and error becomes predictable by the tenth adaptation and the level of error is quite small after twelve adaptations (4.5 million cells). Moreover, Figure 22(a) shows that after twelve adaptations there are essentially no changes in the signature. If the simulation is stopped after twelve adaptations, the wall-clock time is only 18 minutes, where the flow and adjoint solutions require about 4 minutes on the final mesh. Additional speedup is possible by omitting the error analysis on the final mesh, which would reduce the net wall-clock time to about ten minutes on these CPUs.

7.3. Launch-Abort Vehicle

The final example involves the prediction of aerodynamic forces for a launch-abort vehicle prototype. The example explores the limits of Procedure 1 when applied to problems that contain regions of separated, recirculating flow. The vehicle consists of a crew module with a tower-mounted abort system, as shown in figures 24 and 25. During a launch emergency, the four Abort Motors (AMs) ignite to pull the crew module safely away from the rocket stack. Throughout the abort, stability and control of the vehicle is maintained by differential thrust from eight Attitude Control Motors (ACMs) near the nose. For this example, we consider a high-altitude abort case studied previously in [78] at $M_\infty = 4$ and $\alpha = 20^\circ$ that involves strong ACM jets.

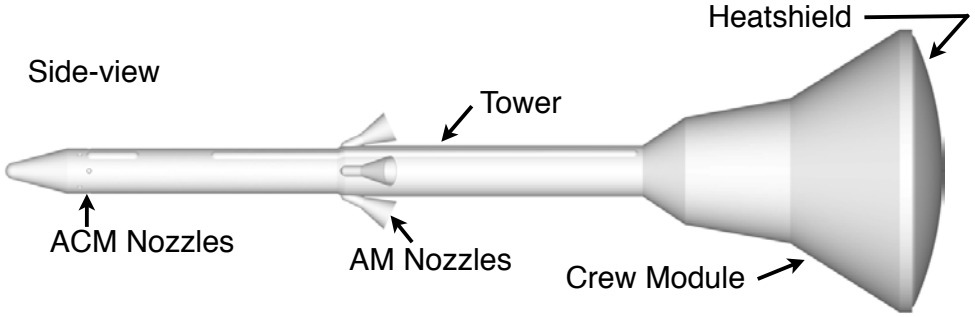


Figure 24. Launch-abort vehicle surface geometry.

A close-up view of the nose with the ACM nozzles is shown on the left side of Figure 25. The jet boundary conditions are applied at the throat faces of the nozzles, which are recessed inside

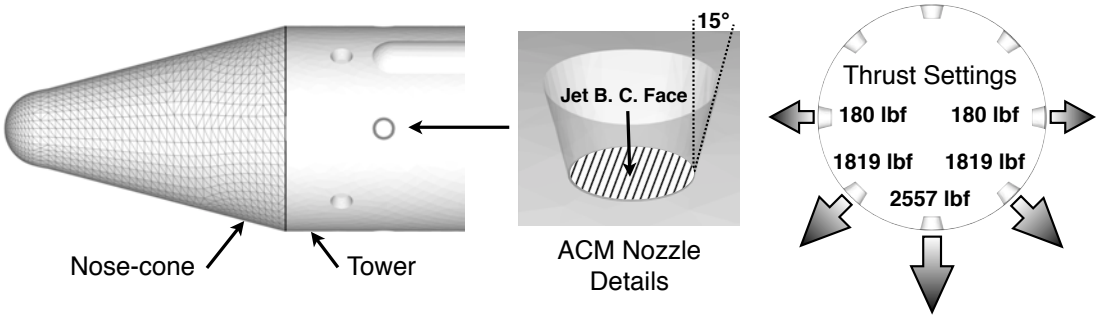


Figure 25. Close-up view of the nose-cone (shown with triangulation) and the ACM nozzles. The thrust settings are indicated in the cutaway view of the right frame.

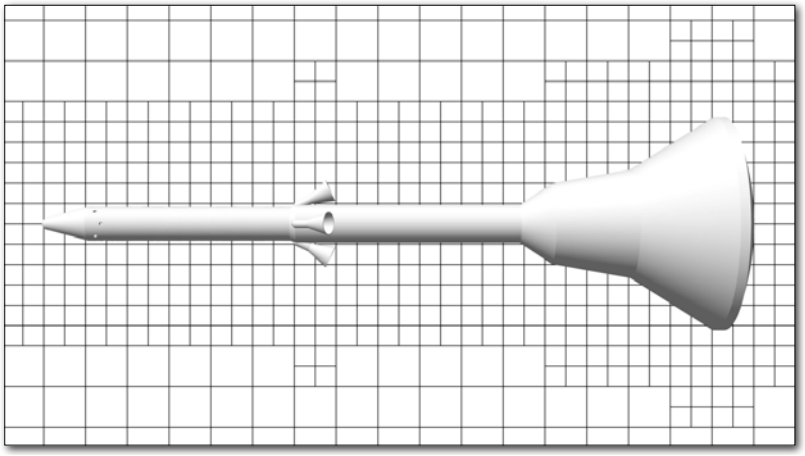


Figure 26. Near-body view of initial mesh on symmetry plane.

the tower, as shown in the middle frame of Figure 25. The jet boundary conditions match the exit momentum, pressure and thrust of the ACM performance data [78]. The right side of Figure 25 shows the thrust setting for each nozzle. The largest thrust is generated by the bottom (south-facing) nozzle and five of the eight nozzles are active.

The surface discretization of the vehicle contains roughly 380,000 triangles and is obtained directly from a CAD model. Figure 25 shows an example of the triangulation for the nosecone component. Figure 26 shows the initial mesh containing 12,880 cells, with only 580 cut cells. As in the previous examples, there is no bias to anticipate features of the flow solution and multiple ACM nozzles are contained within a single cut cell. The refinement threshold for each adaptation cycle is set to the mean error value if the error indicator is decreasing, otherwise it is set to twice the mean value (same as the pressure signature example described in Subsection 7.2). The output of interest is a weighted sum of normal and axial force coefficients:

$$J = C_N + 0.4C_A \tag{42}$$

and N_{max} is set to 50 million cells.

The simulation uses the *minmod* limiter and runs for 13 adaptations. For the last two adaptations the mesh growth is fixed at 2.5 and 3. Figure 27 shows the near-body mesh after 12 adaptations. The mesh contains 16.9 million cells with the finest cells located near the nose, in the windward region near the centerline and at the heatshield shoulder. The refinement of the wake and above the

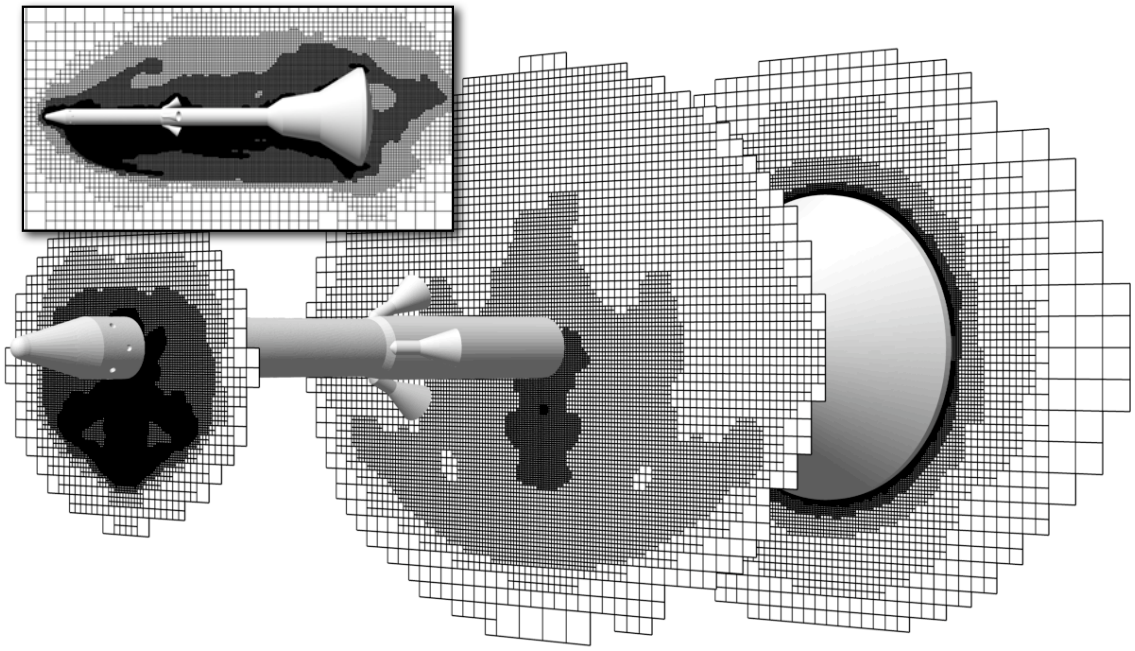


Figure 27. Near-body views of the mesh after 12 adaptations ($M_\infty = 4$ and $\alpha = 20°$).

vehicle is moderate, as shown in the inset of Figure 27. Figures 28 and 29 show the solution after 12 adaptations. The Mach number contours reveal the interaction of several shocks and expansions. The left frame of Figure 29 shows a close-up of the nose bow-shock and the south (downward-pointing) ACM jet on symmetry plane. The bow shock impinges on the ACM jet shock and creates a channel of supersonic flow that decelerates through a series of shocks to subsonic flow just upstream of the ACM nozzle, as shown in the left inset of Figure 29. This is a well-known shock-shock interaction pattern that Edney [79] classified as Type IV interference. The right frame of Figure 29 shows the front view of the solution on a plane just behind the ACM nozzles. Note the distortion of the bow shock due to the ACM jets.

The salient feature of the flow is the interaction of the ACM jets with the abort motor and crew module surfaces. The jet paths cannot be determined in advance of the computation, yet they have a significant influence on forces, moments, and heat transfer. In particular, the interaction of the jets with the surrounding flow can adversely affect the ACM's authority over pitching moment [78]. Figures 27 and 28 show that the finest cells track the bow and jet shocks away from the body and the first transverse cutting plane shows a highly refined mesh for the three main jets near the nose. Such fine cells are necessary because discretization errors introduced in this upstream region influence the functional over the rest of the vehicle. Further along the vehicle, the second cutting plane of Figure 27 shows the adaptation pattern still tracking the jet paths, but with less refinement. The final cutting plane through the heatshield shoulder is nearly symmetric, indicating that at this downstream station the jets have negligible influence on the functional.

To further investigate the refinement of the jets, we use isosurfaces of stagnation enthalpy to visualize the jet paths around the vehicle in Figure 30. The jet surfaces are shaded by Mach number. The bifurcation of the main jet is striking. This is a consequence of the overpressure at the ACM nozzle exit caused by the Type IV interference, followed by downstream interactions of the jet with the supersonic flow. The bifurcated jet contacts the sides of the crew module as it is swept upward by the high angle-of-attack flow. The "Below" view of Figure 30 shows that the shocks from the abort motor

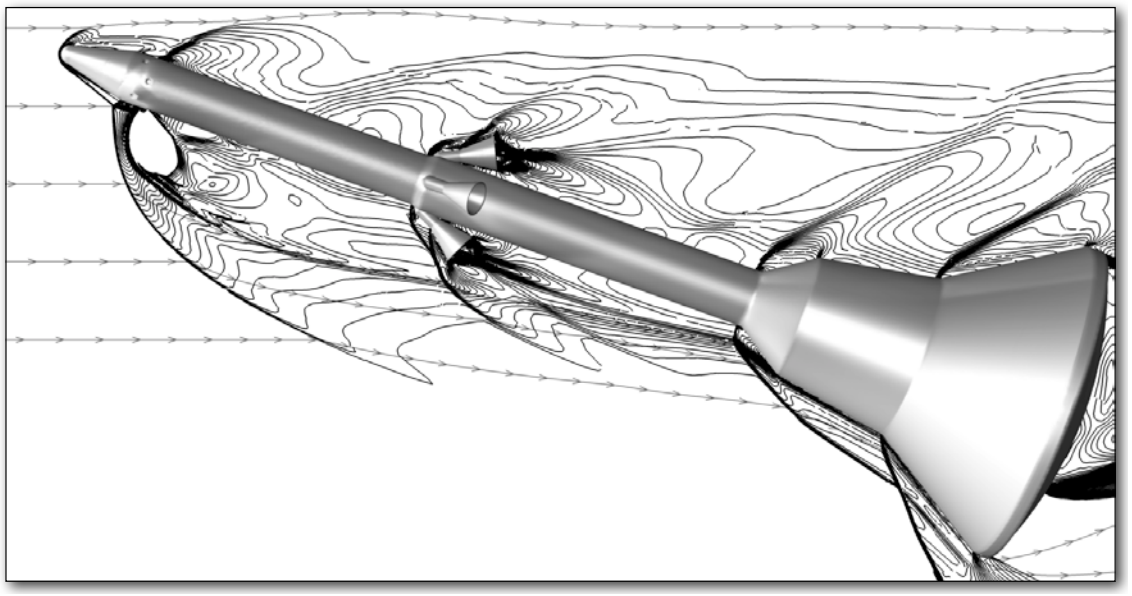


Figure 28. Near-field view of Mach isocontours after 12 adaptations on the symmetry plane. Contours are compressed using $\sqrt{M}$ to improve jet visualization. The surface of the vehicle is shaded by pressure coefficient ($M_\infty = 4$ and $\alpha = 20^\circ$).

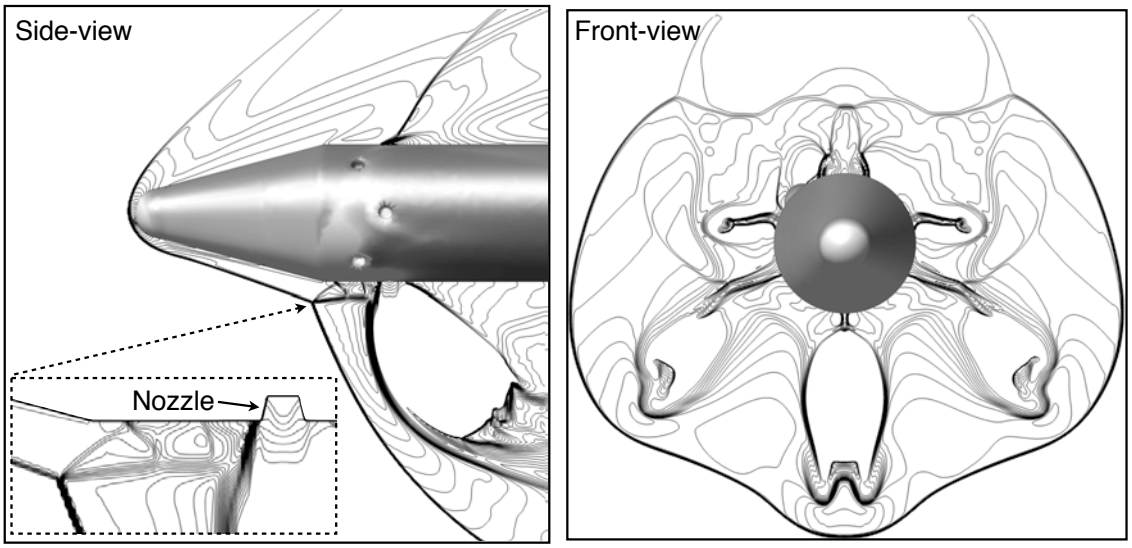


Figure 29. Close-up views near the nose showing Mach isocontours: side-view on symmetry plane (left) and front-view with a cutting plane just behind ACM nozzles (right). Inset on left shows details of the ACM shock interaction ($M_\infty = 4$ and $\alpha = 20^\circ$).

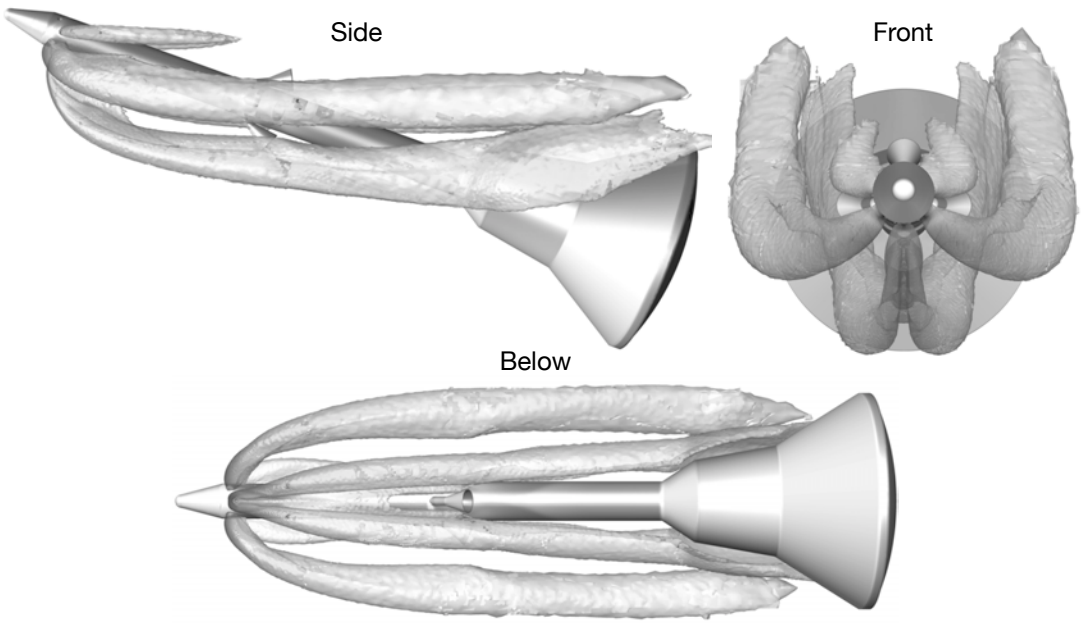
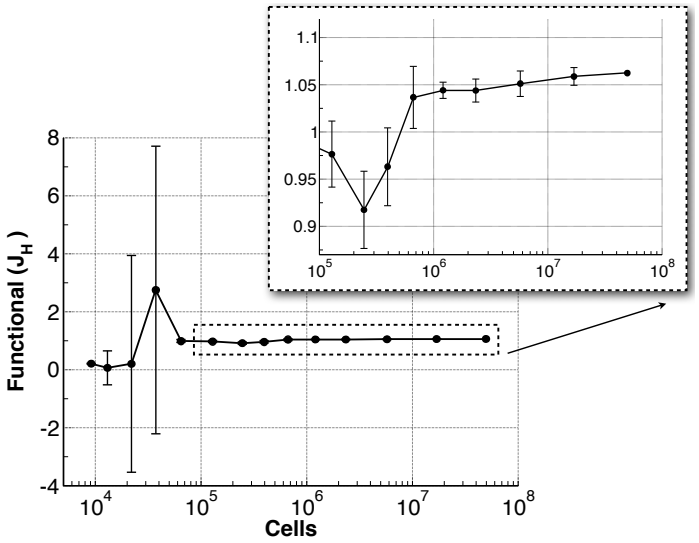


Figure 30. Jet paths around the vehicle ($M_\infty = 4$ and $\alpha = 20°$).

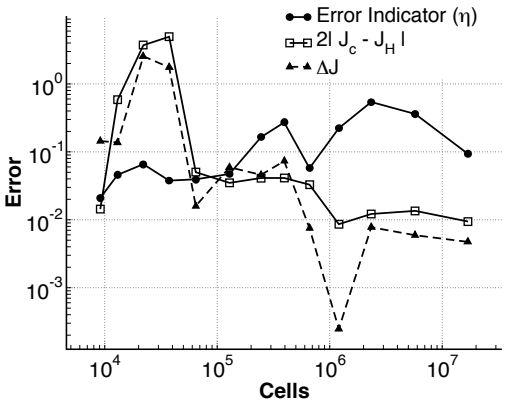
nozzles alter the jet paths away from the crew module and thereby alter the heating experienced by the crew module. Lastly, as indicated previously in Figure 27 and shown in Figure 30, the refinement of the jets wanes as the flow approaches the heatshield shoulder. While in reality the jets persist well into the wake of the vehicle, the error analysis truncates their refinement once they pass beyond the heatshield, where they can no longer impact the functional. This is the same behavior as we observed in the supersonic regime of the airfoil database example, but here in a much more complex flow.

Figure 31 summarizes the convergence of the functional, the error estimates, and the aerodynamic forces. Figure 31(a) shows the value of the functional on each mesh along with error bars representing the level of discretization error $\mathscr{E}$ via Eq. 37 with $C = 2$. As in the pressure signature example, we assume that J_H is $\mathscr{O}(h)$ because the flow is dominated by shocks and contact discontinuities. The large error estimates in the second and third adaptation are primarily associated with refinement of the ACM jets. The inset in Figure 31(a) shows that over the last three adaptations the error bars bracket the functional and the changes in the functional are less than 1%. Figure 31(b) shows convergence of the error-indicator bound η, the error estimate $\mathscr{E}$, and magnitude of the functional update ($\Delta J = |J_H^i - J_H^{i-1}|$). While convergence is not as convincing as in the previous examples, all three error measures are decreasing once the mesh reaches about 2 million cells.

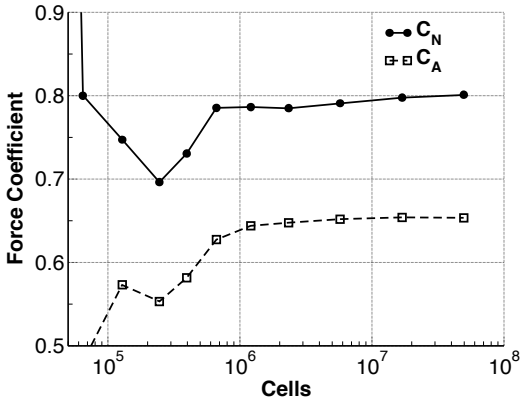
Despite the complicated flowfield, especially the recirculations behind the heatshield and the inactive abort motors, Procedure 1 remains effective in addressing critical regions of the flow that influence the outputs. Figure 31(c) shows that there are virtually no changes in the axial and normal force coefficients over the last two adaptations. More broadly, this example characterizes the performance of Procedure 1 in verification of difficult engineering simulations, and demonstrates the benefits of affordable and automatic error control in providing insight into complex flowfields.



(a) Functional convergence with error bars showing estimate of discretization error



(b) Convergence of error estimates



(c) Convergence of normal and axial force coefficients

Figure 31. Mesh convergence for the launch-abort vehicle example.

8. Conclusions

A procedure for automatic control of discretization error in steady simulations of inviscid flow has been presented. The procedure involves error estimates based on the method of adjoint-weighted residuals in conjunction with incremental mesh enrichment based on a Cartesian cut-cell method. We draw the following conclusions from our results:

- The code verification example demonstrates that both the functional J_H and corrected functional J_c are $\mathscr{O}(h^2)$, which is the expected order of accuracy. The error-indicator bound η converges at about the same rate due to the discontinuous adjoint field and localization. This is sufficient for the discretization error estimate $\mathscr{E}$ to be $\mathscr{O}(h^2)$ with effectivity close to 1.
- The simulation verification examples show that once the mesh is sufficiently fine, $\mathscr{E}$ reliably brackets J_H and decreases by about a factor of two per adaptation cycle.
- The error map $|\eta|_H$ reliably identifies critical regions of the mesh. Consequently, the procedure reliably handles extremely coarse initial meshes and generates a converging sequence of affordable meshes that accurately predict the output.
- The procedure offers a practical alternative to the manual generation of simulation-specific meshes that encompass the knowledge and experience of specialists and instead automatically delivers meshes and error estimates that provide significant insight into the simulation.
- Since a mesh refinement study is intrinsic to every simulation, the procedure automatically provides convergence histories of outputs of interest and error estimates. This makes the task of obtaining verified simulations straightforward and essentially automatic.
- Returning to the opening quote of the Introduction, the results show that the procedure attains a sufficient level of reliability and robustness for routine use in simulation verification.

There are many areas for future work. For example, improving the accuracy of the embedded adjoint representation through the use of affordable approximate solutions could improve the sharpness of the error estimate. Incorporating the procedure within simulation-based design could significantly reduce cost of optimization problems. More broadly, since simulations in practice frequently involve some degree of unsteady flow (as in the launch abort example described in Subsection 7.3), the procedure should be extended to transition smoothly from steady to unsteady flow, especially for applications where the time variation in the outputs is relatively small. Furthermore, the extension of the procedure to handle flows involving multiple components, or species, and turbulence is an important area of future work.

Acknowledgments

This work was supported by the NASA Ames Research Center contract NNA10DF26C and by NASA mission directorates for Aeronautics Research, and Human Exploration and Operations. The authors gratefully thank Marsha Berger (NYU), Mathias Wintzer (Analytical Mechanics Associates, Inc.) and George Anderson (Stanford University) for helpful discussions and suggestions.

References

1. Baker, T. J., "Mesh Adaptation Strategies for Problems in Fluid Dynamics," *Finite Elements in Analysis and Design*, Vol. 25, No. 3/4, 1997, pp. 243–273.

2. Bussoletti, J. E., Johnson, F. T., Bieterman, M. B., Hilmes, C. L., Melvin, R. G., Young, D. P., and Drela, M., "TRANAIR: Solution adaptive CFD modeling for complex 3D configurations," *Recent developments and applications in aeronautical CFD*, Proceedings of the 1993 European Forum, London, United Kingdom, 1993, pp. 10.1–10.14.

3. Löhner, R., "Mesh Adaptation in Fluid Mechanics," *Engineering Fracture Mechanics*, Vol. 50, No. 5/6, 1995, pp. 819–847.

4. Habashi, W. G., Dompierre, J., Bourgault, Y., Fortin, M., and Vallet, M., "Certifiable Computational Fluid Dynamics Through Mesh Optimization," *AIAA Journal*, Vol. 36, No. 5, 1998, pp. 703–711.

5. Fidkowski, K. J. and Darmofal, D. L., "Review of Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics," *AIAA Journal*, Vol. 49, No. 4, April 2011, pp. 673–694.

6. Roache, P. J., "Verification of Codes and Calculations," *AIAA Journal*, Vol. 36, No. 5, May 1998, pp. 696–702.

7. Oberkampf, W. L. and Trucano, T. G., "Verification and Validation in Computational Fluid Dynamics," *Progress in Aerospace Sciences*, Vol. 38, 2002, pp. 209–272.

8. Roy, C. J., "Review of Code and Solution Verification Procedures for Computational Simulation," *Journal of Computational Physics*, Vol. 205, 2005, pp. 131–156.

9. National Aeronautics and Space Administration (NASA), "Standard for Models and Simulations," Technical Standard NASA-STD-7009, 2008, http://standards.nasa.gov, visited March 2014.

10. Slotnick, J., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E., and Mavriplis, D., "CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences," NASA CR 2014-218178, March 2014.

11. Becker, R. and Rannacher, R., "An optimal control approach to a posteriori error estimation in finite element methods," *Acta Numerica 2000*, 2001, pp. 1–102.

12. Süli, E. and Houston, P., "Adaptive Finite Element Approximation of Hyperbolic Problems," *Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*, edited by T. Barth and H. Deconinck, Vol. 25 of *Lecture Notes in Computational Science and Engineering*, Springer-Verlag, 2002.

13. Giles, M. B. and Pierce, N. A., "Adjoint error correction for integral outputs," *Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*, edited by T. Barth and H. Deconinck, Vol. 25 of *Lecture Notes in Computational Science and Engineering*, Springer-Verlag, 2002.

14. Barth, T., "Numerical Methods and Error Estimation for Conservation Laws on Structured and Unstructured Meshes," Lecture notes, von Karman Institute for Fluid Dynamics, Series: 2003-04, Brussels, Belgium, March 2003.

15. Venditti, D. A. and Darmofal, D. L., "Grid Adaptation for Functional Outputs: Application to Two-Dimensional Inviscid Flow," *Journal of Computational Physics*, Vol. 176, 2002, pp. 40–69.

16. Hartmann, R., "Multitarget error estimation and adaptivity in aerodynamic flow simulations," *SIAM J. Sci. Comput.*, Vol. 31, No. 1, 2008, pp. 708–731.

17. Dwight, R. P., "Heuristic *a posteriori* estimation of error due to dissipation in finite volume schemes and application to mesh adaptation," *Journal of Computational Physics*, Vol. 227, No. 5, 2008, pp. 2845–2863.

18. Wang, L. and Mavriplis, D. J., "Adjoint-based *h-p* adaptive discontinuous Galerkin methods for the 2D compressible Euler equations," *Journal of Computational Physics*, Vol. 228, 2009, pp. 7643–7661.

19. Krakos, J. A. and Darmofal, D. L., "Effect of Small-Scale Output Unsteadiness of Adjoint-Based Sensitivity," *AIAA Journal*, Vol. 48, No. 11, Nov. 2010, pp. 2611–2623.

20. Yi, L. Y., Allaneau, Y., and Jameson, A., "Continuous Adjoint Approach for Adaptive Mesh Refinement," AIAA Paper 2011–3982, June 2011.

21. Yano, M. and Darmofal, D. L., "An optimization-based framework for anisotropic mesh adaptation," *Journal of Computational Physics*, Vol. 231, No. 22, 2012, pp. 7626–7649.

22. Copeland, S. R., Lonkar, A. K., Palacios, F., and Alonso, J. J., "Adjoint-Based Goal-Oriented Mesh Adaptation for Nonequilibrium Hypersonic Flows," AIAA Paper 2013–0552, Grapevine, TX, Jan. 2013.

23. Park, M. A. and Darmofal, D. L., "Validation of an Output-Adaptive, Tetrahedral Cut-Cell Method for Sonic Boom Prediction," *AIAA Journal*, Vol. 48, No. 9, Sept. 2010, pp. 1928–1945.

24. Park, M. A., Lee-Rausch, E. M., and Rumsey, C. L., "FUN3D and CFL3D Computations for the First High Lift Prediction Workshop," AIAA Paper 2011–0936, Jan. 2011.

25. Hartmann, R., Held, J., and Leicht, T., "Adjoint-based error estimation and adaptive mesh refinement for the RANS and *k-w* turbulence model equations," *Journal of Computational Physics*, Vol. 230, No. 11, May 2011, pp. 4268–4284.

26. Richter, T., "Goal-oriented error estimation for fluid-structure interaction problems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 223–224, 2012, pp. 28–42.

27. Ceze, M. and Fidkowski, K. J., "Drag Prediction Using Adaptive Discontinuous Finite Elements," *Journal of Aircraft*, 2014.

28. Aftosmis, M. J., Berger, M. J., and Melton, J. E., "Robust and Efficient Cartesian Mesh Generation for Component-Based Geometry," *AIAA Journal*, Vol. 36, No. 6, 1998, pp. 952–960.

29. Berger, M. J. and Aftosmis, M. J., "Aspects (and Aspect Ratios) of Cartesian Mesh Methods," Proceedings of the 16th international conference on numerical methods in fluid dynamics, Arcachon, France, July 1998.

30. Colella, P., Graves, D. T., Keen, B. J., and Modiano, D., "A Cartesian grid embedded boundary method for hyperbolic conservation laws," *Journal of Computational Physics*, Vol. 211, 2006, pp. 347–366.

31. Fidkowski, K. J. and Darmofal, D. L., "A triangular cut-cell adaptive method for high-order discretizations of the compressible Navier-Stokes equations," *Journal of Computational Physics*, Vol. 225, No. 6, 2007, pp. 1653–1672.

32. Meinke, M., Schneiders, L., Günther, C., and Schröder, W., "A cut-cell method for sharp moving boundaries in Cartesian grids," *Computers & Fluids*, Vol. 85, 2013, pp. 135–142.

33. Demargne, A. A. J., Evans, R. O., Tiller, P. J., and Dawes, W. N., "Practical and Reliable Mesh Generation for Complex Real-world Geometries," AIAA Paper 2014–0119, Jan. 2014.

34. Nemec, M., Aftosmis, M. J., Murman, S. M., and Pulliam, T. H., "Adjoint Formulation for an Embedded-Boundary Cartesian Method," AIAA Paper 2005–0877, Reno, NV, Jan. 2005.

35. Nemec, M. and Aftosmis, M. J., "Adjoint Error Estimation and Adaptive Refinement for Embedded-Boundary Cartesian Meshes," AIAA Paper 2007–4187, Miami, FL, June 2007.

36. Nemec, M., Aftosmis, M. J., and Wintzer, M., "Adjoint-Based Adaptive Mesh Refinement for Complex Geometries," AIAA Paper 2008–0725, Reno, NV, Jan. 2008.

37. Aftosmis, M. J. and Berger, M. J., "Multilevel Error Estimation and Adaptive *h*-Refinement for Cartesian Meshes with Embedded Boundaries," AIAA Paper 2002–0863, Reno, NV, Jan. 2002.

38. Wintzer, M., Nemec, M., and Aftosmis, M. J., "Adjoint-Based Adaptive Mesh Refinement for Sonic Boom Prediction," AIAA Paper 2008-6593, Honolulu, HI, Aug. 2008, (26th AIAA Applied Aerodynamics Conference).

39. Aftosmis, M. J. and Nemec, M., "Exploring Discretization Error in Simulation-Based Aerodynamic Databases," *Proceedings of the* 21^{st} *International Conference on Parallel Computational Fluid Dynamics*, ParCFD 2009, May 2009.

40. Wintzer, M., "Span Efficiency Prediction Using Adjoint-Driven Mesh Refinement," *Journal of Aircraft*, Vol. 47, No. 4, July-August 2010, pp. 1468–1471.

41. Bakhtian, N. and Aftosmis, M. J., "Analysis of Inviscid Simulations for the Study of Supersonic Retropropulsion," AIAA Paper 2011-3194, Honolulu, HI, June 2011, (29th AIAA Applied Aerodynamics Conference).

42. Kless, J. E. and Aftosmis, M. J., "Analysis of Grid Fins for Launch Abort Vehicle Using a Cartesian Euler Solver," AIAA Paper 2011-3666, Honolulu, HI, June 2011, (29th AIAA Applied Aerodynamics Conference).

43. Aftosmis, M. J., Nemec, M., and Cliff, S. E., "Adjoint-Based Low-Boom Design with Cart3D," AIAA Paper 2011-3500, Honolulu, HI, June 2011, (29th AIAA Applied Aerodynamics Conference).

44. Kless, J., Aftosmis, M. J., Ning, S. A., and Nemec, M., "Inviscid Analysis of Extended-Formation Flight," *AIAA Journal*, Vol. 51, No. 7, July 2013, pp. 1703–1715.

45. Ning, S. A., Kroo, I., Aftosmis, M. J., Nemec, M., and Kless, J. E., "Extended Formation Flight at Transonic Speeds," *Journal of Aircraft*, January 2014, pp. 0–0, http://dx.doi.org/10.2514/1.C032385.

46. Berger, M. J., Aftosmis, M. J., and Murman, S. M., "Analysis of Slope Limiters on Irregular Grids," AIAA Paper 2005–0490, Reno, NV, Jan. 2005.

47. van Leer, B., "Flux-Vector Splitting for the Euler Equations," ICASE Report 82-30, Sept. 1982.

48. Aftosmis, M. J., "Solution Adaptive Cartesian Grid Methods for Aerodynamic Flows with Complex Geometries," Lecture notes, von Karman Institute for Fluid Dynamics, Series: 1997-02, Brussels, Belgium, March 1997.

49. Aftosmis, M. J., Berger, M. J., and Adomavicius, G., "A Parallel Multilevel Method for Adaptively Refined Cartesian Grids with Embedded Boundaries," AIAA Paper 2000–0808, Reno, NV, Jan. 2000.

50. Aftosmis, M. J., Berger, M. J., and Murman, S. M., "Applications of Space-Filling-Curves to Cartesian Methods for CFD," AIAA Paper 2004–1232, Reno, NV, Jan. 2004.

51. Berger, M. J., Aftosmis, M. J., , Marshall, D. D., and Murman, S. M., "Performance of a New CFD Flow Solver Using a Hybrid Programming Paradigm," *J. of Parallel and Distributed Computing*, Vol. 65, 2005, pp. 414–423.

52. Lu, J., *An* a posteriori *Error Control Framework for Adaptive Precision Optimization using Discontinuous Galerkin Finite Element Method*, Ph.D. thesis, Massachusetts Institute of Technology, 2005.

53. Hicken, J. E. and Zingg, D. W., "Dual consistency and functional accuracy: a finite-difference perspective," *Journal of Computational Physics*, Vol. 256, 2014, pp. 161–182.

54. Nemec, M., *Optimal Shape Design of Aerodynamic Configurations: A Newton–Krylov Approach*, Ph.D. thesis, University of Toronto, 2003.

55. Barth, T. J., "Parallel CFD Algorithms on Unstructured Meshes," *Special Course on Parallel Computing in CFD*, AGARD-R-807, Oct. 1995, pp. 7–1–7–41.

56. Giles, M. B., Duta, M. C., Müller, J.-D., and Pierce, N. A., "Algorithm Developments for Discrete Adjoint Methods," *AIAA Journal*, Vol. 41, No. 2, 2003, pp. 198–204.

57. Nielsen, E. J., Lu, J., Park, M. A., and Darmofal, D. L., "An Implicit, Exact Dual Adjoint Solution Method for Turbulent Flows on Unstructured Grids," *Computers & Fluids*, Vol. 33, 2004, pp. 1131–1155.

58. Mavriplis, D. J., "Multigrid Solution of the Discrete Adjoint for Optimization Problems on Unstructured Meshes," *AIAA Journal*, Vol. 44, No. 1, 2006, pp. 42–50.

59. Park, M. A., *Anisotropic Output-Based Adaptation with Tetrahedral Cut Cells for Compressible Flows*, Ph.D. thesis, Massachusetts Institute of Technology, 2008.

60. Giles, M. B., "On the iterative solution of adjoint equations," *Automatic Differentiation of Algorithms: From Simulation to Optimization*, edited by G. Corliss, C. Faure, A. Griewank, L. Hascoët, and U. Naumann, Springer-Verlag, 2002.

61. Giles, M. B., "On the Use of Runge–Kutta Time-Marching and Multigrid for the Solution of Steady Adjoint equations," Oxford University Computing Laboratory 00/10, June 2000.

62. Jones, W. T., Nielsen, E. J., and Park, M. A., "Validation of 3D Adjoint Based Error Estimation and Mesh Adaptation for Sonic Boom Prediction," AIAA Paper 2006–1150, Jan 2006.

63. Barth, T. J. and Jespersen, D., "The Design and Application of Upwind Schemes on Unstructured Meshes," AIAA Paper 1989–0366, Jan 1989.

64. Müller, J.-D. and Giles, M. B., "Solution Adaptive Mesh Refinement Using Adjoint Error Analysis," AIAA Paper 2001–2550, June 2001.

65. Dannenhoffer III, J. F. and Baron, J. R., "Adaptive Procedure for Steady State Solution of Hyperbolic Equations," AIAA Paper 84–0005, Jan. 1984.

66. Kallinderis, Y. G. and Baron, J. R., "Adaptation Methods for a New Navier–Stokes Algorithm," *AIAA Journal*, Vol. 27, No. 1, Jan. 1989, pp. 37–43.

67. Aftosmis, M. J., Gaitonde, D., and Tavares, T. S., "Behavior of Linear Reconstruction Techniques on Unstructured Meshes," *AIAA Journal*, Vol. 33, No. 11, Nov. 1995, pp. 2038–2049.

68. Abbott, I. H. and Doenhoff, A. E. V., *Theory of Wing Sections*, Dover Publications, Inc., 1959.

69. Page, J. A. and Plotkin, K. J., "An Efficient Method for Incorporating Computational Fluid Dynamics Into Sonic Boom Prediction," AIAA Paper 91–3275, Sept. 1991.

70. Cliff, S. E. and Thomas, S. D., "Euler/Experiment Correlations of Sonic Boom Pressure Signatures," *Journal of Aircraft*, Vol. 30, No. 5, Sept. 1993, pp. 669–790.

71. Meredith, K. B., Dahlin, J. H., Graham, D. H., Malone, M. B., Haering Jr., E. A., Page, J. A., and Plotkin, K. J., "Computational Fluid Dynamics Comparison and Flight Test Measurement of F-5E Off-Body Pressures," AIAA Paper 2005–0006, Jan. 2005.

72. Loseille, A., Dervieux, A., and Alauzet, F., "Fully anisotropic goal-oriented mesh adaptation for 3D steady Euler equations," *Journal of Computational Physics*, Vol. 229, 2010, pp. 2866–2897.

73. Cliff, S. E., Elmiligui, A. A., Campbell, R. L., and Thomas, S. D., "Refined Tetrahedral Meshes with Mach Cone Aligned Prisms for Sonic Boom Analysis," *Journal of Aircraft*, Vol. 50, No. 3, May 2013, pp. 778–790.

74. Hunton, L. W., Hicks, R. M., and Mendoza, J. P., "Some Effects of Wing Planform on Sonic Boom," NASA TN D–7160, Jan. 1973.

75. Park, M. A., Aftosmis, M. J., Campbell, R. L., Carter, M. B., Cliff, S. E., and Bangert, L. S., "Summary of the 2008 NASA Fundamental Aeronautics Program Sonic Boom Prediction Workshop," AIAA Paper 2013–0649, Jan. 2013.

76. Aftosmis, M. J. and Nemec, M., "Cart3D Simulations for the First AIAA Sonic Boom Prediction Workshop," AIAA Paper 2014–0558, Jan. 2014.

77. NASA, "Endeavour Supercomputer, NAS Division," April 2014, http://www.nas.nasa.gov/hecc/resources/endeavour.html.

78. Aftosmis, M. J. and Rogers, S. E., "Effects of Jet-Interaction on Pitch Control of a Launch Abort Vehicle," AIAA Paper 2008–1281, Jan. 2008.

79. Edney, B. E., "Anomalous Heat Transfer and Pressure Distributions on Blunt Bodies at Hypersonic Speeds in the Presence of an Impinging Shock," FFA Report 115, The Aeronautical Research Institute of Sweden, June 1968.

Uncertainty Quantification and Sensitivity Analysis—Some Fundamental Concepts, Terminology, Definitions, and Relationships

Vicente J. Romero

Sandia National Laboratories,[1] Albuquerque, NM 87185

Abstract

This chapter identifies and discusses many types and sources of uncertainty in experiments, models, and simulations. The ubiquitous presence of uncertainty in these primary instruments of science and engineering motivates the importance of uncertainty quantification and sensitivity analysis. Accordingly, some fundamental concepts, terminology, definitions, and relationships pertaining to uncertainty quantification and sensitivity analysis are presented and discussed.

1. Introduction

The following definition of *uncertainty* is adapted from the NASA General Safety Program Requirements [1] as stated in the NASA Technical Standard for Models and Simulations [2]:

> *Uncertainty* is a broad and general term used to describe an imperfect state of knowledge or a variability resulting from a variety of factors including, but not limited to, lack of knowledge, applicability of information, physical variation, randomness or stochastic behavior, indeterminacy, judgment, and approximation.

The following definition of *uncertainty quantification* (UQ) is a close compromise between the NASA M&S Standard [2] and ASME [3] definitions:

> *Uncertainty Quantification* is the process of characterizing all significant uncertainties in a model, simulation, or experiment and of quantifying their effect on computed or experimental results.

A similar definition specifically addressed to computational simulation is given by the National Academy of Sciences in [4], slightly modified here by the words in brackets:

> *Uncertainty Quantification* is the process of quantifying uncertainties in a computed QOI [*quantity of interest*], with the goal of accounting for all [significant] sources of uncertainty and quantifying the contributions of the specific sources to the overall uncertainty [in the computed results].

[1] Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

The latter part of this definition touches on *sensitivity analysis* (SA) as follows.

> *Sensitivity analysis* in the context of UQ is the determination of how much uncertainty an individual source contributes to the total uncertainty in a simulated or experimental QOI.

SA enables resources to be focused on characterization and/or propagation of the most influential uncertainty sources. But SA does not necessarily involve uncertainty in the parameter variables. For example, SA is also used in optimization procedures to move in design space. The following definitions are more general in this regard.

> From Ferston and Troy [5]: "*Sensitivity analysis* is a study of how changes in the inputs to a model influence the results of the model." The National Academies definition [4] states that *sensitivity analysis* is an exploration, often by numerical (rather than analytical) means, of how model outputs (particularly QOIs) are affected by changes in the inputs (parameter values, assumptions, etc.).

Tables 1 and 2 summarize various commonly encountered sources of uncertainty in experiments, models, and simulations. Some additional explanation regarding several of the categories under Models and Simulations in Table 2 is given next. This provides helpful context for the rest of this chapter. Further explanation on the items in tables 1 and 2 is provided in later sections.

Table 1. Some Sources of Uncertainty in Experiments and Experimental Data

Sources of Uncertainty	Description
Measurement/estimation uncertainties	Sensor and data acquisition system inaccuracies, etc., occur in measurement or estimation of: • Experimental conditions and inputs, e.g., initial and boundary conditions. • System responses, behaviors, outputs.
Extrapolation-bias uncertainty	Occurs when bias corrections and uncertainty characterizations of experimental results are extrapolated from where quantified originally.
Random variability over replicate tests	Examples are unit-to-unit geometric and physical variability in replicate tests and variability of measurement accuracy in the different tests.
Data processing and inference uncertainties	Examples: • Uncertain bias error in temporal and/or spatial field point-data interpolation, integration, trend extrapolation, etc. • Uncertain bias in inferring the full population of a random quantity (e.g., a frequency distribution or probability density) from limited data samples. • Uncertain bias error in uncertainty propagation and aggregation procedures that combine data measurement, processing, and inference uncertainties.

Table 2. Some Sources of Uncertainty in Models and Simulations

Sources of Uncertainty	Description
Phenomenological modeling uncertainty expressed through use of multiple model forms	Examples: • Multiple different plausible turbulence models. • Multiple equation-of-state models and material constitutive models. • Geometry models (e.g., model A with bolts explicitly modeled vs. B without explicit bolts).
Uncertain values for model input parameters, or *multiple* values for input parameters	Example: a population of multiple values described by a frequency distribution.
Model prediction-bias uncertainty	Whatever the model form and parameter values, the approximations involved will generally result in some degree of prediction bias even if the mathematical equations are solved exactly and initial & boundary conditions inputs to the model are exact. (Model validation seeks to quantify model prediction bias; model calibration and/or result correction functions seek to reduce it.)
Numerical solution-bias uncertainty	This is associated with spatial and temporal discretizations of continuum physics mathematical models and incomplete iterative convergence of the discretized equations being solved. (This is what *solution or calculation verification* seeks to quantify.)
Simulation results processing and inference uncertainties	Examples: • Uncertain bias in temporal and/or spatial field point-value interpolation, integration, trend extrapolation, etc. • Uncertain bias in uncertainty propagation and aggregation procedures for the modeling and simulation uncertainties above.

Uncertainty often exists regarding the most appropriate way to mathematically represent or model particular physical phenomena, geometry, boundary and initial conditions, materials properties and behaviors, and so on. This representational "model form" uncertainty is sometimes explicitly expressed through the presence and use of multiple candidate model forms or structures that are distinct from each other. That is, they are not just simple parametric variations of each other given by the same model form with different numerical values for the parameters. For a given model form, prescribed variations of its parameter values may be used to model stochastically varying system attributes and phenomena. Other parameter uncertainties may stem from lack of knowledge regarding the parameter values, and/or may represent an attempt to compensate for model-form deficiencies. Indeed, whatever the model form and parameter values, the approximations involved will generally result in some degree of prediction bias even if the mathematical equations are solved exactly and initial and boundary conditions input to the model are exact.

Model calibration seeks to reduce prediction bias error. Model-form and/or parameter range constraints may prevent the model results from exactly matching the data being calibrated to. A calibration "matching or fitting gap" occurs particularly often when an interval or distributional uncertainty (due to experimental measurement uncertainties and/or data processing and inference uncertainties) of a scalar result is to be matched, or a vector of responses is to be matched (with experimental uncertainty or not)—such as temporal and/or spatial response fields and/or several datasets from several experiments (at experimental conditions that are nominally the same or purposefully different). Calibration results and any fitting gap are functions of the calibration procedure used, and of any model-form and parameter errors and uncertainties that are not calibration degrees of freedom but come into the calibration as measured, processed, modeled, and inferred elements of the calibration experiments—such as initial conditions (ICs), boundary conditions (BCs), and geometries of the system being calibrated. The calibration results and any fitting gap are also dependent on numerical solution bias error/uncertainty in the calibration model. Sometimes, correction functions are applied to adjust QOI results to approximately correct for any fitting gap after calibration. When calibration and/or model output correction functions are used, the author refers to this as "model conditioning" [12, 13, 15, 35]. Model conditioning is discussed more in Section 2.

All these components of model prediction bias are subject to another (*extrapolative*) component of model prediction bias when predictions are made at conditions (ICs, BCs, geometry, etc.) different from the calibration conditions. Due to unavoidable imperfections in model-form and parameter values, even if a fully converged model with no numerical solution error can be calibrated and/or output-adjusted to perfectly match experimental results at the calibration conditions, under extrapolation to the other conditions the model will generally exhibit extrapolative prediction bias and/or significant uncertainty thereof.

A practical objective of *model validation* is to quantify prediction bias and uncertainty for an identified QOI when the model is assessed at validation conditions that present relevant and significant extrapolative tests of model predictivity away from the calibration conditions. Besides extrapolation related error/uncertainty, the net prediction bias/uncertainty at the validation conditions depends on numerical solution bias/uncertainty there. Validation quantification of prediction bias should also include any significant uncertainties of measured, processed, modeled, and inferred elements of the validation experiments, such as ICs, BCs, and geometries. How to best use a model's validation- or calibration-characterized prediction bias and uncertainty to potentially adjust or bias-correct the model to mitigate prediction risk beyond the validation or calibration conditions is a very difficult question and an active area of research. This is discussed more in Section 2.

Ultimately, estimation of prediction bias error and uncertainty at points where no experimental data exists should include extrapolation of validation-quantified prediction bias/uncertainty (after any model and output QOI corrections for this), and should also include estimated error/uncertainty associated with this extrapolation. These uncertainties should be combined with the effects of "traveling" uncertainties intrinsically affiliated with the model (see Section 2) and with "non-traveling" uncertainties particular to the new prediction setting's ICs, BCs, geometries, numerical solution bias, and result processing and inference procedures.

Figure 1 presents an integrative high-level diagram of some important aspects and elements of UQ and SA that can arise in analysis projects involving experiments, modeling, and simulation. Many UQ elements in the figure have already been introduced and others remain to be introduced. All are important to recognize in connection with engineering analysis and are briefly addressed in this chapter.

The top row of the diagram shows a linking between uncertainty quantification, sensitivity analysis, and UQ-based resource allocation. *UQ-based resource allocation* is the quantification, management, and optimization of experimental, analysis, and computational resources such that their value-added impact on total QOI uncertainty is adequately estimated and used to prioritize activities and resources to most cost-effectively quantify, control, and reduce uncertainty. UQ and SA results, along with time, labor, and resource cost models for experimentation and analysis, are necessary to enable UQ-based resource allocation. Current approaches to UQ-based resource allocation in engineering practice seem to be relatively ad hoc. This does not necessarily mean that such resource allocation is ineffective in practice, but more formal methods for UQ-based resource allocation are actively being researched and developed, e.g. Refs. 6-8.

The Uncertainty Quantification subject box in the top row of the diagram is divided into three subject boxes in the middle row. These are respectively discussed in the next three sections of this chapter, including the sub-elements per the diagram. The chapter ends with a section that expresses some views on the importance of uncertainty quantification in science and engineering.

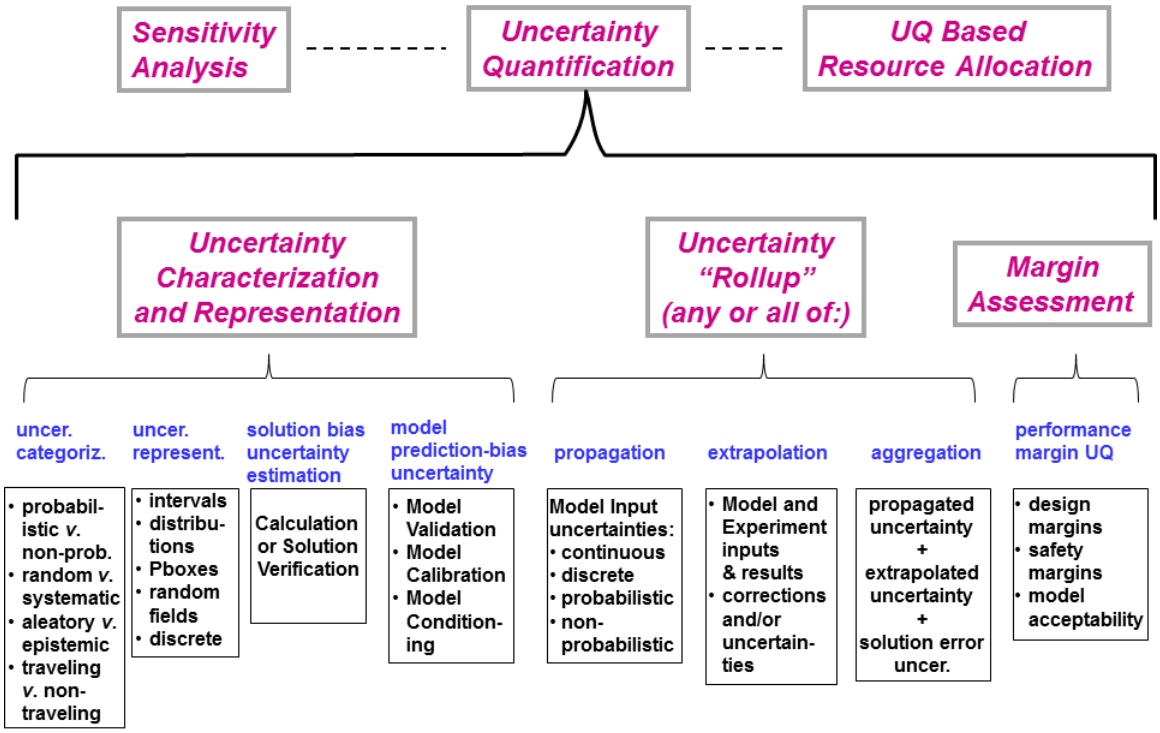


Figure 1. Some important aspects of uncertainty quantification and sensitivity analysis that can arise in analysis projects.

2. Uncertainty Characterization and Representation

Here we discuss the sub-elements of the Uncertainty Characterization and Representation box in the middle row of Figure 1, and address the definition of uncertainties categorically, interpretationally, and quantitatively/mathematically.

Uncertainties are categorized into various types according to their nature in the context of a given uncertainty analysis. The different natures determine the different implications and treatments of the uncertainties in an analysis. The different natures or types of uncertainty discussed here are:

aleatory vs. epistemic; probabilistic vs. non-probabilistic; random vs. systematic; and traveling vs. non-traveling.

Aleatory uncertainty characterizes the inherent randomness or variability of a quantity in a set or population of units, events, etc. (paraphrased from Helton, et al. [9]). For example, random variability of material properties in a set of units being tested or modeled is an aleatory uncertainty. Alternative terminologies include: random variability, stochastic variability, and irreducible uncertainty.

Aleatory uncertainties are usually characterized by random variables that are assigned probability density functions (PDFs), or their transform-equivalent cumulative distribution functions (CDFs). Probabilistic modeling is most firmly established when suitable experimental data exists as a basis for a frequency distribution of values. Even then, when very limited (sparse) data samples are available, significant uncertainty exists regarding the true PDF from which the samples come. Among five methods studied by Romero et al. [10], a simple Tolerance Interval approach for dealing with this PDF-form uncertainty was found to be most practical. The method has been found to be very robust in avoiding underestimation of true PDF variance from sparse samples of the PDF (for many different PDF shapes) [10, 18], but it can potentially be very conservative and exaggerate variance considerably. Despite this, the method is very simple and inexpensive, making it a pragmatic choice.

Epistemic uncertainty characterizes the lack of knowledge about the appropriate value to use for a quantity that is assumed to have a fixed value in the context of a specific application (paraphrased from Ref. 9). Alternative terminologies for epistemic uncertainty include: lack-of-knowledge uncertainty, subjective uncertainty, or reducible uncertainty.

Epistemic uncertainty regarding a quantity's value can be probabilistic or non-probabilistic. Uncertainty *intervals* are simple non-probabilistic representations of uncertainty, often used when the uncertainty is obtained largely from expert opinion and/or rough quantitative estimates. Epistemic uncertainty can be modeled with various other types of uncertainty representations between the extremes of interval and probabilistic PDF representations, such as Possibility Theory and Dempster-Shafer belief structures from Evidence Theory; see e.g. Helton et al. [9]. Such intermediate representations are relatively involved and expensive and are beyond the scope of this chapter. A simplified probability box representation approach is discussed later in this section for handling uncertainties involving both interval and PDF sources, where the PDF can represent either aleatory or epistemic uncertainty.

Aleatory and epistemic uncertainties signify different types of uncertainty, so it is often important to keep them separate in model calibration, validation, and margin analysis. This helps in interpreting the different implications of these uncertainties in the analysis, e.g., Refs. 9, 19, 25, 26, 31, and 32.

Another common distinction made between types of uncertainties comes from the experimental uncertainty literature: *systematic* vs. *random*.[2] *Systematic uncertainty* exists when measurement

[2] The terms *systematic uncertainty* and *random uncertainty* are not strictly proper. Nevertheless, the terms *systematic uncertainty, random uncertainty,* and *random variability* are commonplace in engineering literature and practice, and are used here. Paraphrasing from Ref. 11: the concept of uncertainty varying randomly from test to test is perplexing, perhaps even nonsensical. It is measurement errors themselves, or physical quantities themselves, that are conceived as randomly varying over a set of units or events, or systematically having the same value in the multiple units or events.

errors and/or properties, parameters, and inputs (such as boundary conditions) of a system or systems under consideration do *not* vary over a population of nominally similar units or events, but the quantities are uncertain in the epistemic sense. *Random uncertainty* or *random variability* exists when measurement errors or system conditions vary randomly over a set or population of nominally similar units or events. This is the equivalent of aleatory uncertainty.

A distinction is also made between *traveling* and *non-traveling* uncertainties [12-15] in the context of model validation and calibration when using the Real Space framework for comparing experimental and simulation results and their uncertainties [14-16, 18, 31, 32]. Non-traveling uncertainties in the experiments and/or simulations are those that do not "travel" consistently to model use beyond the calibration or validation setting. Conversely, traveling uncertainties *do* extrapolate consistently to downstream applications of the model. For example, innate random variability of a material in a system being tested and modeled is a traveling uncertainty, whereas random measurement variability in the tests is a non-traveling uncertainty; it will not be innately present in model prediction scenarios where the model is being used subsequent to the calibration or validation activity. Both examples are sources of random variability in the calibration or validation activity, but one is a traveling uncertainty and one is not. Thus, these two sources of random variability (even if both are modeled probabilistically) are treated differently in the Real Space uncertainty accounting and comparison system. This reflects their different significance and consequences to prediction error and uncertainty; see References 13 and 15.

Experimental or simulation uncertainty often comes in a discrete form as a set of experimental or simulated values of an uncertain quantity. Very often the discrete values are proposed to come from a continuum of values or possible values. Thus, they can be modeled with continuum representations of uncertainty, such as PDFs and intervals. But sometimes uncertainty cannot be treated with a continuum approach and must be treated with less familiar *discrete uncertainty* approaches. Such cases include discrete model forms, examples of which are given in Table 2. Reference 16 demonstrates a method for representing, propagating, and aggregating discrete model-form uncertainty (epistemic).

A situation not involving discrete model forms, but instead model inputs that come in discrete (not parametrically continuous) form, is exemplified in the following. In electronics modeling applications, the Gummel-Poon (GP) model parameters (often numbering 10 or more) determined through calibration are unique to each individual transistor tested [17]. When tests on nominally identical devices are performed, the resulting sets of GP parameters define different points in the calibration parameter space. However, the parameter space is generally not treated as a continuous parameter space that can be interpolated or extrapolated. The concern is that running the transistor model with parameter sets corresponding to other points in the space may not represent physically realizable devices, or may not reflect devices representative of the population from which the calibration devices came. These concerns appear applicable in many other calibration situations involving aleatory variations of the physical specimens or systems calibrated to. However, many popular calibration approaches do not address this concern.

Discrete random *function* inputs to models are also sometimes encountered. For example, consider multiple stress-strain curves from repeated (replicate) tests of a material. A continuum parametric or spectral representation is theoretically possible to infer from the multiple stress-strain curves from replicate tests of the material, but in practice this is very difficult to accomplish [18]. Instead, the authors of Refs. 18, 19, and 31-33 employ a very simple approach for approximately representing, propagating, and aggregating random variability information that comes in discrete form, whether discrete parameter sets as in the electronics case, or random functions like stress-strain curves.

When frequency-based probabilistic (PDF) sources of uncertainty and epistemic sources contribute to the uncertainty of a given quantity, then the combined uncertainty involves many potential candidate PDFs, as shown figuratively in Figures 2–4. It is usually prohibitively costly to work with populations of PDFs, so simplifying approximate treatments are necessary. In the special case of mixed aleatory and epistemic uncertainty arising from data sample sparseness, a tolerance interval approach is often effective as explained previously. Otherwise, when a mix of epistemic interval or effective-interval [34] sources and frequency-based probabilistic sources are present, a simplified approach to handling these disparate types of uncertainty using probability boxes is explained next.

Probability Boxes (Pboxes**)** [20, 21] are simplified representations of families or populations of PDFs/CDFs, in which the CDFs in the family do not cross each other. Thus, the extreme upper and lower CDFs of the family form a "probability box" which bounds all the CDFs in the family. This restriction is often met in real application situations, especially in model validation and calibration where relatively controlled (thus small) uncertainty magnitudes exist in the experiments and simulations and this limits significant interaction effects between the probabilistic and interval uncertainties. Because of the accommodating nature of this type of CDF family, uncertainty analysis often needs to be conducted with only the two bounding CDFs that comprise the Pbox in order to bound uncertainty results for the family of PDFs/CDFs. The concept is demonstrated in Refs. 19, 31, and 32 where even simpler and less expensive "Level 1" approximate Pboxes [34] are used to make the model validation assessments feasible by completely decoupling the probabilistic and the interval or effective-interval uncertainties.

Numerical Solution-Bias Uncertainty Estimation - Solution bias error typically comes from spatial and/or temporal discretization of the governing continuum physics equations and geometry, and from incomplete convergence in iterative solutions of the discrete equations (due to non-zero error tolerances needed for computational affordability). "Solution" or "calculation" verification attempts to quantify solution bias error.

Model Prediction-Bias Uncertainty – As established in Section 2, model validation attempts to quantify model prediction-bias uncertainty, and model conditioning (which may include model calibration) attempts to reduce prediction bias and uncertainty. Model validation and model conditioning involve experimental and simulation sources of uncertainty, which potentially involve everything in Figure 1 except for Margin Assessment, which is discussed in Section 4.

In model validation or calibration, model results may not adequately match experimental results. But because validation and calibration are fairly complex and expensive endeavors, generally the model being worked with is the best that can be afforded or obtained, so the model will not typically be rejected and abandoned if it does not meet accuracy goals. Rather, efforts will usually first be made to better reconcile the model with reality. Usually the model or some modified version will be leveraged for prediction needs, even if that means lowered expectations and reduced prediction duties and domain of applicability.

Reconciliation can come from investigating experimental and modeling factors that are perceived to contribute most to the failure to meet the accuracy goals, with remediation of the factors as practical. This can take the form of: more accurate and precise measurements and control of experimental inputs and conditions; more experiments if experimental uncertainty is being significantly driven by too few experimental samples of stochastically varying phenomena; improved data processing procedures; reducing the discretization-related uncertainties in the model solutions; improving the model form through modified and/or added behavioral mechanism and parameters.

If these types of actions do not fully reconcile the agreement between the model and experiments, or cannot be afforded or conducted, then the best chance for success in upcoming uses of the model may be to condition it to match the experimental data as well as possible. An argument is constructed in Ref. 12 that model conditioning is a good strategy for likely reduction of risk in extrapolative prediction; the conditioned model will certainly be more accurate in at least a local neighborhood of extrapolative prediction, and the advantage may extend to larger extrapolations as well.

Thus, if the model is found inadequate initially, model conditioning can add value to it before going forward to other predictions. There is no single model-conditioning approach that works best in all circumstances. Approaches in at least the following two categories exist. A combination of these approaches can also be used.

- Approach 1 – Model Calibration
  - Manipulate suitable model parameters to correct the model output results to match the experimental data. Here, this is generically called *model calibration*. Other terms are also used in engineering literature and practice.
- Approach 2 – Output Adjustment Function
  - A corrective value or function (with or without uncertainty) can be applied to the prediction results from the unaltered model. (This is called "add factoring" in the work by Sterman [22].) For example, a prescribed amount may be added to or subtracted from the predicted result; or the added or subtracted amount might be a function of the response magnitude and/or boundary condition inputs of the problem [23]; or the correction might be a multiplicative scaling of the predicted result by a constant percentage or by a scaling function that varies according to the response magnitude and/or boundary conditions. An example of the latter in Ref. 12 gives a self-adjusting correction that is reasoned to be somewhat robust in extrapolation over the range of circumstances for the particular applications the model was to be used for. A calibration (Approach 1) was not feasible, given resource constraints. It would have been ineffective anyway because the calibratable traveling parameters of the model did not have the sensitivity needed to adequately address the large experimental uncertainty in the boundary condition—which drove the validation uncertainty.

Model conditioning subsequent to a model validation finding of inadequate agreement between model predictions and experimental results may involve just Approach 1 (an initial calibration or a recalibration of the model based on the experiments/results from the validation activity), or just Approach 2, or Approach 1 followed by Approach 2.

The result of model conditioning is an adjusted prediction model and its associated uncertainty. How to best use a model's validation- or calibration- characterized prediction bias and uncertainty to potentially adjust or bias-correct the model to mitigate prediction risk beyond the validation or calibration conditions is a very difficult question and an active area of research. See Refs. 12-15, 19, and 23-30 for extended discussions and methodology proposals and demonstrations.

3. Uncertainty "Roll Up"

Figures 2–4 describe and distinguish between uncertainty propagation, aggregation, and roll up. These are discussed next, along with extrapolation.

The caption in Figure 2 defines uncertainty *propagation* and shows a general case where continuous interval and probabilistic uncertainties (PDFs) are propagated along with a discretely represented uncertainty. If continuous and/or discrete *non-probabilistic* uncertainty is propagated along with continuous and/or discretely represented *probabilistic* uncertainty, then various potential PDFs of response will exist, as illustrated at right in Figure 2. Representation and propagation of uncertainty in this circumstance can often be handled pragmatically with Level 1 approximate probability boxes as discussed earlier and demonstrated in Refs. 19, 31, 32, and 34.

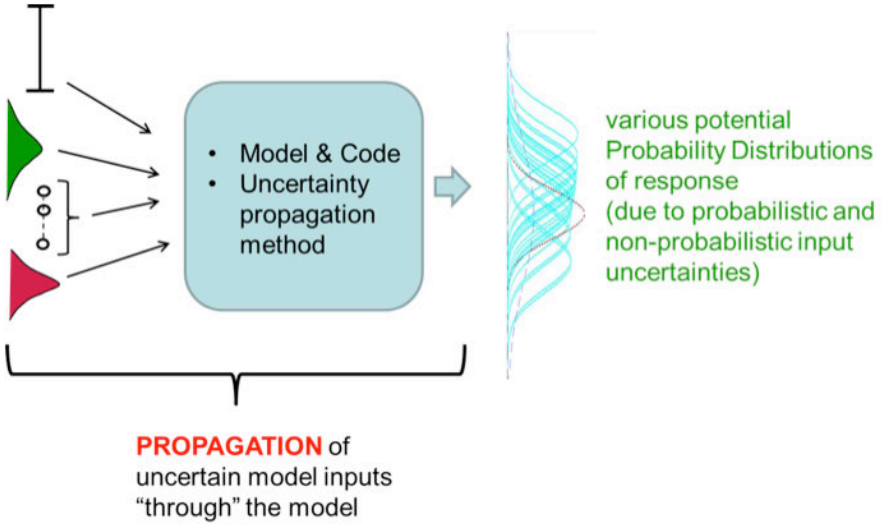


Figure 2. Uncertainty Propagation (depicted here) is the transformation of input uncertainties of a model into corresponding uncertainty in model output results by use of an uncertainty propagation method.

The caption in Figure 3 defines uncertainty *aggregation* and shows an example with several different sources and types of uncertainty being aggregated. Procedures for uncertainty aggregation are discussed and illustrated in Refs. 13, 16, 19, 31, 32, and 34.

Extrapolation occurs when, for example, model predictions are made at points in the modeling space away from model calibration or validation points where prediction error has been quantified, or when experimental bias corrections and uncertainty characterizations are extrapolated from where quantified originally. Such extrapolations are ubiquitous in engineering practice. Extrapolation error can be very important—even dominant—in engineering analysis. Even so, it is rarely estimated explicitly. It usually cannot be quantified rigorously, but it might be boundable in some situations. Estimation of extrapolation error and its uncertainty is usually a highly subjective matter of engineering experience and judgment—for example, manifesting as safety factors in design practice. Currently there are no well established approaches for estimating extrapolation error—this is presently an active area of research that includes: developing formal methods for extrapolation; making bias corrections and uncertainty characterizations more robust in extrapolation; and estimating extrapolation error. Examples of current research include the work done in References 12-15, 19, and 23-30.

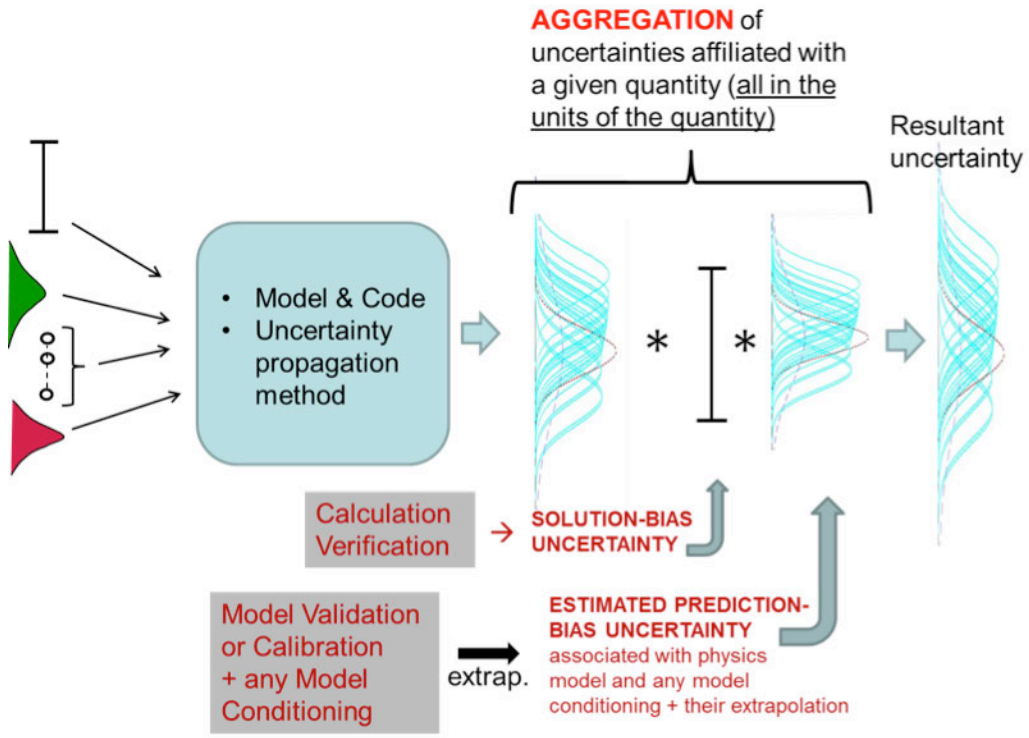


Figure 3. Uncertainty aggregation (exemplified here) is the combining of various uncertain aspects of a given quantity. All the uncertainties associated with the quantity, such as solution-bias uncertainty, prediction-bias uncertainty, and propagated uncertainties, have the same physical units as the quantity itself.

Uncertainty roll-up (Figure 4) consists of all uncertainty propagations, aggregations, and other uncertainty projection procedures used in quantifying the cumulative uncertainty associated with experimental or simulation results. An example of an uncertainty projection procedure besides propagation or aggregation is the extrapolation of a correction function uncertainty from a validation or calibration setting.

4. Margin Analysis

Estimation of the margin between system response and important performance, safety, or failure thresholds is often sought in engineering design and safety analyses. Margin analysis is frequently accompanied by significant uncertainties in model predictions of system response. Significant uncertainty may also be present in critical threshold levels for system degradation or failure. Margin analysis can also have bearing on model acceptability and/or the need for model refinement, depending on the magnitude of prediction uncertainty relative to the magnitude of the predicted margin. An example of margin analysis involving uncertainty in system responses and failure thresholds for multi-component systems with competing-mode failure races is presented in Ref. 35.

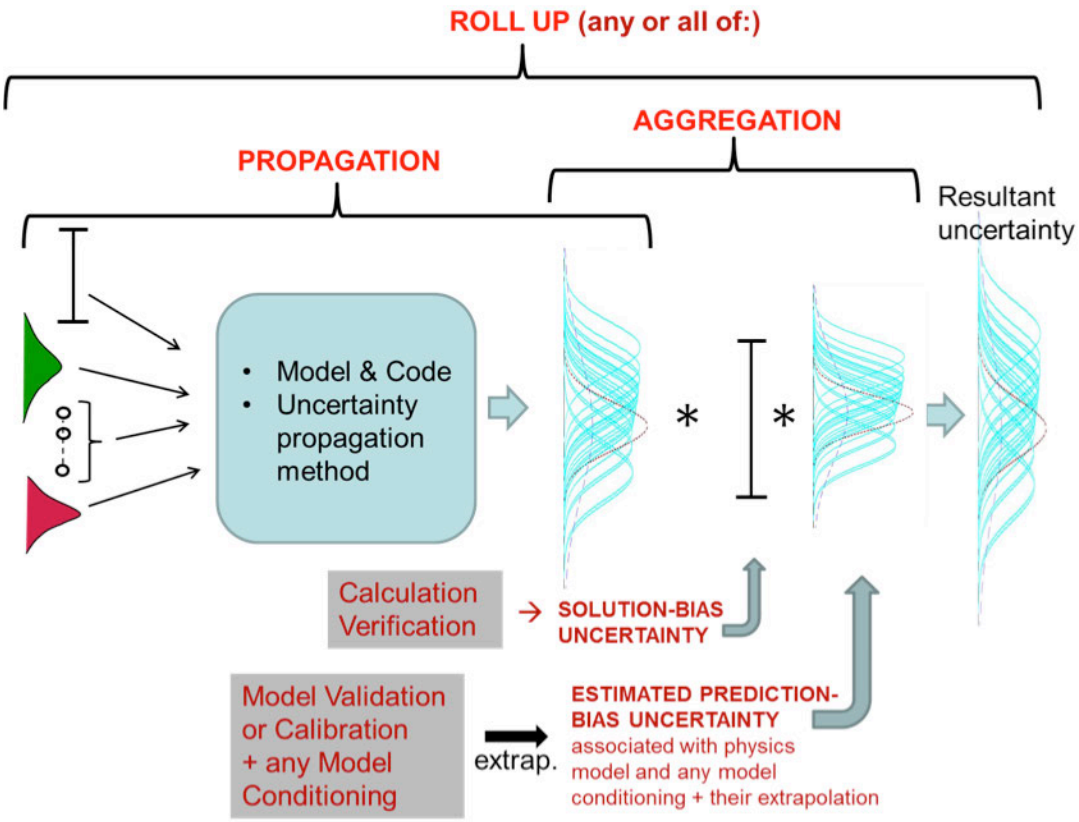


Figure 4. Example of uncertainty roll-up.

5. Summary

The material in this chapter provides a brief introduction and exposure to many types and sources of uncertainty in scientific and engineering experiments, models, simulations, and analysis. The ubiquitous presence of uncertainty in these primary instruments of science and engineering motivates the importance of uncertainty and sensitivity quantification and analysis. Prediction Science is still a relatively young discipline and requires far more development of effective and robust methodologies for integrating experimental results and their uncertainties with modeling and simulation, including optimized experiment design and analysis, parameter estimation and model calibration, verification, validation, and extrapolation. Progress in prediction will depend on advancing these methodologies as well as physics models and numerical algorithms, computers, and computing.

References

1. National Aeronautics and Space Administration, *NASA General Safety Program Requirements*, NPR 8715.3C (updated version with Change 9 dated 2/08/13).
2. National Aeronautics and Space Administration, *Standard for Models and Simulations*, NASA-STD-7009, 2008.

3. American Society of Mechanical Engineers, *Guide for Verification and Validation in Computational Solid Mechanics*, American Society of Mechanical Engineers (ASME) V&V 10-2006, 2006.

4. National Research Council of the National Academies, *Assessing the Reliability of Complex Models: The Mathematical and Statistical Foundations of Verification, Validation, and Uncertainty Quantification*, Appendix A – Glossary, National Academic Press, 2012.

5. Ferson, S., and Troy, W., "Sensitivity in Risk Analyses with Uncertain Numbers," Sandia National Laboratories report SAND2006-2801, June 2006.

6. Drignei, D., Mourelatos, Z. P., Kokkolaras, M., and Pandey, V., "Reallocation of Testing Resources in Validating Optimal Designs Using Local Domains," *Structural and Multidisciplinary Optimization,* Vol. 50, No. 5, pp. 825-838, 2014.

7. Park, C. Y., Kim, N. H., and Haftka, R. T., "How coupon and element tests reduce conservativeness in element failure prediction," *Reliability Engineering and System Safety,* Vol. 123, pp. 123–136, 2014.

8. Quintana, C., Millwater, H., Singh, G., and Golden, P., "Optimal Allocation of Testing Resources for Statistical Simulations," *Engineering Optimization,* Vol. 47, Issue 7, 2014.

9. Helton, J. C., Johnson, J. D., Oberkampf, W. L., and Sallaberry, C. J., "Representation of Analysis Results involving Aleatory and Epistemic Uncertainty," Sandia National Laboratories report SAND2008-4379, Sept. 2008.

10. Romero, V., Swiler, L., Urbina, A., and Mullins J., "A Comparison of Methods for Representing Sparsely Sampled Random Quantities," Sandia National Laboratories report SAND2013-4561, Sept. 2013.

11. International Organization for Standardization (ISO), *Guide to the Expression of Uncertainty in Measurement* (corrected and reprinted, 1995), ISO, Geneva, Switzerland, 1995.

12. Romero, V. J., "Validated Model? Not So Fast. The Need for Model 'Conditioning' as an Essential Addendum to Model Validation," 9th AIAA Non-Deterministic Methods Conference, Paper AIAA-2007-1953, April 2007.

13. Romero, V. J., "Type X and Y Errors and Data & Model Conditioning for Systematic Uncertainty in Model Calibration, Validation, and Extrapolation," SAE 2008 World Congress*,* SAE Paper 2008-01-1368, April 2008.

14. Romero, V. J., "Comparison of Several Model Validation Conceptions against a 'Real Space' End-to-End Approach," *Society Automotive Engineers International Journal of Materials and Manufacturing*, June 2011.

15. Romero, V. J., "Elements of a Pragmatic Approach for dealing with Bias and Uncertainty in Experiments through Predictions: Data and Model Conditioning; 'Real Space' Model Validation and Conditioning; Hierarchical Modeling and Extrapolative Prediction," Sandia National Laboratories report SAND2011-7342, Nov. 2011.

16. Romero, V. J., Luketa, A., and Sherman, M., "Application of a Versatile 'Real Space' Validation Methodology to a Fire Model," *AIAA Journal of Thermophysics and Heat Transfer*, Vol. 24, No. 4, pp. 730-744, Oct.-Dec. 2010.

17. Kambour, K., Hembree, C., and Keiter, E., "Calibration of Xyce Neutron Models for the QASPR Complex Prototype Circuit," Sandia National Laboratories report SAND2010-7988, Dec. 2010.

18. Romero, V., Dempsey, J. F., Schroeder, B., Lewis, J., Orient, G., Breivik, N., Antoun, B., and Winokur, J., "Assessment of a UQ Approach for Dealing with Sparse Samples of Discrete Random Functions (Material Stress-Strain Curves)," Sandia National Laboratories document SAND2016-3343C; presented at the SAE 2016 World Congress and Exhibition, April 12-14, 2016.

19. Romero, V., Rutherford, B., and Castro, J. "Model Calibration, Extrapolative Prediction, and Validation with Segregated Aleatory and Epistemic Uncertainties—Demonstration on Prototype Silicon Device," Sandia National Laboratories report (to be published).

20. Williamson, R. C., and Downs, T., "Probabilistic Arithmetic I: numerical methods for calculating convolutions and dependency bounds," *International Journal of Approximate Reasoning,* Vol. 4, pp. 89–158, 1990.

21. Ferson, S., *RAMAS Risk Calc 4.0 Software: Risk Assessment with Uncertain Numbers,* Lewis Publishers, Boca Raton, FL, 2002.

22. Sterman, J. D., Letter to the Editor, in *Science* Vol. 264, April 1994, re: "Verification, Validation, and Confirmation of Numerical Models in the Earth Sciences," by Oreskes et al., *Science,* Vol. 263, pp. 641–646 Feb. 1994.

23. Chen, X., Zhanpeng, S., Qinshu, H., and Liu, X., "Prediction Considering Multi-Model and Model Form Uncertainty in the Parameter Space," paper 15IDM-0015, SAE 2015 World Congress, April 21–23, 2015.

24. Chen, W., Xiong, Y., Tsui, K.-L., and Wang S., "A Design-Driven Validation Approach using Bayesian Prediction Models," *ASME Journal of Mechanical Design*, Vol. 130, No. 2, 2008.

25. Roy, C. J., and Oberkampf, W. L., "A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing," *Computer Methods in Applied Mechanics and Engineering*, Vol. 200, 2011.

26. Sankararaman, S., and Mahadevan, S., "Comprehensive Framework for Integration of Calibration, Verification, and Validation," 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, April 23–26, 2012.

27. Zhan, Z., Fu, Y., and Yang, R.-J., "On Stochastic Model Interpolation and Extrapolation Methods for Vehicle Design," *SAE International Journal of Materials and Manufacturing,* Vol. 6, No. 3, 2013.
doi:10.4271/2013-01-1386.

28. Xi, Z., Fu, Y., and Yang, R.-J., "An Ensemble Approach for Model Bias Prediction," *SAE International Journal of Materials and Manufacturing,* Vol. 6, No. 3, 2013.
doi:10.4271/2013-01-1387.

29. Hills, R. G., "Roll-up of Validation Results to a Target Application," Sandia National Laboratories report SAND2013-7424, Sept. 2013.

30. Oliver, T., Terejanu, G., Simmons, C., and Moser, R., "Validating Predictions of Unobserved Quantities," *Computer Methods in Applied Mechanics and Engineering*, Vol. 283, pp. 1320-1335, Jan. 2015.

31. Romero, V., Black, A., Breivik, N., Orient, G., Suo-Anttila J., Antoun, B., and Dodd A., "Advanced UQ and V&V Procedures applied to Thermal-Mechanical Response and Weld Failure in Heated Pressurizing Canisters," Sandia National Laboratories report SAND2015-3005C, presented at SAE 2015 World Congress, April 21-23, 2015.

32. Romero, V., Dempsey, J. F., and Antoun, B., "Application of UQ and V&V to Experiments and Simulations of Heated Pipes Pressurized to Failure," *Simulation Credibility—Advances in Verification, Validation, and Uncertainty Quantification*, U. Mehta, ed., D. Eklund, V. Romero, J. Pearce, and N. Keim, NASA/TP-2016-219422 and JANNAF/GL-2016-0001, 2016.

33. Jamison, R., Romero, V., Stavig, M., Buchhiet, T., and Newton, C., "Experimental Data Uncertainty, Calibration, and Validation of a Viscoelastic Potential Energy Clock Model for Inorganic Sealing Glasses," 2016 ASME Verification and Validation Symposium, May 2016.

34. Romero, V., Approximate Probability Boxes and Other Shortcuts in a Broad-before-Deep approach to Balanced UQ," 2015 ASME Verification and Validation Symposium, May 2015.

35. Romero, V., "Issues and Needs in Quantification of Margins and Uncertainty (QMU) for Phenomenologically Complex Coupled Systems," 8th AIAA Non-Deterministic Analysis Conference*,* AIAA Paper 2006-1989, May 2006.

A Survey of Probabilistic Uncertainty Propagation and Sensitivity Analysis Methods for Computational Applications

Laura P. Swiler, Vicente J. Romero

Sandia National Laboratories, [1] *Albuquerque, NM 87185*

Abstract

This chapter focuses on the following probabilistic uncertainty quantification (UQ) technologies for computational applications: sampling design, sensitivity analysis, model response approximation, and uncertainty propagation. The presentation is at an introductory and illustrative level. Our purpose is to provide a guide to help you get started, with suitable references that you can consult for more details. Section 1 describes some sources and types of uncertainty that the methods in this chapter address. Section 2 further defines and contextualizes these methods. Section 3 outlines sampling methods; Section 4 discusses sensitivity analysis; Section 5 provides an overview of model response approximation methods; and Section 6 discusses optimization-based reliability methods. Section 7 provides results of many of the methods applied to a simple example problem, and Section 8 provides a brief summary.

1. Sources and Types of Uncertainty in Computational Applications

There are many sources of uncertainty that are of interest to users, developers, and computer model analysts:

- Solution error uncertainty (associated with spatial and temporal discretization and incomplete convergence of the discretized equations being solved)
- Uncertainties in boundary or initial conditions
- Uncertainties in model input parameters
- Model form uncertainties (e.g., are we using the right model? Does the model incorporate the physics we need?)

This chapter addresses uncertainties in model input parameters where the uncertainties are continuous random variables described by probability distributions or probability density functions (PDFs). Uncertainties can be characterized into one of two types: aleatory or epistemic.

[1] Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Aleatory uncertainty characterizes the inherent randomness or variability of a quantity in a set or population of units, events, etc. For example, aleatory uncertainty can result from the possibility of random component failures or a distribution of possible material properties in the systems being analyzed. Aleatory uncertainty is irreducible except through design modifications, tighter tolerances on material quality controls, etc. Aleatory uncertainties are almost always characterized by probability distributions, and will result in probability distributions for affected response quantities when propagated through a model.

Epistemic uncertainty characterizes the lack of knowledge about the appropriate value to use for a quantity that is assumed to have a fixed value in the context of a specific application. (Other terms for epistemic uncertainty include state-of-knowledge uncertainty, subjective uncertainty, and reducible uncertainty.) Epistemic uncertainties can be reduced through increased understanding (research), increased data, or more relevant data. Epistemic uncertainties are often characterized by degrees of belief and are modeled using a variety of approaches: intervals, Dempster-Shafer belief structures, fuzzy sets, probability distributions, etc. A 2009 report by Helton [1] provides a comprehensive set of measures and approaches used to define aleatory vs. epistemic uncertainties for the purpose of margin analysis.

A Department of Energy document [2] that provides guidelines for quantifying margins and uncertainties for engineering design, performance, and safety assessment using modeling and simulation advises: “Where it is practical, calculation input characterizations should separate aleatory and epistemic uncertainties.” Separating aleatory vs. epistemic uncertainties is not a trivial task and it can be challenging to identify each type. With both types of uncertainty, care must be taken to elicit characterizations of uncertainty from experts and from data. There are a number of approaches that elicit uncertainty information from expert judgment, such as those described by Meyer and Booker [3] and Morgan and Henrion [4]. Typically, expert elicitation focuses on characterizing probability distributions on inputs. For example, if we have a number of model parameters (M), we would like to obtain a joint probability density function (JPDF) over the M parameters. The JPDF defines the relative likelihood that the M parameters will take a particular set of values. In practice, the JPDF is not always specified. Instead, one identifies the marginal distributions on the individual parameters and also a correlation matrix specifying the correlation between parameters.

This chapter focuses on methods for propagating aleatory and epistemic probabilistic uncertainties from simulation input parameters to output.

2. Uncertainty Propagation and Sensitivity Analysis for Probabilistic Uncertainty in Computational Applications

Simulation models for engineering applications are often developed and used to help assess a design or performance requirement. Accordingly, the capability to quantify the impact of aleatory variability and epistemic uncertainty in the decision context is critical. Thus, it is necessary to characterize uncertainties in simulation input parameters and propagate these uncertainties to outputs. This process results in a characterization of the simulation output uncertainties, usually in the form of histograms of outputs or probability density functions and cumulative density functions (CDFs). The results can be analyzed to determine characteristics of the response, such as mean and variance, how skewed the response distribution is, and so on. Output distributions can also be used to assess predicted performance against a system requirement. Such requirements usually involve “tail probabilities” of the output distribution and are often stated as

"the probability must be less than X that a system response quantity exceeds a threshold value of Y," where X is a small probability such as 0.01 to 10^{-6}.

The research on uncertainty quantification methods and approaches is quite extensive. For the purposes of this chapter we focus on the following key aspects. We assume the analyst has already performed some data analysis and/or expert elicitation to construct probability distributions on the input parameters and a joint probability density function.

1. **Sampling design**: The design identifies how many computer runs will be conducted and what input parameter values will be associated to the JPDF for those runs. Again, this chapter is confined to probabilistic UQ and sampling that supports this.
2. **Sensitivity analysis**: As part of the UQ process, we want to be able to identify important parameters that most contribute to the output uncertainty. We will discuss a few sensitivity analysis methods and how they relate to sampling design.
3. **Model response approximation**: The computational cost of engineering simulation models is often quite expensive; high-fidelity finite-element simulations may run from hours to weeks on hundreds to thousands of processors. Thus, understanding how the uncertainty propagation methods work and their relative advantages and costs is very important. Additionally, response surface approximations (RSAs) are often used as inexpensive replacements—meta-models or surrogates—for computationally expensive computer simulations. RSAs can be used in a variety of contexts, including uncertainty propagation and sensitivity analysis.
4. **Uncertainty propagation**. Often, this is done by evaluating the computational model at the values of the inputs (samples) prescribed by a global sampling design. However, it may also be performed using reliability methods, which determine sample locations dynamically in an adaptive optimization procedure that is described later.

We will provide some examples of the above four technologies using a simple engineering example involving a cantilever beam.

3. Global Sampling Approaches

Global sampling methods sample the uncertainty parameter space according to various different approaches, designs, and objectives. In contrast, adaptive methods progressively narrow the sampling focus to localized regions of the space as the sampling progresses and information is gained about the response behavior or function. This section describes two types of global sampling methods commonly used in Design and Analysis of Computer Experiments (DACE): random sampling and structured sampling for quadrature methods.

Many other types of global sampling designs are also available [5-7]. For example, classical experimental designs like central composite designs and orthogonal arrays are good at quantifying main effects and low-order factor interactions, which identify significant inputs for relatively low numbers of samples (<50) if the number of uncertain factors or inputs is 5 or fewer. A class of methods called quasi-Monte Carlo (QMC) generates sequences that are deterministic but are designed to fill in an N-dimensional uncertainty space, or hypercube, with a good approximation of a uniform volumetric distribution. (Hypercube refers to the bounded box in M dimensions—for M input parameters—in which the samples will be contained.) Many UQ problems can be mapped into uniform distributions with bounds of *(0,1)* and so the hypercube becomes a unit hypercube with all edge lengths equal to 1. Many of the QMC methods are based

on sequences linked to a prime number [8, 9]. Their high volumetric uniformity of sampling would appear to make them good sampling algorithms for surrogate model approximation methods that do not require particular sampling schemes such as those described in Section 3.2. Empirical results in some studies [10-12] support the suggestion that better sampling uniformity over the parameter space correlates with better response surface performance, although another study [13] did not find a strong correlation.

Finally, there is a set of optimal designs that are optimized according to specific criteria [5]. These include maximin designs, which maximize the minimum distance between points in the hypercube, and designs that optimize a particular information criterion. One of these is a D-optimal design, which seeks to maximize the determinant of the information matrix of the design, which results in maximizing the Shannon information of the parameter estimates.

We note that sampling is easy to parallelize for expensive computational physics models. For example, to evaluate the computational model at 100 sample points, one may request an allocation of 64 nodes on a parallel computer. If each job runs on 16 nodes, 4 jobs can run concurrently, so the effective throughput will be four times faster than running all 100 jobs serially. A software tool that can generate sample designs and run them in a variety of parallel computing configurations is DAKOTA [14].

3.1. Random Sampling

The most common method of incorporating uncertainty into simulations is to assume particular distributions on the uncertain input values, then randomly sample from those distributions, and run the model with the sampled values to build up a distribution of the output values. This is classic statistical Monte Carlo (MC) propagation of uncertainty. The output values can be analyzed to determine characteristics of the response (e.g., what are the maximum and minimum response values, what is the mean and the variance of the response, how skewed is the response distribution, etc.) The main advantage of MC sampling is that the accuracy and computational burden are essentially independent of the number of uncertain parameters, as opposed to many of the other approaches presented in this chapter.

Figure 1 shows the concept of MC sampling, where 3 samples are taken from distributions on inputs (in this case, 3 input variables). A green triangle, a yellow diamond, and a blue circle denote the three samples. Each sample is run through the simulation model, resulting in 3 realizations on the 2 outputs. Note that a sample taken at the lower end of an input distribution does not imply that the resulting simulation evaluated at that sample value would be at the lower end of the output sample population, unless the inputs and outputs are highly positively correlated. For example, although the lowest value from the first input distribution X_1 is a green triangle, this sample results in the highest value of the first output Y_1.

Random sampling (particularly Latin hypercube random sampling, which is explained below) can also be used in a space-filling mode to put samples in a unit hypercube with reasonable volumetric uniformity. This is done by initially prescribing uniform PDFs for each uncertainty dimension in the UQ problem, and then randomly sampling the uniform PDFs. This provides a reasonable set of samples for response surface approximation methods that do not require particular sampling layouts but benefit from high volumetric uniformity of samples over the hypercube.

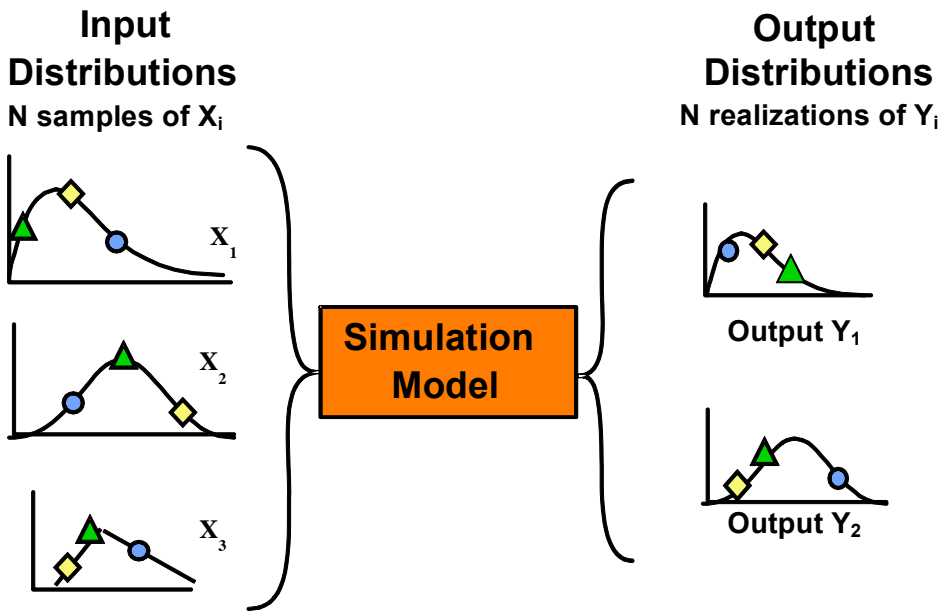


Figure 1. Monte Carlo random sampling approach to propagate uncertainties.

When the values of the random sampling are not used directly to provide statistical information on response, but are instead used to provide a basis for building response surfaces, this is a non-statistical use of the random samples. Once the response surface is constructed, it can be employed in various UQ propagation approaches. Perhaps the most common approach is to exhaustively sample the response surface with statistical MC sampling using the actual PDFs of the UQ problem. Exhaustive MC sampling can be performed because the response surface is very fast and inexpensive to sample millions of times. If the application model itself is cheap to evaluate, then response surfaces are not necessary and the MC sampling can be performed directly on the model.

Next we describe how to generate random samples of the input PDFs. The goal of random sampling methods is to provide a good probabilistically weighted coverage of the uncertainty space, as illustrated in Figure 2.

Pure random sampling can be implemented using a large variety of tools, algorithms, and software, such as Excel, Minitab, the "R" statistical language, MATLAB, etc. These methods simply generate samples from the particular distribution specified using a random number generator. There is no structure to the samples: one simply picks a random draw each time, and the particular sample realization does not know where it falls in relation to samples that come before or after it. Random sampling has the following advantages:

- Easy to implement
- Easy to explain
- Preferred when a sufficiently large number of samples are affordable
- Produces unbiased estimates for means, variances and percentiles

However, to get accurate estimates for the statistics obtained directly from the random samples, a large number of samples is required. The accuracy of the mean estimate obtained from a set of

random samples displays $1/\sqrt{N}$ convergence, meaning that on average one needs to quadruple the number of sample points to halve the error. In order to resolve small probabilities regarding predicted responses, hundreds of thousands (or more) sample points may be required to obtain a desired accuracy with random sampling. Some rules of thumb are offered at the end of this subsection.

Since computational simulations are often expensive, it is often not feasible to use pure random sampling to generate sufficiently large sample sizes for adequately resolved statistics. Therefore, other methods have been developed. A good alternative to pure random sampling is Latin hypercube sampling (LHS) [15-17]. LHS is a stratified random sampling method where the distribution is divided into strata or bins. Each stratum is chosen to be equally probable, so that the strata are of equal length for uniform distributions but of unequal length for normal distributions. For example, the strata near the center of normal distributions are shorter than the strata near the tails (see Figure 2). To create a total sample of size N using LHS, an individual sample value is chosen from each of the N equally probable strata for each input variable. This stratification approach serves to force a better sampling across the entirety of each distribution and eliminates some of the clustering of sample points often seen in pure random sampling.

For multi-dimensional sampling, LHS also serves to achieve a good mix of sample values from different input variables. For example, one would not want the sample in strata 1 from input A to be paired with the sample in strata 1 from input B. Instead, the pairing of the strata is performed in such a way to generate multi-dimensional samples that are well-mixed or randomized. Pairing algorithms have been designed to achieve this [18]. Importantly, these pairing algorithms allow the generation of independent inputs but in the case where some or all of the inputs are correlated with one another, the pairing algorithms also allow the generation of samples that honor a user-specified correlation structure. For example, one might want input C to have a correlation of 0.8 with input D. Many random sample generators do not allow a correlation structure amongst samples. In real problems, however, inputs often are correlated and thus the capability to handle correlations is important.

Finally, LHS is more efficient than pure Monte Carlo in the sense that it requires fewer samples to achieve the same accuracy in statistics (standard error of the computed mean, for example). LHS gives an estimator for a function mean that has lower variance than MC for any function having finite second moment [19, 20]. Further, the convergence behavior of LHS improves if the function is additively separable, meaning it can be decomposed into additive functions of the individual input parameters. For resolving small tail probabilities (e.g., percentiles < 0.001), LHS may lose its efficiency advantage over pure random sampling [21] but it will not perform worse than pure MC.

Figure 2 shows plots for four cases: pure random sampling and LHS with uncorrelated and correlated inputs. The plots show the samples for two random variables: a uniform and a normal. Grids divide the plots into ten equi-probable bins for each variable. Pure random sampling, especially with a small number of samples, has less coverage of the space than LHS. For example, in the upper left plot showing simple random sampling, there are four samples of the uniform variable with values between 0.9 and 1, but zero samples of the uniform variable with values between 0.61 and 0.92. In the upper plots, the uniform and normal are treated as uncorrelated, but in the lower plots, they are sampled with a correlation of 0.7 (a reasonably strong positive correlation).

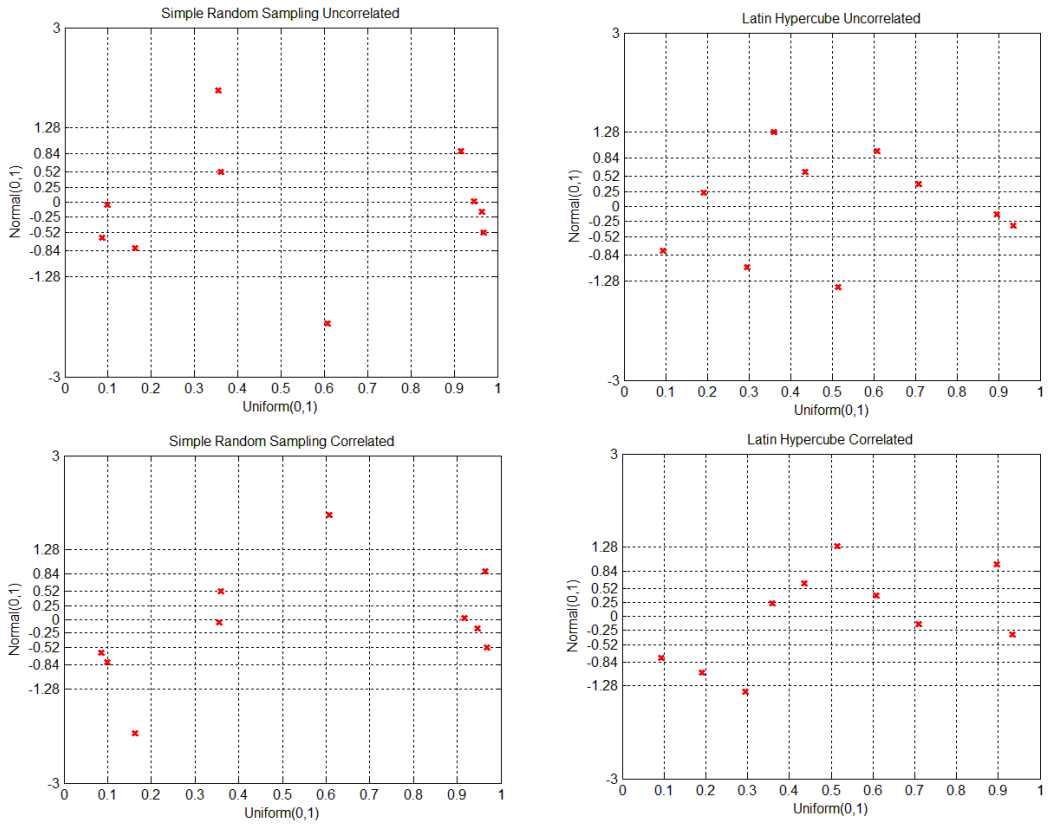


Figure 2. Pure random sampling (left) and LHS (right), with uncorrelated (top) and correlated (bottom) inputs.

A common question when using statistical MC samples is "How many samples do I need to take?" The answer depends on the output statistics one is trying to resolve. For example, often 20–30 samples are sufficient to obtain a reasonable estimate of the mean. Formally, one can use statistical confidence intervals to help determine a sufficient number of samples based on the quantity of interest. A confidence interval is an interval estimate (a range) on a population parameter such as a mean or variance of a simulation response. The level of confidence of the interval indicates the probability that the confidence range captures the true population parameter given a distribution of samples. The confidence interval on the mean can be obtained with standard statistic and the use of a T-test [22].

A similar procedure may be used to estimate confidence intervals for the variance or for percentiles of the population. If one wants to obtain a reasonably accurate estimate of the variance, a few hundred samples may be necessary, but to estimate the 99th percentile of the output distribution, thousands of samples may be required. One rule of thumb, based on confidence interval considerations, is to take 10 times N, where one expects a 1 out of N failure probability [23]. When using samples to construct response surface models, another rule of thumb is to start with a number of samples that is at least ten times the number of uncertain variables. For example, if one has 15 uncertain variables, one should specify at least 150 samples.

3.2. Structured Sampling for Quadrature Methods

In this section, we describe a set of points that are used for a special type of surrogate called stochastic expansion methods, which are discussed in Section 5.2. These points are not random samples taken over the input uncertainty space. Instead, they are very structured point sets over this space. They are based on quadrature methods [24, 25]. The points are designed to calculate the stochastic expansion functional representation (also discussed in Section 5.2) very efficiently. However, they cannot be used directly to calculate statistics. That is, one cannot take a simple average over quadrature points to calculate the mean response. Instead, one needs to incorporate more information about the weights of the points in the quadrature scheme to calculate moments, as in point estimate methods [26], or use stochastic expansion methods.

Stochastic expansion methods have become popular the past ten years as ways to represent computer output. These methods are quite mathematical and their construction is beyond the scope of this report. The interested reader is encouraged to read books by Dongbin Xiu [27] and Ghanem and Spanos [28], or a number of papers in this area [29-31]. The purpose of this discussion is to simply give the reader a sense of the sample points used in these methods and a sense of the functional form that the stochastic expansions take.

In a single dimension, a set of quadrature points is used. In multiple dimensions, a straight extension is called a tensor product grid, because all of the points for one dimension must be evaluated with all combinations of points for the other dimensions. Figure 3 shows plots that are analogous to Figure 2, but use tensor product quadrature. The top left plot shows 3 quadrature points for each variable; the top right plot shows 4 quadrature points per variable; the bottom left plot shows 5 quadrature points per variable, and the bottom right plot shows the same 5 points per variable with a correlation of 0.7 between the variables. Note that the number of quadrature points that are taken controls the order of accuracy of the resulting stochastic expansion.

If the number of uncertain variables is larger than 4 or 5, sparse grids are preferable over tensor product grids because sparse grids use a drastically reduced number of evaluation points while maintaining a high level of accuracy [24, 25, 32]. Sparse grids use linear combinations of the tensor product rules, with the property that only products involving a small number of points are retained. An example of the reduction in the number of points for a tensor product grid compared with a sparse grid is shown in Figure 4.

Several variations of sparse grids exist depending on the growth rate used and whether the one-dimensional quadrature rules are nested. Anisotropic sparse grids can also be constructed using either a priori information regarding the significant dimensions, or using a posteriori error indicators [33].

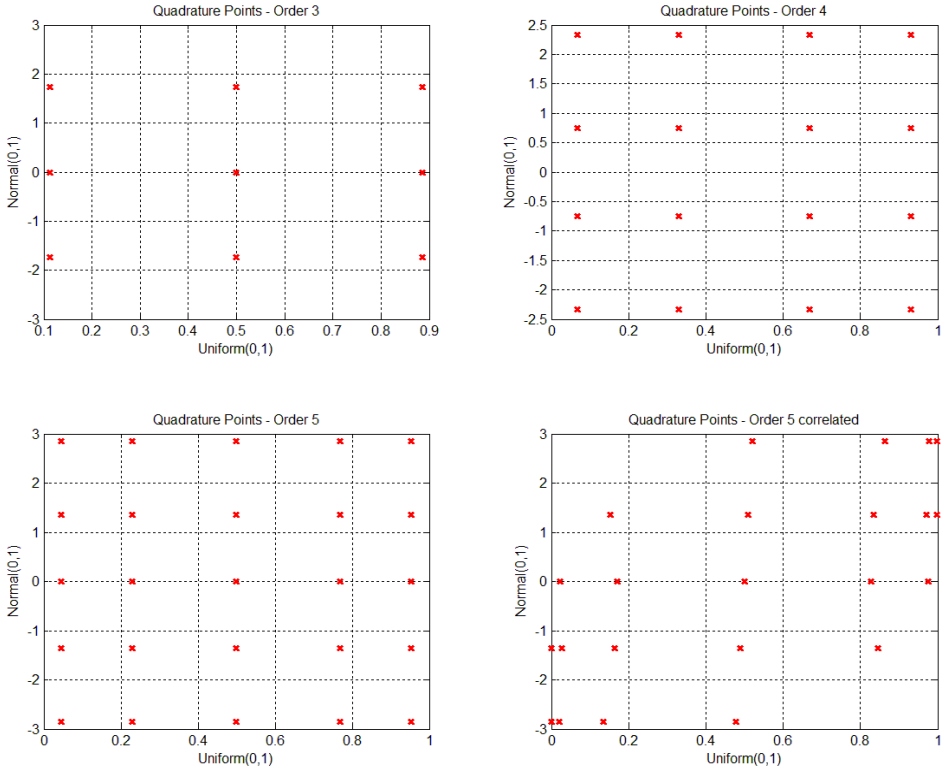


Figure 3. Tensor product quadrature example.

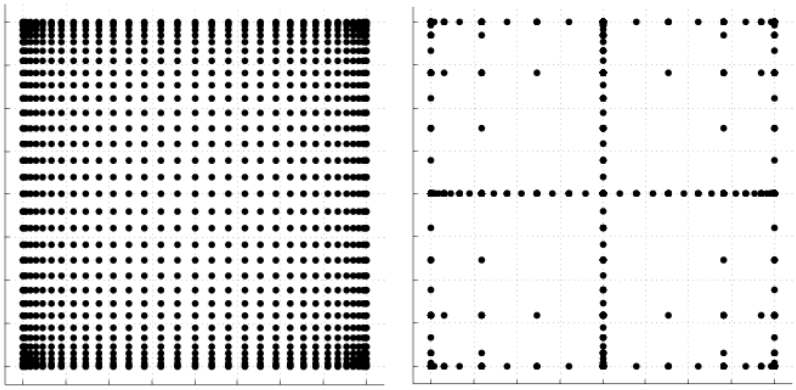


Figure 4. Comparison of a tensor product grid in 2D using Clenshaw-Curtis points (left) and a sparse grid (right).

4. Sensitivity Analysis

The purpose of sensitivity analysis is to identify the most significant factors or variables affecting the uncertainty of the model predictions. In this section, we summarize two approaches to performing sensitivity analysis: correlation analysis and variance-based decomposition. The reader is encouraged to refer to one of the books published by a group at the European Research

Commission on sensitivity analysis, such as *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models* [34].

4.1. Correlation Analysis

Correlation refers to a statistical relationship between two random variables or two sets of data. In analysis of computer experiments, where an ensemble of simulation runs have been performed according to some type of sampling design, we have a set of results. The convention is to have each sample (run) of the simulation written on a separate row of the ensemble matrix. For example, if N simulation runs were performed, with D inputs and P outputs, the resulting ensemble matrix would be of dimension N*(D+P). In this scenario, we could perform a correlation analysis on the entire matrix; however, often the correlations between the various inputs are not interesting—especially if the sample design has been constructed so that the inputs are independent, in which case the correlations between inputs would be near zero. Likewise, correlations between various outputs may not be interesting unless some of the outputs are very strongly correlated, and thus perhaps we could reduce the analysis by focusing only on a subset of outputs. The main focus of correlation analysis of computer experiments is the correlation between inputs and outputs.

There are several types of correlations that can be calculated: simple, rank, and partial. Simple correlation, which measures the strength and direction of a linear relationship between variables, refers to correlations performed on the actual input and output data, calculated by the Pearson correlation coefficient. For example, the Pearson correlation between input X and output Y is given by $\rho_{(X,Y)}$ [22]:

$$\rho_{(X,Y)} = \frac{\mathrm{cov}(X,Y)}{\sqrt{\mathrm{var}(X).\mathrm{var}(Y)}} = \frac{E[(X-\bar{x})(Y-\bar{y})]}{\sigma_X \sigma_Y} \approx \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}} \quad (1)$$

The Pearson correlation is +1 in the case of a perfect positive (increasing) linear relationship, -1 in the case of a perfect decreasing (negative) linear relationship, and some value between -1 and 1 in all other cases. A simple correlation near zero means there is not a linearly organized relationship between the variables. Figure 5 shows some example correlation patterns and corresponding correlation coefficients. Note that if two variables are independent, they will have zero correlation but the converse is not true: they may have zero or near-zero correlation but show a strongly organized nonlinear relationship (for example, see the last row of Figure 5). The best way to identify such zero-correlation but strongly patterned relationships is to plot them in a scatterplot as shown in Figure 5.

Rank correlation refers to correlations performed on the ranks of the data. Ranks are obtained by replacing the actual data by the ranked values, which are obtained by ordering the data in ascending order. For example, the smallest value in a set of input samples would be given a rank 1, the next smallest value a rank 2, etc. Rank correlations are useful when some of the inputs and outputs differ greatly in magnitude; it is easier to check whether the smallest ranked input sample is correlated with the smallest ranked output, for example. Rank correlations can also be used when monotonic nonlinear relationships exist. A rank correlation coefficient is also called a Spearman correlation coefficient.

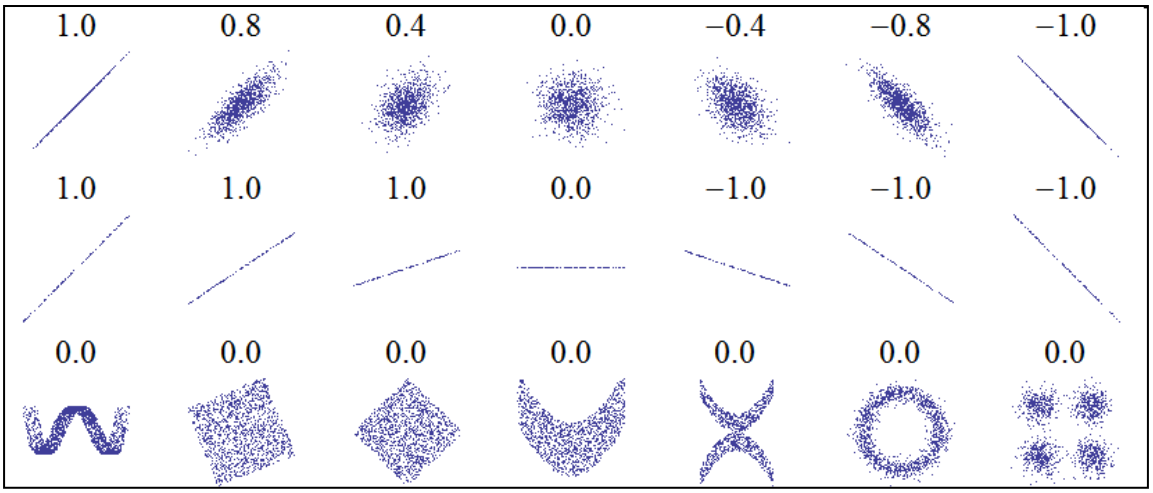


Figure 5. Example correlation relationships (Source: http://en.wikipedia.org/wiki/Correlation).

Partial correlation coefficients are similar to simple correlations, but a partial correlation coefficient between two variables measures their correlation while adjusting for the effects of the other variables. Partial correlation measures the strength and direction of a linear relationship between an input X_j and an output Y after the linear effects of the remaining input parameters have been removed from both X_j and Y. For example, if one has a problem with two highly correlated inputs and one output, the correlation of the second input and the output may be very low after accounting for the effect of the first input. Partial correlations are calculated in the following way: for a particular input, X_j, a regression model of the response Y is constructed over all of the inputs except X_j. The residuals (differences between the actual Y values and the regression model predictions) are constructed, indicating how well the other variables are able to predict the response Y. Then, a regression model of X_j is constructed as a function of the other inputs, and the residuals based on this regression are constructed. The partial correlation between Y and X_j is given by the correlation between these two sets of residuals. If Y is highly dependent on the other variables, and X_j is also highly dependent on the other variables, the residuals for each will be small and their correlation will be low, indicating that X_j is not highly correlated with Y when the effects of the other variables have been incorporated.

4.2. Variance-based Decomposition

The correlation coefficients described in Section 4.1 only detect a linear or monotonic relationship. In contrast, the variance-based indices (also referred to as Sobol´ indices) are not limited in this way. The variance-based indices identify the fraction of the variance in the output that can be attributed to an individual variable alone or with interaction effects [35, 36]. There are two classes of variance-based sensitivity indices: main effects and total effects. The main effects indices, $\mathbf{S}_i$, identify the fraction of uncertainty in the output Y attributed to input X_i alone. The total effects indices, $\mathbf{T}_i$, correspond to the fraction of the uncertainty in output Y attributed to X_i and its interactions with other variables. These sensitivity indices are represented as:

$$S_i = \frac{Var(E(Y \mid X_i))}{Var(Y)} \tag{2}$$

$$T_i = \frac{E[Var(Y \mid X_{-i})]}{Var(Y)} \tag{3}$$

where Var(·) is the variance, E(·) is the expected value, and $E(Y|X_i)$ is the expected value of Y conditioned on X_i. $Var(Y|X_{-i})$ is the variance of Y conditioned on all the inputs except X_i. These indices involve multi-dimensional integrals that, in practice, are evaluated approximately. Note that each S_i varies between 0 and 1. Values close to 1 mean that the uncertainty in variable X_i is very significant in contributing to the uncertainty in output Y. The sum of S_i over all variables *i* must equal 1. However, there are not the same restrictions on T_i. The values of T_i are greater than or equal to 0, but are not upper-bounded by 1 and their sum over all variables does not add to 1.

The team led by Andrea Saltelli at the European Research Commission is generally credited with popularizing the use of variance-based indices for sensitivity analysis. In the past 10–15 years, several approaches have been developed for calculating the Sobol' sensitivity indices. A recent paper by Saltelli et al. [37] provides a detailed comparison of sampling approaches, with some comments about the relationship between the estimators and the sampling methods used.

Ideally, a full factorial sample would be performed with *m* samples taken in each of *d* input dimensions. Then, the integrals in the Sobol' formulas can easily be calculated given the m^d samples. For example, when calculating the numerator in Eq. 2, we calculate the inner expectation term m times, each time averaging over the remaining m^{d-1} points in the other dimensions. We calculate: $E(Y|X_i = x_i)$ for each of the *m* points in dimension *i*, then take the variance of *m* expected values to obtain the numerator for the main effects indices. The total effects indices are calculated in a similar manner.

The full factorial approach requires m^d samples, which may not be practical when each sample is an evaluation of a computationally costly function. Saltelli et al. [34] developed an approach that uses fewer samples: *(2+d)m* samples. In Section 7, we discuss results obtained using a recent formulation [37] for the *(2+d)m* samples, which has been improved to remove bias and better capture interaction effects. The actual formulas are described in a report by Weirs et al. [38].

Finally, these sensitivity indices may be calculated when stochastic expansion methods such as polynomial chaos or stochastic collocation are used to propagate the uncertainty from inputs to outputs. When using stochastic expansion methods, the sensitivity indices S_i and T_i can be calculated as analytic functions of the coefficients of the expansion. This is an efficient property, since one does not have to take additional samples beyond those used to construct the expansion initially. Formulations of the sensitivity indices based on polynomial chaos are derived by Sudret [39]; the sensitivity indices based on stochastic collocation are derived by Tang et al. [40]. We present the results of variance-based decomposition using polynomial chaos in Section 7.

5. Surrogate Models

It is often useful to construct a meta-model to use in uncertainty and sensitivity analysis of a computational physics model when it is computationally demanding. Once constructed, the meta-model is relatively inexpensive to evaluate so it is often used as a surrogate for the physics model in uncertainty propagation and sensitivity analysis that may require thousands to millions of function evaluations.

There are many different types of surrogate models, including neural networks, regression models, radial basis functions, splines, etc. One popular approach in the literature is to develop an emulator that is a stationary smooth Gaussian process [5, 6]. There are many good overview articles that compare various meta-model strategies. Various smoothing predictors and nonparametric regression approaches are compared in [41-43]. Simpson et al. provide an

excellent overview not just of various statistical meta-model methods but also approaches that use low-fidelity models as surrogates for high-fidelity models [12]. The paper also addresses the use of surrogates in design optimization, which is a popular research area for computationally expensive disciplines such as computational fluid dynamics in aeronautical engineering design. Hafka and his students developed an approach that uses ensembles of emulators or hybrid emulators [44].

For the purposes of this chapter, we emphasize Gaussian processes because they are able to model a wide range of functional forms. Also, based on a set of training points, Gaussian processes have the ability to predict not only a point value at a new point, but also a measure of uncertainty in that point value. This latter aspect can be very important when deciding how much to trust the surrogate. In Section 5.2, we also emphasize stochastic expansions that use very particular sample locations as build points (quadrature points), as discussed earlier in Section 3.2.

5.1. Gaussian Process Models

A Gaussian process is defined as follows: A stochastic process—a collection of random variables $\{Y(\mathbf{x}) \mid \mathbf{x} \in \mathbf{X}\}$ indexed by a set $\mathbf{X}$ (in most cases, $\mathbf{X}$ is $\Re^d$, where d is the number of inputs and $\Re$ is the space of real numbers)—is defined by giving the joint probability distribution for every finite subset of variables $Y(\mathbf{x}_1)$, ..$Y(\mathbf{x}_k)$. A Gaussian process (GP) is a stochastic process for which any finite set of Y-variables has a joint multivariate Gaussian distribution. A GP is fully specified by its mean function $\mu(\mathbf{x}) = E[Y(\mathbf{x})]$ and its covariance function $C(\mathbf{x}, \mathbf{x}')$. The basic steps for using a GP are:

1. Define the mean function. The mean function can be any type of function. Often the mean is taken to be zero or a constant, but this is not necessary. A common representation, for example, is a regression model $y(\mathbf{x}) = \sum_j \beta_j x_j$.
2. Define the covariance. There are many different types of covariance functions that can be used, such as squared exponential, Matern, cubic, etc. We focus on stationary covariance functions where $C(\mathbf{x}, \mathbf{x}')$ is a function of the distance $(\mathbf{x} - \mathbf{x}')$ and is invariant to shifts of the origin in the input space. A commonly used covariance function is:

$$C(\mathbf{x},\mathbf{x}') = \nu \exp\{-\sum_{u=1}^{d} \rho_u^2 (\mathbf{x}_u - \mathbf{x}'_u)^2\} \tag{4}$$

   This covariance function involves the product of *d* squared-exponential covariance functions with different length scales ρ on each dimension, and ν is the process variance. The form of this covariance function captures the idea that nearby inputs have highly correlated outputs.

3. Perform the prediction calculations. Given a set of *n* input data points $\{\mathbf{x}_1, \mathbf{x}_2, .. \mathbf{x}_n\}$ and a set of associated observed responses or targets $\{z_1, z_2, .. z_n\}$, we use the GP to predict the target z_{n+1} at a new input point $\mathbf{x}_{n+1}$. The target is usually represented as the sum of the "true" response, y, plus an error term: $z_i = y_i + \varepsilon_i$, where ε_i is a zero-mean Gaussian random variable with constant variance σ_ε^2. If C is the $n \times n$ covariance matrix with entries $C(\mathbf{x}_i, \mathbf{x}_j)$, then the prior distribution on the targets z_i is N(0,C). The distribution of the predicted term z_{n+1}, which is conditional on the data $\{z_1, z_2, .. z_n\}$, is Gaussian with the following mean and variance:

$$E[z_{n+1} \mid z_1, z_2, .. z_n] = k^T C^{-1} z \tag{5}$$

$$Var[z_{n+1} \mid z_1, \ldots, z_n] = C(\mathbf{x}_{n+1}, \mathbf{x}_{n+1}) - k^T C^{-1} k \tag{6}$$

where k is the vector of covariances between the n known targets and the new n+1 data point: $k = (C(\mathbf{x}_1, \mathbf{x}_{n+1}), \ldots. C(\mathbf{x}_n, \mathbf{x}_{n+1}))^T$, C is the $n \times n$ covariance matrix of the original data, and z is the $n \times 1$ vector of target values.

The equations for the mean and variance of the predictive distribution for z_{n+1} both require the inversion of C, an $n \times n$ matrix. In general, this is a $O(n^3)$ operation. Also, the covariance matrix may be near singular. Several approaches have been developed to deal both with the ill-conditioning and with large data sets (e.g., greater than 1,000 data points) [45].

Steps 1–3 above give the general framework for defining a Gaussian process and using it for prediction. However, the length scale parameters in the covariance matrix must be calculated to perform the prediction in Steps 2 and 3. There are two main approaches. One is to use maximum likelihood estimation, in which the likelihood function [45] is maximized. This results in point estimates of the covariance parameters. The other approach is to use Markov chain Monte Carlo (MCMC) sampling to generate posterior distributions on the hyperparameters that govern the covariance function (and the mean function) [46]. The assumption of zero-mean GPs is often made, so the Bayesian updating only involves hyperparameters governing the covariance function. Since these may be quite complicated, a MCMC sampling method is usually still needed to generate the posterior. Maximum likelihood methods are more commonly used to obtain the correlation lengths.

5.2. Stochastic Expansion Methods

We now discuss stochastic expansion methods briefly. These methods approximate the outputs of an uncertain system through series expansions of standard random variables.

Polynomial chaos expansion (PCE) is a stochastic expansion method whereby the output response is modeled as a function of the input random variables using a carefully chosen set of polynomials—typically, Hermite polynomials. The series expansion is typically based on unit-variance independent Gaussian random variables. So, with PDFs that can be non-normal and correlated, the uncertain variables **X** in the problem must be transformed to a space of independent zero-mean and unit-variance Gaussian random variables, **u** (i.e., standard normal variables). The Nataf and Rosenblatt transformations [47, 48] are used to transform the uncertain variables to standard normal variables. Then the expansion is formulated in terms of Hermite polynomials as functions of the Gaussian random variables. The goal of a PCE analysis is to determine the unknown coefficients of the Hermite polynomials in the expansion. For other distributions, the polynomials are usually chosen according to the Weiner-Askey scheme that provides an orthogonal basis with respect to the probability density function for the input random variables [49].

There are a variety of approaches for generating the sample points, but typically tensor product grids or sparse grids are used. Structured sampling schemes typically result in more accurate estimates of the coefficients of the polynomials than random sampling does, but this depends on the order of the polynomials needed to resolve the uncertainty in the response functions [29].

Orthogonal polynomials can be generated numerically for arbitrary PDFs, but this is beyond the scope of this chapter.

An important distinguishing feature of the PCE methodology is that the solution series expansions are expressed as random processes, not merely as statistics as in the case of many nondeterministic methodologies. PCE can be most efficient when there is some additional information from the physics of the problem about what type of approximating polynomials are most appropriate. One advantage of stochastic expansion methods is that the moments of the expansion (e.g., the mean or variance of the response) can be written analytically, along with analytic formulations of the derivatives of these moments with respect to the uncertain variables. This nice feature can be exploited in design optimization under uncertainty problems [50] or epistemic uncertainty problems [51]. A disadvantage of polynomial chaos is that the tensor product grids upon which they are often constructed do not scale well, especially as the number of variables and the polynomial order increases. Recent research in adaptive methods and sparse methods is addressing this limitation [52].

The propagation of input uncertainty through a model using PCE consists of the following steps: (1) input uncertainties are transformed in terms of a set of uncorrelated random variables; (2) a functional form such as Hermite polynomials is assumed; and (3) the parameters of the functional approximation are determined.

The general polynomial chaos expansion for a response R has the form:

$$R = a_0 B_0 + \sum_{i_1=1}^{\infty} a_{i_1} B_1(\xi_{i_1}) + \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{\infty} a_{i_1,i_2} B_2(\xi_{i_1}, \xi_{i_2}) + \dots \tag{7}$$

where each B term represents an orthogonal polynomial that is a function of a number of the uncertain inputs ξ_i. The number of random variables and the order of the expansion are unbounded prior to truncation. This expression is usually written in terms of the order-based indexing:

$$R = \sum_{j=0}^{\infty} \alpha_j \Psi_j(\xi). \tag{8}$$

where there is a one-to-one correspondence between $a_{i1,i2}$ in Eq. 7 and α_j in Eq. 8, and between $B_n(\xi_{i1}, \xi_{i2}, ..., \xi_{in})$ in Eq. 7 and $\psi_j(\xi)$ in Eq. 8. Each $\psi_j(\xi)$ is a multivariate polynomial that involves products of the one-dimensional polynomials. In practice, both the number of random variables and the order of the expansion are truncated yielding an expansion of the form:

$$R \approx \sum_{j=0}^{P} \alpha_j \Psi_j(\xi). \tag{9}$$

If a total-order polynomial is constructed (e.g., a total order of 2 would involve terms whose exponents are less than or equal to 2, such as x_1^2, x_2^2, and x_1x_2 but not $x_1^2x_2^2$), the total number of terms *N* in a polynomial chaos expansion of arbitrary order p for a response function involving n uncertain input variables is given by: $\frac{(n+p)!}{n!p!}$. If an isotropic tensor product expansion is used with order p in each dimension, $(p+1)^n$ sample points must be used. If the order *p* of the expansion captures the behavior of the true function and if the number of quadrature points taken is

sufficient to estimate the coefficient terms, polynomial chaos methods will give very accurate results for the output statistics of the response.

Similar to PCE, stochastic collocation methods construct a polynomial approximation of the output response. The key difference is that the stochastic collocation approximation is a multi-dimensional Lagrange based on a set of collocation points. These points may be based on tensor product grids or on the Smolyak sparse grids presented previously.

6. Optimization-Based Reliability Methods

Reliability methods provide an optimization-based alternative approach to uncertainty quantification that can be less computationally demanding than sampling techniques for certain types of UQ analysis. Reliability methods are based on adaptive searching and function approximation approaches that compute approximate response distribution statistics based on specified random variable distributions. These response statistics include response mean, standard deviation, and percentiles of cumulative or complementary cumulative distribution functions (CDF/CCDF). Reliability methods are often more efficient at computing statistics in the tails of the response distributions (events with low probability) than sampling-based approaches, as the number of samples required to resolve a low probability can be prohibitive. However, there are many challenges with their use: the approximation of the failure domain may be poor for small failure probabilities, the results are unreliable when the system is subject to multiple limit state functions, and the results depend heavily on convergence to a global minimum.

The methods strive to answer the fundamental question: "Given a set of uncertain input variables **X**, and a scalar response function *g*, what is the probability that the response is below or above a certain level z?" This can be written as P[$g(\mathbf{X}) \leq z$] = $F_{g(z)}$, where $F_{g(z)}$ is the cumulative distribution function (CDF) of the uncertain response $g(\mathbf{X})$ over a set of response levels. P[$g(\mathbf{X}) > z$] can be written as 1 - P[$g(\mathbf{X}) \leq z$] and defines the complementary cumulative distribution function (CCDF). Either probability calculation involves a multi-dimensional integral in **X** space over an irregularly shaped domain.

All reliability methods involve transforming the user-specified uncertain variables into standard normal variables. The joint probability density contours are circular in nature in standard normal space. Also, the multi-dimensional integrals can be parameterized by simple functions of a single parameter, β, called the reliability index. In a given local reliability problem, β has a value that is determined by searching/optimization as the minimum Euclidean distance from the origin in the transformed space to the failure surface. This point on the failure surface is also known as the most probable point (MPP) of failure. Figure 6 depicts an example of reliability optimization in two dimensions (u_1 and u_2 on the axes). The failure region is shaded; an example circular contour of the unit Gaussian variables is shown in pink. The minimum distance from the origin to the failure surface, β, is labeled, as is the MPP on the failure surface. The true failure surface is outlined in maroon; a linear approximation of the limit state function used in the first-order reliability method (FORM) is shown by the blue dashed line; the second-order approximation used in the second-order reliability method (SORM) is shown in red. Note that the reliability methodology is equally applicable for generic applications, not simply those corresponding to failure problems; this nomenclature is due to the origin of these methods within the disciplines of structural safety and reliability.

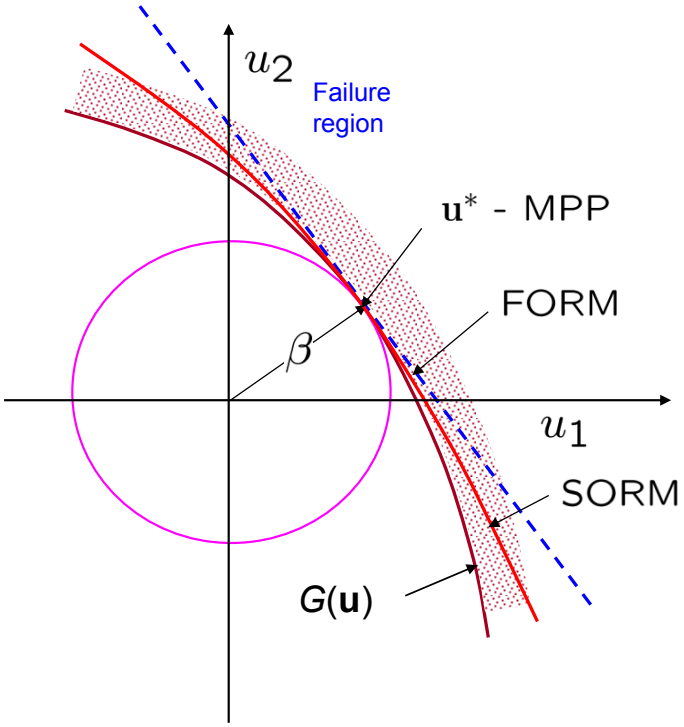


Figure 6. Example of reliability optimization.

There are local and global reliability methods. The majority of the methods available are local, meaning that a local optimization formulation is used to locate one MPP. In contrast, global methods can find multiple MPPs if they exist. Space limitations do not permit going into much detail about reliability methods. The interested reader is encouraged to start with Haldar and Mahadvan's *Probability, Reliability, and Statistical Methods in Engineering Design* [53], which provides an excellent overview of local reliability methods.

Next, we provide a brief description of one global reliability method, Efficient Global Reliability Analysis (EGRA) [54], because of its leveraging of Gaussian process surrogates.

The idea is to construct an initial Gaussian process (GP) model of the response function and then update it strategically by adding points that are likely to be near the region of the failure surface, to get a better approximation of the failure surface. EGRA begins with a GP model built from a very small number of samples, and then adaptively chooses where to generate subsequent samples to increase accuracy in the vicinity of the limit state. The resulting GP model is then sampled using multimodal adaptive importance sampling to calculate the probability of exceeding (or failing to exceed) the response level of interest.

By locating multiple points on or near the limit state, problems having more complex and nonlinear limit states can be modeled, leading to more accurate probability integration. By concentrating samples in the area where accuracy is important (i.e., in the vicinity of the limit state), a much-reduced number of function evaluations is required to build an accurate surrogate model of the limit state surface. Hence, the method is often more computationally efficient than other UQ approaches. Note that EGRA has limitations: the response function needs to be well approximated by a GP, and the GP construction is limited to dozens of random variables (GPs are

usually built on ten variables or fewer). Thus, EGRA is not a method to use if one has hundreds or thousands of random inputs.

7. Illustration of UQ Methods on an Example Problem

This section demonstrates and compares the various UQ methods described above for an example problem. It is not an exacting performance study, but it is meant to be illustrative of the different UQ methods. We chose sample numbers per methods that experience indicates are appropriate for each method, but other sample sizes could have been tried. For example, we took 1,000 Monte Carlo samples for the baseline case. For the polynomial chaos method, we chose a fifth-order expansion in each dimension resulting in $5^3 = 125$ samples. The EGRA reliability method was run to convergence, and the number of samples varied per starting seed, but typically took about 30 samples. We used 50 samples to build a GP surrogate, then sampled it 10,000 times to obtain statistics on the output. These choices are somewhat arbitrary but are meant to give some sense of what a practitioner might do.

7.1. Cantilever Beam Example Problem

This section describes the example that will be used to demonstrate the UQ approaches. The problem involves a simple cantilever beam as shown in Figure 7. The goal is to understand how the deflection of the beam varies with respect to the length, width, and height of the beam, as well as to applied load and elastic modulus of the beam.

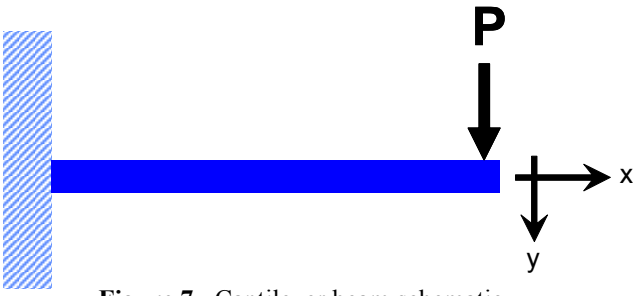


Figure 7. Cantilever beam schematic.

The variables characterizing the beam problem are shown in Table 1, along with nominal values.

Table 1. Variables characterizing the cantilever beam problem.

Variable	Description	Nominal Value
L	Length	1 m
W	Width	1 cm
H	Height	2 cm
I	Area moment of inertia	1/12 WH^3
P	Load	100 N
E	Elastic modulus of aluminum 6061-T6	69 GPa

The formula for beam deflection, D, assuming simplifying assumptions such as isotropic material properties of the beam, elastic displacement, and negligible beam mass is:

$$D = PL^3/(3EI). \tag{10}$$

For this problem with the nominal parameter values from Table 1, the deflection is 7.2 cm.

7.2. Sampling

We first demonstrate sampling with one uncertain variable for the cantilever beam example. In this example, we assume that the elastic modulus, E, is normally distributed with a mean of 6.9 GPa and a standard deviation of 10% of the mean (0.69 GPa). A histogram of 1,000 Latin hypercube (LHS) samples of E samples is shown in Figure 8.

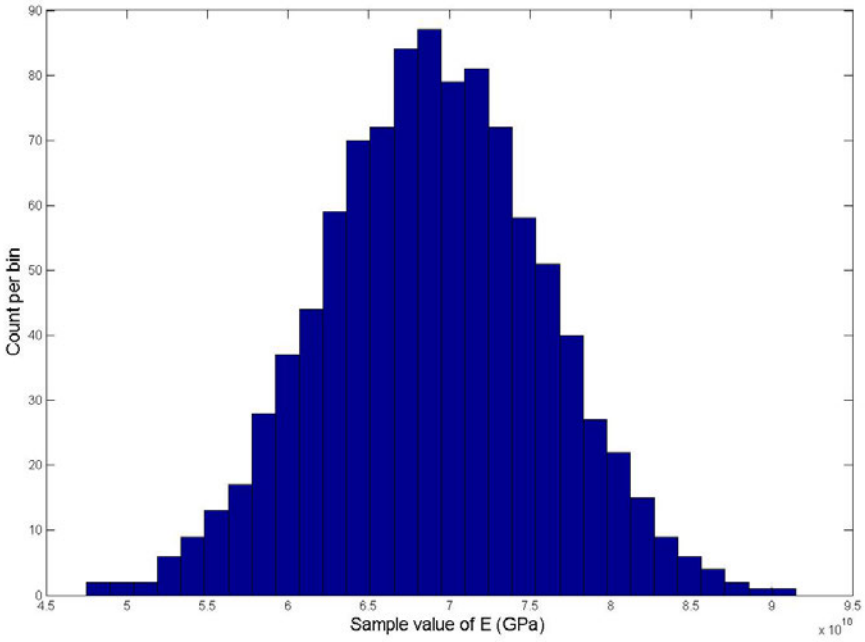


Figure 8. Histogram of 1000 input samples of E.

A corresponding histogram of the beam displacement is shown in Figure 9. This histogram is based on these 1,000 input samples of E substituted into the formula for beam deflection with the other parameter values assigned the nominal values specified in Table 1. Note that the output histogram is non-normal due to the nonlinearity of deflection with respect to E. Normal inputs only translate to normal outputs when the models are linear in the inputs.

Also note that Figure 8 involves sampling of only one input, E. If we looked at all combinations of sampling each of the six variables in Table 1 at five levels, we would get a large number of combinations: $5^6 = 15{,}625$. To study all combinations for 10 levels of all of the factors would result in one million combinations. Such enumeration of factor levels is seldom practical for real applications because of the combinatorial explosion of the number of combinations as the number of variables and levels increase. The combinatorial explosion issue motivates the use of other methods (e.g., statistical sampling, reliability, polynomial chaos) such as those we present here.

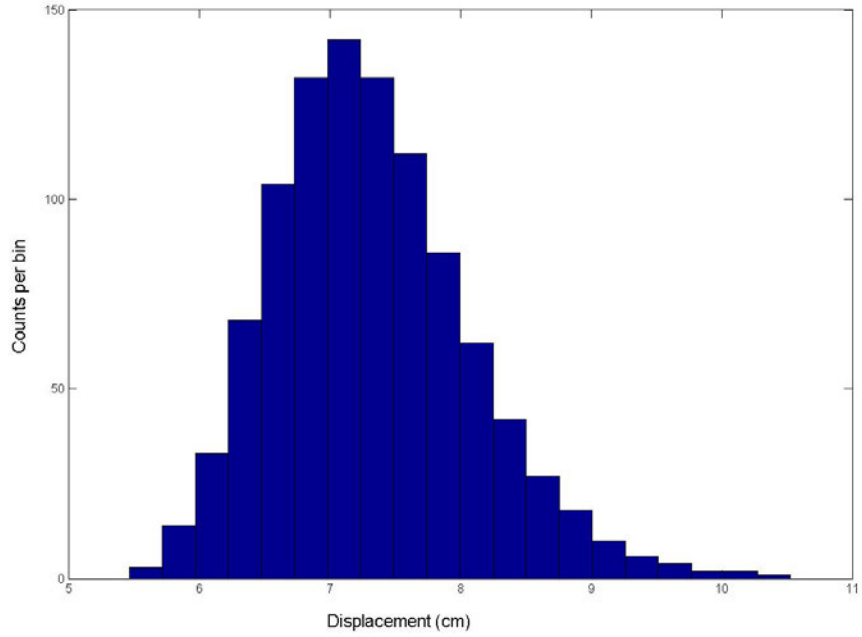


Figure 9. Histogram of 1,000 output samples of displacement.

The next example treats multiple inputs as uncertain, distributed with the probability distributions specified in Table 2. Note that 3 variables are uncertain in this problem: L, P, and E. Also note that the size of the standard deviations relative to the means (coefficient of variation) varies quite a bit, to represent differences in the fidelity of the measurements and the knowledge about these quantities. For example, the length of the beam can presumably be measured accurately, so the coefficient of variation of L is 0.01, whereas the elastic modulus may be quite uncertain: its coefficient of variation is 0.1.

Table 2. Uncertain variable specification for cantilever example.

Variable	Distribution	Distribution Parameters
L	Normal	– Mean = 1m – Std. Dev. = 0.01 m
W	Fixed	1 cm
H	Fixed	2 cm
P	Normal	– Mean = 100 N – Std. Dev. = 5 N
E	Normal	– Mean = 69 GPa – Std. Dev. = 6.9 GPa

When generating 1,000 LHS samples from these three normal distributions and then using the sample values to calculate deflection of the beam, we obtain the histogram of beam displacement shown in Figure 10:

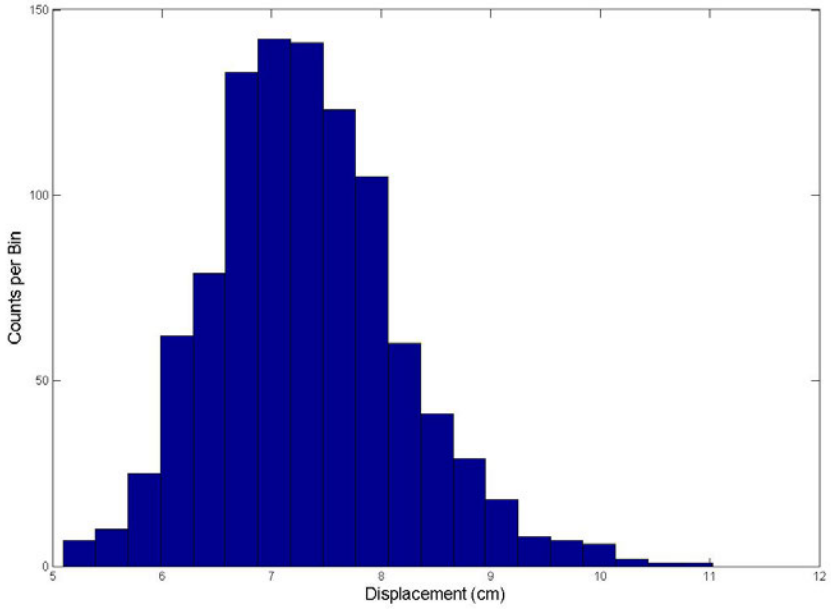


Figure 10. Histogram of 1,000 output samples of displacement.

The cumulative distribution function (CDF) of the output samples is shown in Figure 11. The CDF is defined as the probability that the unknown response is less than a particular value, in this case the Prob(D < x). For example, in Figure 11, we see that the probability that displacement is less than 7 cm is about 40% or 0.4, but the probability that displacement is less than 8 cm is about 80%.

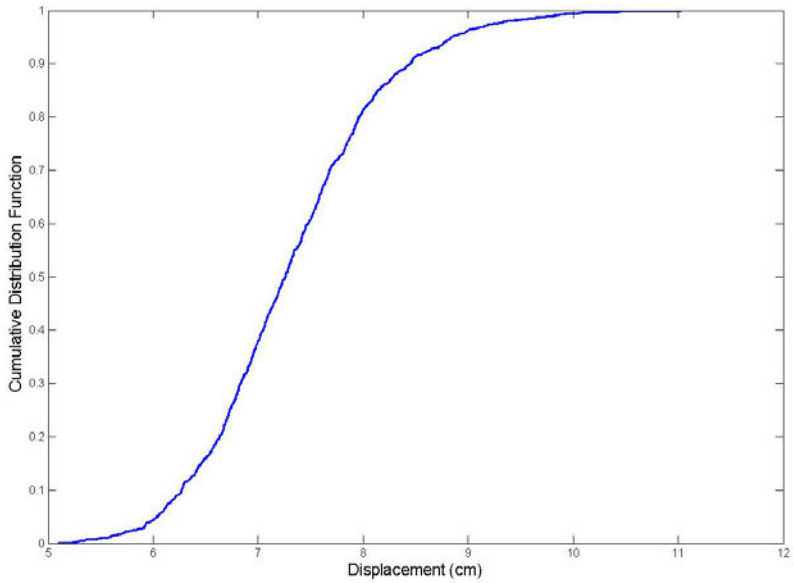


Figure 11. CDF of 1,000 output samples of displacement.

We can also examine the correlations of the output with respect to the inputs. The scatterplots in Figure 12 show that the displacement uncertainty is not strongly correlated to the uncertainty in

length L. It is more strongly correlated with load P, but is very strongly negatively correlated to the uncertainty in the modulus of elasticity E. The graphs also enable quick identification of outliers.

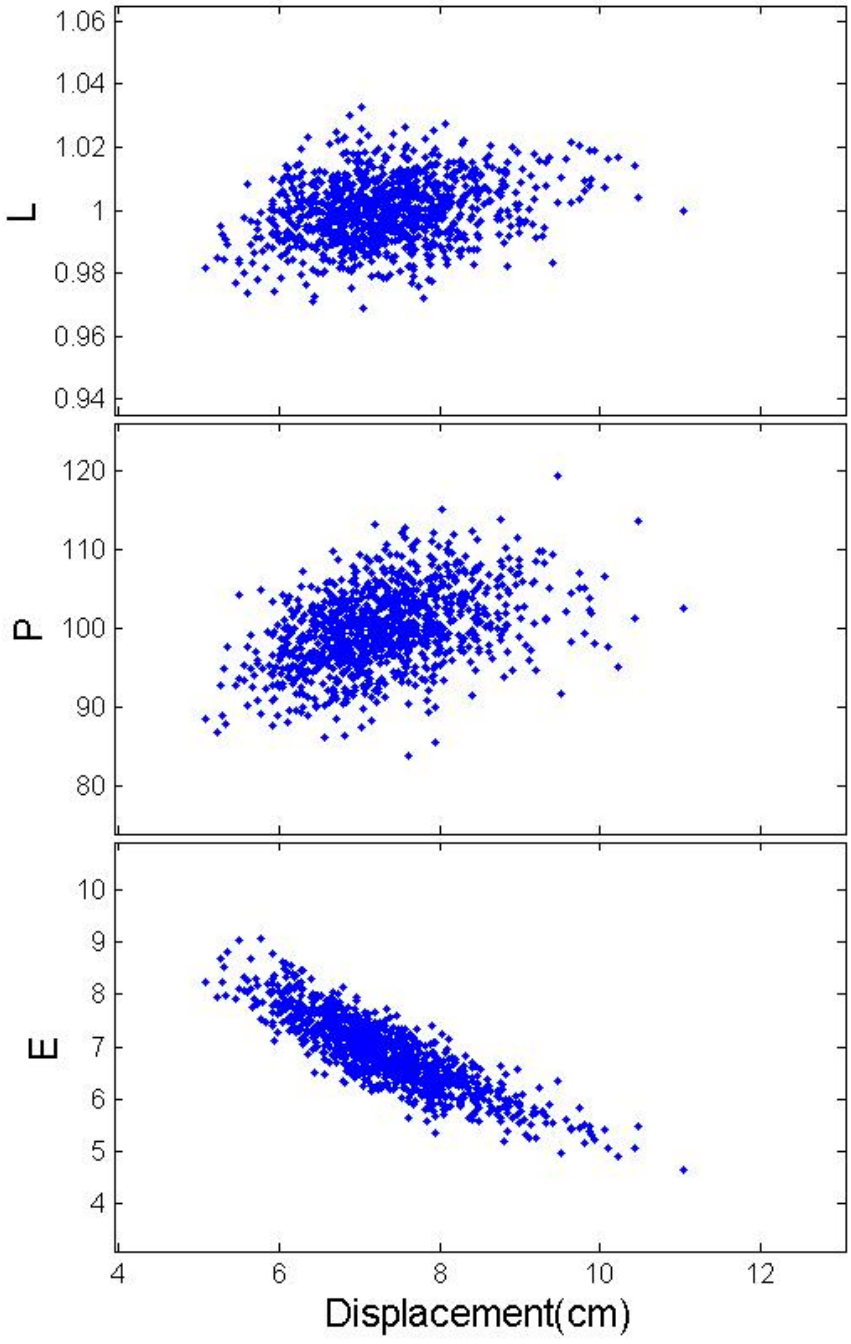


Figure 12. Scatterplot of displacement vs. uncertain input variables.

In addition to uncertainty analysis (examining the distribution of the displacement output in figures 10 and 11), the 1,000 LHS output samples may be used for sensitivity analysis. First, we will examine simple and partial correlations between the inputs and the output, shown in Table 3:

Table 3. Correlation analysis based on 1,000 LHS samples.

INPUT	Simple Correlation wrt Displacement	Partial Correlation wrt Displacement
L	0.26	0.88
P	0.42	0.95
E	- 0.86	- 0.99

This example shows how simple correlations (Pearson correlations) may differ from the partial correlations. For example, simple correlations would indicate that E is the most important input variable and strongly negatively correlated with the output. Partial correlations, which can account for the effect of the other variables (specifically the large contribution of E), indicate that L and P have a strong positive correlation with displacement as well. The simple correlations of L and P with respect to displacement are not insignificant in an absolute sense even though they are overshadowed in a relative sense by the uncertainty contribution from E.

If we perform a variance-based decomposition (VBD), we obtain the variance-based indices shown in Table 4:

Table 4. Variance-based decomposition indices based on LHS.

INPUT	Main Effect VBD Index	Total Effect VBD Index
L	0.067	0.064
P	0.176	0.173
E	0.756	0.765

The main effect VBD index indicates the percentage of output variance that is attributed to the effect of the individual variables. The variance of the elastic modulus contributes about three quarters of the displacement variance. The total effect index measures the effect of the individual variable as well as its interaction with other variables. In this example, since main and total effect indices are about the same, there is little interaction between the variables in terms of contribution to displacement variance.

The example presented in Figures 10–12 above, with 3 uncertain variables (L, P, and E), will also be used in the subsequent uncertainty analysis methods.

7.3. Polynomial Chaos Expansion

Figure 13 shows the result of a CDF based on the sampling of a polynomial chaos expansion. The expansion was constructed over quadrature points. The order of the quadrature is 5 in each uncertain dimension, for a tensor product total of 125 integration points. We calculated the quantile estimates by taking 10,000 samples of the polynomial chaos expansion and sorting them. For comparison purposes, we plotted the CDF obtained from the 1,000 LHS samples, which was also shown in Figure 11. The PCE estimate is smoother because it is a summation of polynomials. Finally, note that in this example, the expansion approximates the beam displacement function D(L,P,E). This is an analytic formula that is well-behaved with no discontinuities, non-smoothness, or bifurcations. A fifth-order quadrature will resolve polynomial terms to order (5-1)=4, which is adequate for this function.

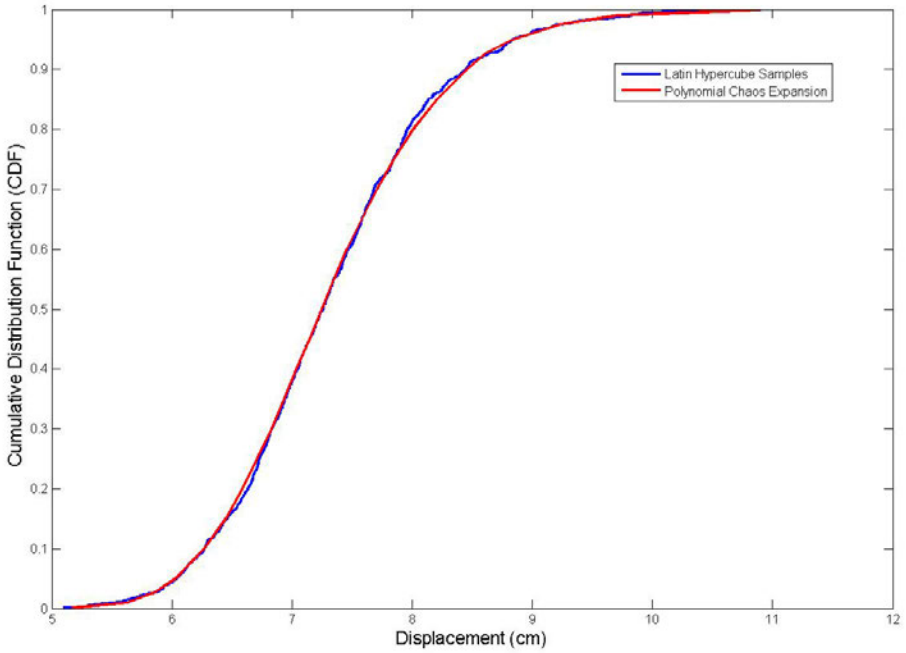


Figure 13. CDF based on polynomial chaos expansion method.

One advantage of the PCE is the variance-based indices can be calculated for free, essentially: they are analytic functions of the expansion coefficients. Table 5 shows the VBD indices for the three uncertain variables. Note that they are almost identical to those calculated by the LHS sample in Table 4.

Table 5. Variance based decomposition indices based on polynomial chaos expansion.

INPUT	Main Effect VBD Index	Total Effect VBD Index
L	0.064	0.065
P	0.177	0.179
E	0.756	0.759

7.4. Gaussian Process

The next example involves constructing a Gaussian process (GP) surrogate over a limited number of function evaluations, then sampling the surrogate to obtain the quantile estimates for the CDF and also the variance-based indices. We constructed a Gaussian process surrogate based on 50 function evaluations, and then sampled it 10,000 times. The resulting CDF is shown in Figure 14. Again, we plotted the CDF obtained from the 1,000 LHS samples for comparison purposes.

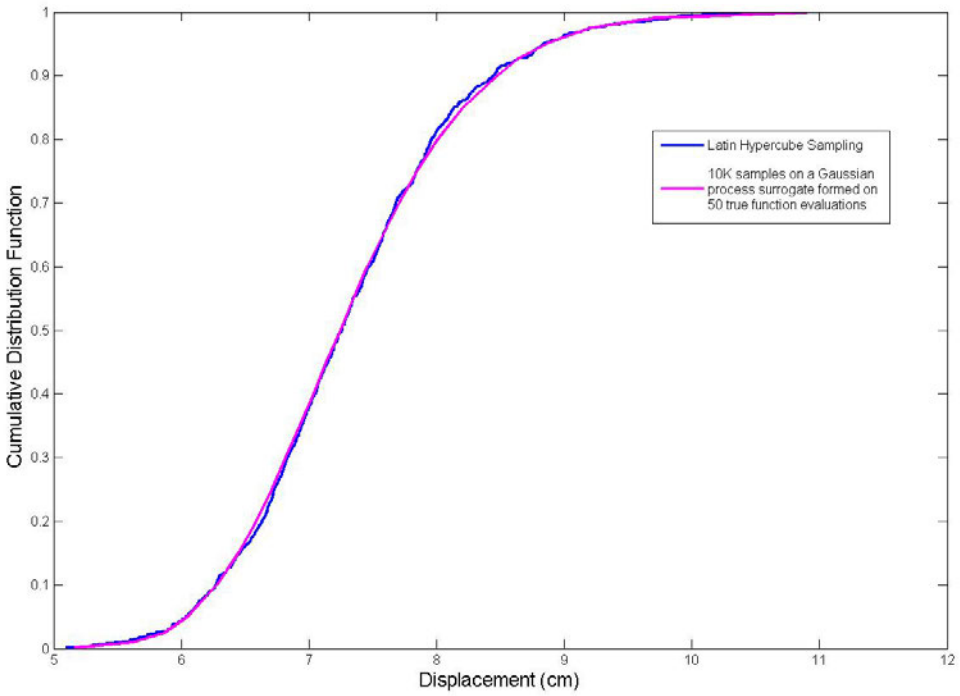


Figure 14. CDF based on sampling a Gaussian Process Surrogate model.

Table 6 shows the VBD indices, which are almost identical to the results obtained by LHS and PCE. For this problem, it appears that a GP surrogate based on 50 samples is sufficiently accurate to model the function for many aspects of uncertainty quantification.

Table 6. Variance based decomposition indices based on a Gaussian process surrogate.

INPUT	Main Effect VBD Index	Total Effect VBD Index
L	0.065	0.064
P	0.180	0.178
E	0.760	0.759

7.5. Reliability Analysis

Finally, we performed a global reliability EGRA optimization on this problem. Reliability optimization is focused on finding a particular percentile: given a response value R, the reliability optimization attempts to find the CDF value corresponding to that response, e.g., the probability

that displacement is less than R. This focus represents a difference from the other methods, which all calculate numerous responses and then order them to obtain the PDFs/CDFs from which the quantile estimates are obtained. We defined a set of response values for which we performed this optimization, specifically twelve response levels: 4, 5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9, 10, and 11 cm. For each of these twelve levels, an optimization is performed, resulting in 29 function evaluations total. This involved an EGRA run at 4 cm, which took 15 samples. Then, an EGRA targeting the next response level was run, leveraging all previous samples. This was repeated until all levels were analyzed, for a total of 29 samples. The resulting CDF is shown in Figure 15; again, it is very similar to the others. As found previously, a GP surrogate works well for modeling the displacement, and this enables the EGRA-based approach to perform well.

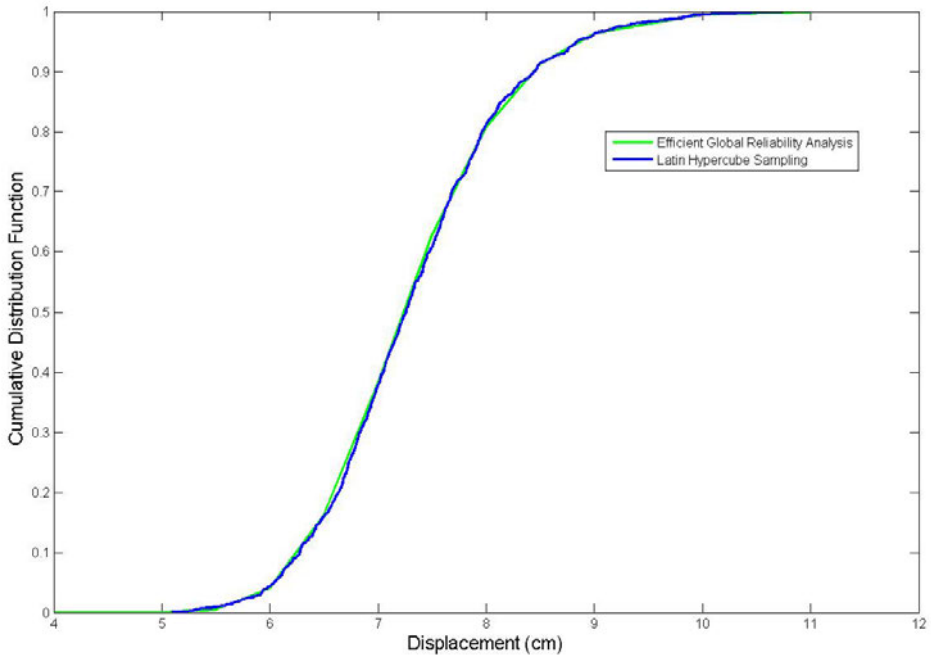


Figure 15. CDF based on the efficient global reliability analysis method.

7.6. Summary Statistics and Error Estimation

Finally, we list the estimated displacement values for various percentiles (CDF levels) calculated by each method. These are shown in tables 7 and 8. We wanted to provide some estimates of the precision error associated with each method because one can obtain slightly different results with most of these methods by running with different seeds. Thus, in contrast to the results presented above where the method (e.g., LHS or polynomial chaos) was run just one time, for these results we have run each method 100 times. The percentile values are then averaged across the 100 runs. Thus, for one LHS sample set of 1,000 sample points we calculate the 0.001 percentile, the 0.01 percentile, and so on. Then, we repeat this 100 times for different LHS sets of 1,000 samples, at which point we average the 0.001 percentile estimates. These are reported in the tables as mean estimates.

The two error measures we examine are the mean and maximum absolute value of the percent error: mean |% error| and max |% error|. These are defined as follows:

$$\text{mean } |\%\ \text{error}| = \frac{\sum_{i=1}^{n} \left| \frac{\text{estimate}_i - \text{gold standard}}{\text{gold standard}} \right|}{n} \tag{11}$$

$$\max |\%\ \text{error}| = \max_i \left| \frac{\text{estimate}_i - \text{gold standard}}{\text{gold standard}} \right| \quad (12)$$

The data in tables 7 and 8, below, indicate that there are some differences in the performance of the methods, especially between sampling and the other two methods (PCE and the GP surrogate) at the tail percentiles near 0.001 and 0.999. For example, LHS had a mean |% error| of 2.76% for the 0.001 percentile, while PCE and the GP surrogate both had a mean |% error| of less than 1%. The mean |% errors| at the median (percentile = 0.5) are much smaller: only 0.21% for sampling, but 0.07% and 0.06% for PCE and the GP surrogate respectively. Overall, LHS exhibits about three times the error of the other methods. Table 8, which shows the maximum value |% error|, indicates that sampling can be particularly bad at the tails: the max |% error| is 11.5% for LHS but less than 3% for PCE and the GP surrogate. Finally, note that for this particular problem, the GP surrogate has slightly lower error estimates than the PCE method. In both cases, one constructs a surrogate (a GP or a polynomial chaos expansion). To obtain the percentile estimates, each surrogate is sampled 10,000 times and the percentiles are calculated from those estimates.

Finally, we looked at the bias in the errors, where error for each particular percentile value is defined as the estimate minus the gold standard, with no absolute value or relative scaling:

$$error_i = estimate_i - gold\ standard.$$

For all methods, the bias in the errors was close to zero. For example, the mean of the errors for the 0.001 percentile is -0.04 cm for LHS, -0.004 cm for PCE, and -0.007 cm for the GP surrogate. The mean of the errors for the median (0.5 percentile) is -0.003 cm for LHS, +0.0009 cm for PCE, and -0.0001 cm for the GP surrogate.

Histograms of the errors for the 0.001 percentile and the 0.5 percentile are shown in Figure 16, below the tables.

Table 7. Comparison of the percentile estimates across methods.

	Gold Standard	LH Sampling	LH Sampling	PCE	PCE	GP Surrogate	GP Surrogate
CDF	Mean Estimate	Mean	Mean	Mean	Mean	Mean	Mean
Percentile	10M Samples	Estimate	|% error|	Estimate	|% error|	Estimate	|% error|
0.001	5.196	5.152	2.76%	5.192	0.78%	5.189	0.68%
0.01	5.621	5.612	0.68%	5.618	0.28%	5.618	0.25%
0.025	5.841	5.848	0.47%	5.839	0.17%	5.839	0.17%
0.05	6.039	6.042	0.33%	6.036	0.13%	6.038	0.12%
0.1	6.279	6.279	0.28%	6.279	0.08%	6.279	0.08%
0.15	6.449	6.447	0.24%	6.449	0.07%	6.449	0.08%
0.2	6.589	6.586	0.21%	6.589	0.07%	6.589	0.08%
0.25	6.713	6.709	0.23%	6.713	0.07%	6.714	0.07%
0.3	6.826	6.822	0.18%	6.827	0.06%	6.827	0.06%
0.35	6.934	6.931	0.20%	6.935	0.06%	6.935	0.06%
0.4	7.039	7.036	0.19%	7.040	0.07%	7.040	0.06%
0.45	7.143	7.139	0.23%	7.144	0.07%	7.143	0.06%
0.5	7.247	7.244	0.21%	7.248	0.07%	7.247	0.06%
0.55	7.354	7.350	0.20%	7.355	0.06%	7.353	0.07%
0.6	7.465	7.463	0.20%	7.465	0.07%	7.464	0.06%
0.65	7.582	7.579	0.21%	7.582	0.06%	7.582	0.07%
0.7	7.709	7.711	0.22%	7.708	0.07%	7.708	0.08%
0.75	7.849	7.848	0.22%	7.849	0.07%	7.849	0.06%
0.8	8.011	8.008	0.26%	8.010	0.07%	8.011	0.07%
0.85	8.205	8.198	0.25%	8.205	0.09%	8.205	0.08%
0.9	8.461	8.462	0.36%	8.462	0.10%	8.459	0.09%
0.95	8.866	8.871	0.44%	8.865	0.15%	8.863	0.13%
0.975	9.243	9.238	0.64%	9.240	0.18%	9.241	0.18%
0.99	9.718	9.696	1.00%	9.717	0.28%	9.715	0.29%
0.999	10.853	10.685	2.40%	10.810	0.93%	10.821	0.88%

Note: For a fixed percentile (e.g., 0.95), these methods provide an estimate of the displacement response. Note that each method has been run for 100 trials, so the results presented are mean estimates. We also present the mean absolute value of the % error.

Table 8. Comparison of the percentile estimates across methods.

CDF Percentile	Gold Standard Mean Estimate 10M Samples	LH Sampling Mean Estimate	LH Sampling Max |% error|	PCE Mean Estimate	PCE Max |% error|	GP Surrogate Mean Estimate	GP Surrogate Max |% error|
0.001	5.196	5.152	11.50%	5.1915	2.66%	5.189	2.44%
0.01	5.621	5.612	2.81%	5.6176	0.81%	5.618	0.73%
0.025	5.841	5.848	1.44%	5.8392	0.56%	5.839	0.47%
0.05	6.039	6.042	1.37%	6.0365	0.51%	6.038	0.40%
0.1	6.279	6.279	1.05%	6.2790	0.26%	6.279	0.43%
0.15	6.449	6.447	0.78%	6.4493	0.32%	6.449	0.23%
0.2	6.589	6.586	0.78%	6.5889	0.26%	6.589	0.26%
0.25	6.713	6.709	0.83%	6.7128	0.21%	6.714	0.22%
0.3	6.826	6.822	0.65%	6.8268	0.20%	6.827	0.19%
0.35	6.934	6.931	0.76%	6.9352	0.25%	6.935	0.24%
0.4	7.039	7.036	0.60%	7.0400	0.27%	7.040	0.23%
0.45	7.143	7.139	0.81%	7.1441	0.23%	7.143	0.21%
0.5	7.247	7.244	0.76%	7.2481	0.21%	7.247	0.26%
0.55	7.354	7.350	0.76%	7.3546	0.20%	7.353	0.22%
0.6	7.465	7.463	0.67%	7.4651	0.18%	7.464	0.25%
0.65	7.582	7.579	0.71%	7.5818	0.21%	7.582	0.26%
0.7	7.709	7.711	0.75%	7.7082	0.21%	7.708	0.23%
0.75	7.849	7.848	0.68%	7.8493	0.22%	7.849	0.23%
0.8	8.011	8.008	0.66%	8.0099	0.26%	8.011	0.22%
0.85	8.205	8.198	1.00%	8.2051	0.29%	8.205	0.22%
0.9	8.461	8.462	1.37%	8.4620	0.28%	8.459	0.37%
0.95	8.866	8.871	1.57%	8.8650	0.46%	8.863	0.57%
0.975	9.243	9.238	2.18%	9.2400	0.63%	9.241	0.56%
0.99	9.718	9.696	2.93%	9.7173	1.03%	9.715	0.95%
0.999	10.853	10.685	6.91%	10.8100	3.51%	10.821	2.99%

Note: For a fixed percentile (e.g., 0.95), these methods provide an estimate of the displacement response. Note that each method has been run for 100 trials, so the results presented are mean estimates. We also present the mean absolute value of the % error.

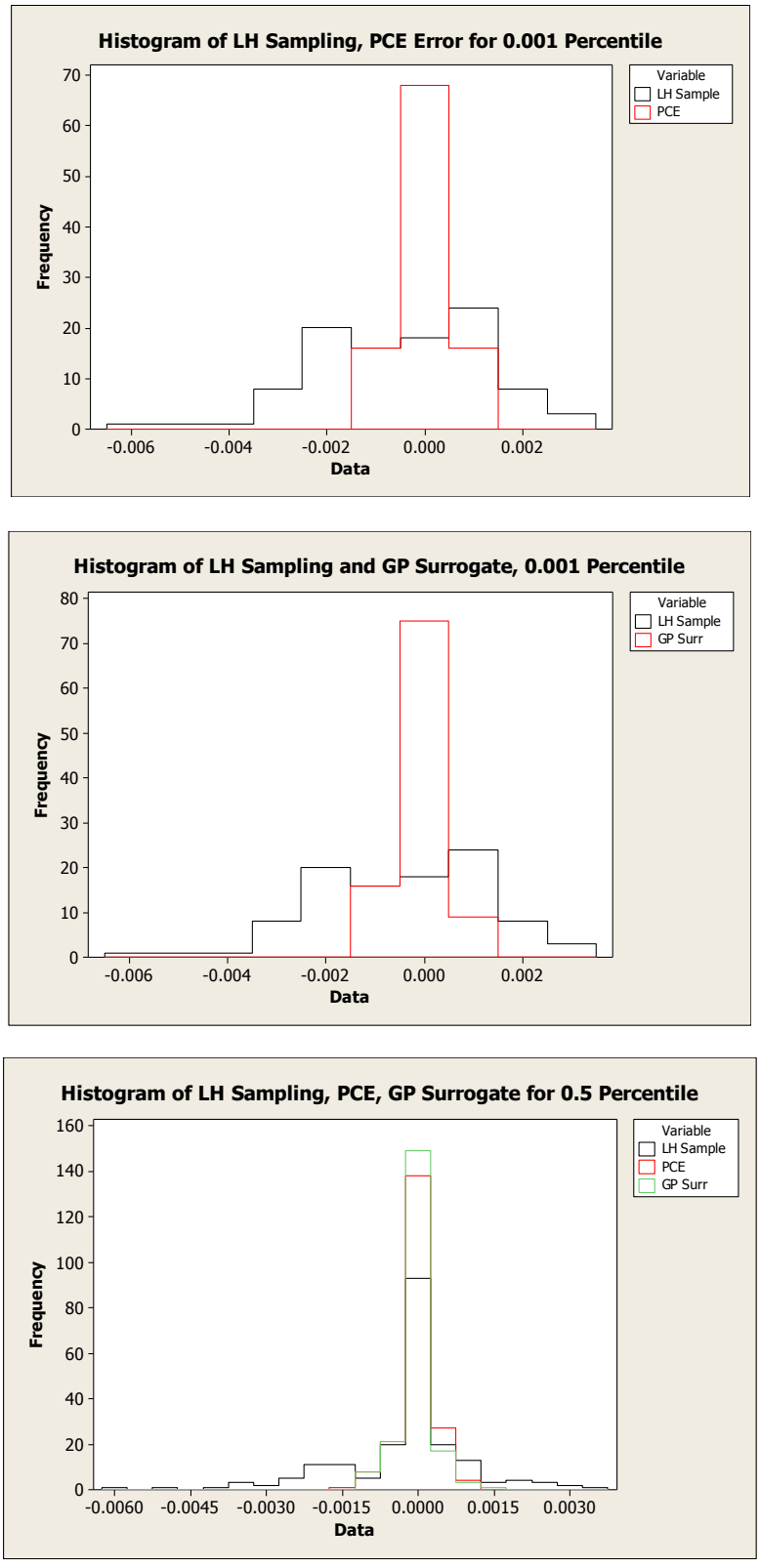


Figure 16. Histograms of error for 0.001 percentile (top two plots) and 0.5 percentile (bottom plot).

Since EGRA operates by calculating a percentile value for a particular response, the results EGRA produces are slightly different from the sampling and surrogate methods. We therefore list the results for the EGRA errors in Table 9.

The sampling and surrogate methods produce complementary CDF estimates. Recall that the CDF of a function $g(\mathbf{X})$ is defined as:

$$F_g(z) = P[\, g(\mathbf{X}) \leq z]$$

F_g has values between *(0,1)*, whereas the response z has values over the range defined by $g(\mathbf{X})$. For the sampling methods, we specify particular values of F_g such as 0.01, 0.5, etc., and estimate the corresponding z values. For reliability methods, we specify the z values of the response, and estimate the F_g percentiles. Finally, note that the error estimates are highest for the smallest displacement levels. This is partly due to the way we define error—as the absolute relative error—so we are dividing by smaller percentile estimates for lower displacements values. The absolute error is on the order of 0.001 for most of these percentile estimates.

Table 9. Percentile estimates with EGRA. For a fixed response (e.g., 8 cm), the reliability method provides an estimate of corresponding percentile.

Displacement (cm)	Gold Standard CDF Percentile Estimate	EGRA Mean Estimate	EGRA Mean |% error|	EGRA Max |% error|
4	0.00000	5.39E-09	---	---
5	0.00026	0.00024	11.50%	34.13%
5.5	0.00562	0.00536	6.51%	22.82%
6	0.04405	0.04345	4.09%	17.88%
6.5	0.16721	0.16677	2.59%	11.63%
7	0.38115	0.38106	1.58%	7.46%
7.5	0.61542	0.61957	1.12%	4.77%
8	0.79695	0.79880	0.60%	4.18%
8.5	0.90610	0.90594	0.29%	2.77%
9	0.96074	0.95993	0.16%	1.57%
10	0.99430	0.99391	0.04%	0.34%
11	0.99926	0.99919	0.01%	0.05%

A summary comparison of the uncertainty in the displacement response is shown in Table 10, where the first four moments of the displacement distribution are estimated by three of the methods (again, EGRA does not provide moment estimation). The results presented are the means of 100 iterations of each method. Thus, for LHS, the first four moments (mean, standard deviation, skewness, and kurtosis), are calculated for one sample set of 1,000 samples. This is done 100 times, and the estimate in the table represents the mean of 100 estimates of the mean. Similarly, the mean of 100 standard deviation estimates is reported in the table, and similarly for skew and kurtosis.

All methods give accurate estimates of the mean: the mean |% error| is less than 0.01 percent error. The mean |% error| for the standard deviation is also small; LHS has the largest error at 0.43%. The mean |% error| is much larger for skewness and kurtosis, especially for LHS: 11.6% error for skewness and 38.9% error for kurtosis. Note that PCE performs the best for moment estimation. This is partly due to the nature of the polynomial chaos expansions, which use customized bases to capture response distribution characteristics accurately. The max |% error| values are very large for LHS, and much better for PCE and the GP surrogate.

Table 10. Comparison of moments across methods.

	First Four Moments of Displacement (cm) Mean estimates from 100 iterations of methods			
Method	Mean	Std. Deviation	Skewness	Kurtosis
Gold Standard 10M Samples	7.323318	0.869388	57.163783	70.471190
LH Sampling	7.323235	0.869277	56.312778	68.359362
PCE	7.323328	0.869415	57.122174	69.576398
GP Surrogate	7.323313	0.869173	56.764502	67.385377
	Mean |% error| First Four Moments of Displacement (cm)			
Method	Mean	Std. Deviation	Skewness	Kurtosis
LH Sampling	0.0061%	0.430%	11.62%	38.93%
PCE	0.0001%	0.003%	0.07%	1.27%
GP Surrogate	0.0012%	0.107%	3.32%	12.46%
	Max |% error| First Four Moments of Displacement (cm)			
Method	Mean	Std. Deviation	Skewness	Kurtosis
LH Sampling	0.0251%	1.688%	44.88%	183.49%
PCE	0.0001%	0.003%	0.07%	1.27%
GP Surrogate	0.0044%	0.392%	11.79%	54.81%

In this section, we have attempted to show the types of results that can be obtained for sensitivity analysis and uncertainty quantification using a variety of methods. Though the demonstration was on a simple problem, the example is illustrative for the reader to understand the costs, benefits, and issues associated with each method. Table 11 shows the relative cost of each method demonstrated in this example.

Table 11. Relative cost of methods demonstrated in this section.

	Latin Hypercube Sampling	Polynomial Chaos	GP Surrogate	Global Reliability
Number Fn. Evals	1000	125	50	29

Please keep in mind that a real analysis may involve different costs. For example, you could choose to use more or fewer Latin Hypercube samples; you could perform the polynomial chaos expansion with a lower or higher fidelity set of quadrature points, resulting in a different number

of total samples; or you could use a different number of build points for the Gaussian process surrogate. Also, the optimization performed for the global reliability method may take a different number of function evaluations, depending on the number of inputs and convergence criteria for stopping.

8. Summary of UQ Methods

This chapter has outlined some useful approaches and critical issues when performing uncertainty quantification of computer models. We have discussed sample design, including Monte Carlo sampling methods, Latin Hypercube sampling, and quadrature methods, as well as two main approaches for sensitivity analysis: correlation and variance-based indices. We have also discussed using surrogates as approximations to computer models, in particular Gaussian process models and stochastic expansion methods; and propagation methods, including sampling and reliability methods. Finally, we demonstrated these approaches (sampling, quadrature, sensitivity analysis, use of surrogates, and reliability analysis) on a cantilever beam example problem.

When choosing an approach, one should consider a variety of factors, including the computational cost of the function evaluations, the transparency of the method, the type of outputs generated, and the question being addressed: is the goal to get an accurate assessment of the response level at a particular percentile, such as a failure probability, or are the mean and/or variance of the output distribution sufficient? Is an entire output CDF necessary? Sampling and surrogate approaches are geared to obtain entire CDF estimates, but reliability methods only focus on one particular percentile of the CDF at a time.

Figure 17 shows one depiction of the choices one can make while selecting an approach. The boxes highlighted in pink are methods that were discussed in this chapter. At a high level, one can choose a structured sampling approach, a random sampling approach, or a reliability-based approach. Note that the sizes of the boxes are not proportionate to the prevalence of the method. The most common method is one that goes through the center of the diagram: random sampling such as LHS, used in a non-adaptive setting, where the response statistics are computed directly from the samples on the computer simulation. There are many ways to categorize sampling, surrogate, and uncertainty analysis approaches. Figure 17 shows one such contextual map.

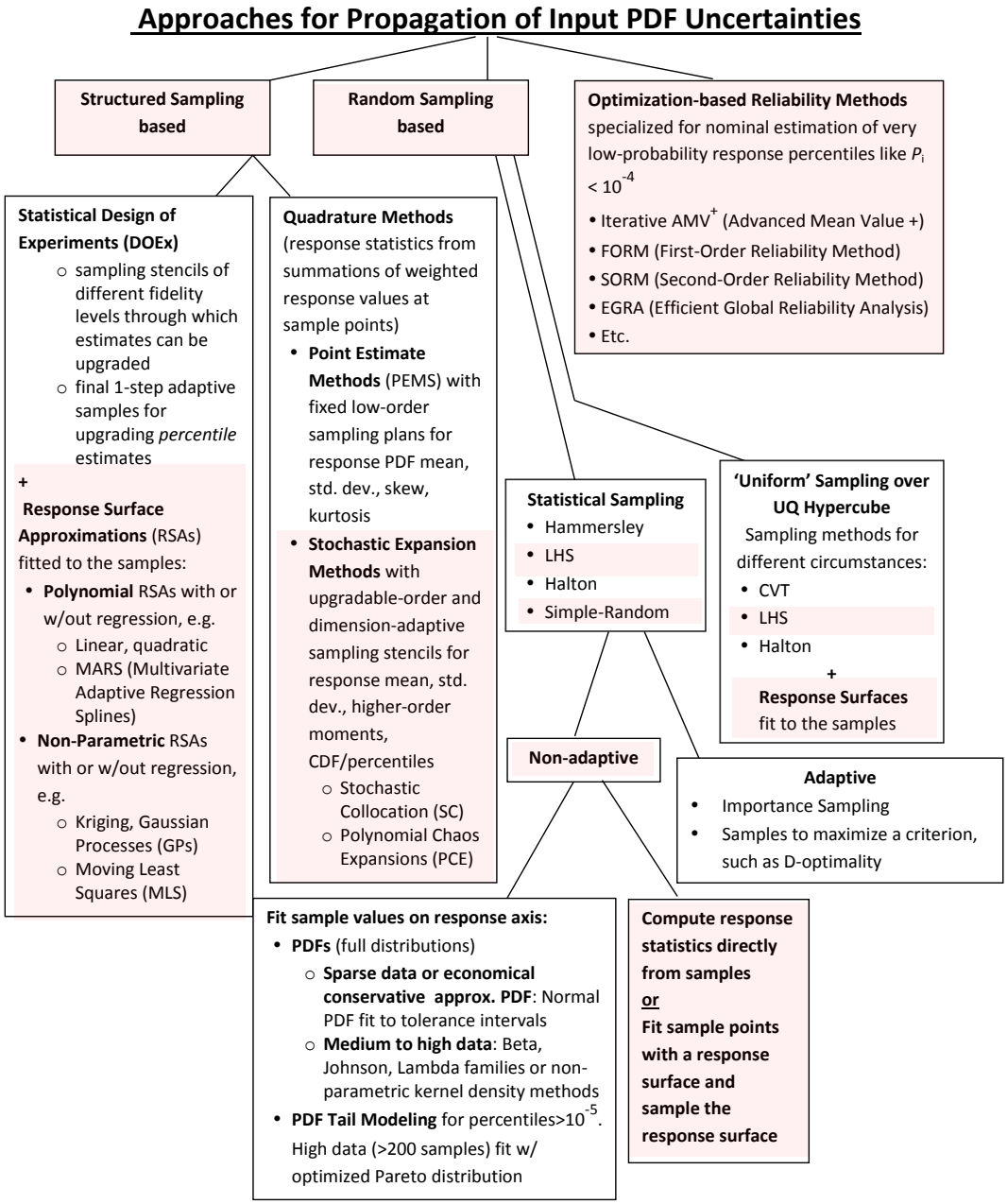


Figure 17. Categorization of various approaches for propagating input uncertainties.

References

1. Helton, J. C., "Conceptual and Computational Basis for the Quantification of Margins and Uncertainty," Sandia Technical Report 2009-3005.
2. Diegert, K., Klenke, S., Novotny, G., Paulsen, R., Pilch, M., and Trucano, T., "Toward a More Rigorous Application of Margins and Uncertainties within the Nuclear Weapons Life Cycle – A Sandia Perspective," Sandia Technical Report SAND2007-6219.
3. Meyers, M. A., and Booker, J. M., *Eliciting and Analyzing Expert Judgment: A Practical Guide*, ASA-SIAM Series on Statistics and Applied Mathematics, SIAM 2001. doi: http://dx.doi.org/10.1137/1.9780898718485.
4. Morgan, M. G., and Henrion, M., *Uncertainty: A guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis,* Cambridge University Press, 1990.
5. Santner, T., Williams, B., and Notz, W., *The Design and Analysis of Computer Experiments,* New York, NY: Springer, 2003.
6. Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., "Design and analysis of computer experiments," *Statistical Science*, Vol. 4, No.4, pp. 409-435, 1989.
7. Helton, J. C., and Davis, F. J., "Latin Hypercube Sampling and the Propagation of Uncertainty in Analyses of Complex Systems," *Reliability Engineering and System Safety*, Vol. 81, pp. 23-69, 2003.
8. Halton, J. H., "On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals," *Numerische Mathematik*, Vol. 2, pp. 84-90, 1960.
9. Knuth, D. E., *The Art of Computer Programming*, Vol. 2: Seminumerical Algorithms. 3rd edition, 1997.
10. Romero V. J., Burkardt, J. S., Gunzburger, M. D., Peterson, J. S., and Krishnamurthy, T., "Initial application and evaluation of a promising new sampling method for response surface generation: Centroidal Voronoi Tessellation," paper AIAA-2003-2008, 44th Structures, Structural Dynamics, and Materials Conference, Norfolk, VA, Apr. 7-10, 2003.
11. Romero, V. J., Krishnamurthy, T., Swiler, L. P., "Progressive Response Surfaces," 9th ASCE Specialty Conference on Probabilistic Mechanics and Structural Reliability, Albuquerque, NM, July 26-28, 2004.
12. Simpson, T. W., Toropov, V., Balabanov, V., and Viana, F. A. C., "Design and analysis of computer experiments in multidisciplinary design optimization: A review of how far we have come or not," in the proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference), Victoria, British Columbia, Canada, paper AIAA 2008-5802, September 2008.
13. Swiler, L. P., Slepoy, R., Giunta, A. A., "Evaluation of Sampling Methods in Constructing Response Surface Approximations," 8th AIAA Non-Deterministic Approaches Conference (part of 47th AIAA/ASME/ASCE/AHS/ASC SDM conference) Newport, RI, paper AIAA 2006-1827, May 2006.
14. Adams, B. M., Bauman, L. E., Bohnhoff, W. J., Dalbey, K. R., Ebeida, M. S., Eddy, J. P., Eldred, M. S., Hough, P. D., Hu, K. T., Jakeman, J. D., Stephens, J. A., Swiler, L. P., Vigil, D. M., and Wildey, T. M., "Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.0 User's Manual," Sandia Technical Report SAND2014-4633.

15. Swiler, L. P., and Wyss, G. D., "A User's Guide to Sandia's Latin Hypercube Sampling Software: LHS Unix Library/Standalone Version," Sandia Technical Report SAND2004-2439.

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

17. Iman, R. L., and Conover, W. J., "Small Sample Sensitivity Analysis Techniques for Computer Models, with an Application to Risk Assessment," *Communications in Statistics: Theory and Methods*, Vol. 9, Issue 17, pp. 1749-1842, 1980.

18. Iman, R. L., and Conover, W. J., "A Distribution-Free Approach to Inducing Rank Correlation Among Input Variables," *Communications in Statistics: Simulation and Computation*, Vol. 11, Issue 3, pp. 311-334, 1982.

19. Owen, A. B., "A central limit theorem for Latin hypercube sampling," *Journal of the Royal Statistical Society: Series B*, Vol. 54, No. 2, pp. 541–551, 1992.

20. Stein, M., "Large sample properties of simulations using Latin hypercube sampling," *Technometrics*, Vol. 29, No. 2, pp. 143–51, 1987.

21. Romero, V. J., "Proposed Replicated Sampling Strategies for Calculating Confidence Intervals on Latin Hypercube Statistics," 43rd Structures, Structural Dynamics, and Materials Conference, Denver, CO, paper AIAA-2002-1276, April 22-25, 2002.

22. Larsen, R. J., and Marx, M. L., *An Introduction to Mathematical Statistics and its Applications*, 2nd ed, Prentice-Hall, Edgewood Cliffs, NJ: 1986.

23. Chakraborti, S., and Li, J., "Confidence Interval Estimation for a Normal Percentile," *The American Statistician*, Vol. 61, No. 4, pp. 331-336, Nov. 2007.

24. Gerstner, T., and Griebel, M., "Numerical integration using sparse grids," *Numerical Algorithms*, 18(3-4), pp. 209–232, 1998.

25. Smolyak, S. A., "Quadrature and interpolation formulas for tensor products of certain classes of functions," *Soviet Mathematics Doklady*, Vol. 4, pp. 240–243, 1963.

26. Hong, H. P., "An efficient point estimate method for probabilistic analysis," *Reliability Engineering and System Safety (RESS)*, Vol. 59, pp. 261–267, 1998.

27. Xiu, D., *Numerical Methods for Stochastic Computations: A Spectral Method Approach*, Princeton University Press, 2010.

28. Ghanem, R., and Spanos, P., *Stochastic Finite Elements: A Spectral Approach*, Springer Verlag, New York, 2002.

29. Constantine, P. G., Eldred, M. S., and Phipps, E. T., "Sparse Pseudospectral Approximation Method," *Computer Methods in Applied Mechanics and Engineering*, Volumes 229-232, pp. 1-12, July 2012.

30. Eldred, M. S., Webster, C. G., and Constantine, P. G., "Evaluation of Non-Intrusive Approaches for Wiener-Askey Generalized Polynomial Chaos," in the proceedings of the 49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference (10th AIAA Non-Deterministic Approaches Conference), Schaumburg, IL, paper AIAA-2008-1892, April 7-10, 2008.

31. Xiu, D., and Karniadakis, G., "The Wiener-Askey polynomial chaos for stochastic differential equations," SIAM Journal on Scientific Computing, Vol. 24, pp. 619-644, 2002.

32. Xiu, D., and Hesthaven, J. S., "High-order collocation methods for differential equations with random inputs," SIAM Journal on Scientific Computing, Vol. 27, No. 3, pp.1118–1139 (electronic), 2005.

33. Nobile, F., Tempone, R., and Webster, C. G., "An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data," SIAM Journal on Numerical Analysis, Vol. 46, No. 5, pp. 2411–2442, 2008.

34. Saltelli, A., Tarantola, S., Campolongo, F., and Ratto, M., *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models,* New York: Wiley, 2004.

35. Sobol', I. M., "Sensitivity analysis for non-linear mathematical models," *Mathematical Modeling and Computational Experiment*, Vol. 1, pp. 407-414, 1993.

36. Saltelli, A., Chan, K., and Scott, E. M., *Sensitivity Analysis,* New York: Wiley, 2000.

37. Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., and Tarantola, S., "Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index," *Computer Physics Communication*, Vol. 181, pp. 259-270, 2010.

38. Weirs, V. G., Kamm, J. R., Swiler, L. P., Ratto, M., Tarantola, S., Adams, B. M., Rider, W. J., and Eldred, M. S., "Sensitivity Analysis Techniques Applied to a System of Hyperbolic Conservation Laws," *Reliability Engineering and System Safety*, Vol. 107, pp. 157-170, November 2012.
doi:10.1016/j.ress.2011.12.008.

39. Sudret, B., "Global sensitivity analysis using polynomial chaos expansions," *Reliability Engineering & System Safety*, Vol. 93, No. 7, pp. 964-979, 2008.

40. Tang, G., Iaccarino, G., and Eldred, M. S., "Global sensitivity analysis for stochastic collocation expansion," in the proceedings of the 12th AIAA Non-Deterministic Approaches Conference, Orlando, FL, paper AIAA-2010-2922, 2010.

41. Storlie, C. B., and Helton, J. C., "Multiple predictor smoothing methods for sensitivity analysis: Description of techniques," *Reliability Engineering and System Safety*, Vol. 93, No. 1, pp. 28-54, 2008.

42. Storlie, C. B., Swiler, L. P., Helton, J. C., and Sallaberry, C. J., "Implementation and evaluation of nonparametric regression procedures for sensitivity analysis of computationally demanding models," *Reliability Engineering and System Safety*, Vol. 94, pp. 1735-1763, 2009.

43. Seber, G. A. F., and Wild, C. J., *Nonlinear Regression*, Wiley & Sons, 2003.

44. Viana, F. A. C., Haftka, R. T., and Steffan , V. Jr., "Multiple surrogates: How cross-validation errors can help us obtain the best predictor," *Structural and Multidisciplinary Optimization*, Vol. 39, No. 4, pp. 439-457, 2009.

45. Martin, J. D., "Computational improvements to estimating kriging metamodel parameters," *Journal of Mechanical Design*, Vol. 131, No. 8, 084501-7, 2009.
doi: 10.1115/1.3151807

46. Rasmussen, C. E., and Williams, C. K. I., *Gaussian Processes for Machine Learning,* MIT Press, 2006.

47. Rosenblatt, M, "Remarks on a multivariate transformation," *Annals of Mathematical Statistics*, Vol. 23, No. 3, pp. 470-472, 1952.

48. Der Kiureghian, A., and Liu, P. L., “Multivariate distribution models with prescribed marginals and covariances,” *Probabilistic Engineering Mechanics*, Vol. 1, Issue 2, pp. 105-112, 1986.

49. Askey, R., and Wilson, J., “Some basic hypergeometric polynomials that generalize Jacobi polynomials,” *Memoirs of the American Mathematical Society*, Vol. 319, Providence, RI, 1985.

50. Eldred, M. S., "Design Under Uncertainty Employing Stochastic Expansion Methods," I*nternational Journal for Uncertainty Quantification*, Vol. 1, No. 2, pp. 119-146, Feb. 2011.

51. Eldred, M. S., Swiler, L. P., and Tang, G., "Mixed Aleatory-Epistemic Uncertainty Quantification with Stochastic Expansions and Optimization-Based Interval Estimation," *Reliability Engineering and System Safety (RESS)*, Vol. 96, No. 9, pp. 1092-1113, Sept. 2011.

52. Najm, H. N., Debusschere, B., Sargsyan, K., Jakeman, J. D., Safta, C., Eldred, M. S., "Sparse Polynomial Representations of High Dimensional Models," conference paper, 10th World Congress on Computational Mechanics, June 2012.

53. Haldar, A., and Mahadevan, S., *Probability, Reliability, and Statistical Methods in Engineering Design,* John Wiley and Sons, 2000.

54. Bichon, B. J., Eldred, M. S., Swiler, L. P., Mahadevan, S., and McFarland, J. M., “Efficient global reliability analysis for nonlinear implicit performance functions,” *AIAA Journal,* Vol. 46, No. 10, pp. 2459–2468, 2008.

On the Calculation of Uncertainty Statistics with Error Bounds for CFD Calculations Containing Random Parameters and Fields

Timothy J. Barth
NASA Ames Research Center, Moffett Field, California 94035

Abstract

This chapter discusses the ongoing development of combined uncertainty and error bound estimates for computational fluid dynamics (CFD) calculations subject to imposed random parameters and random fields. An objective of this work is the construction of computable error bound formulas for output uncertainty statistics that guide CFD practitioners in systematically determining how accurately CFD realizations should be approximated and how accurately uncertainty statistics should be approximated for output quantities of interest. Formal error bounds formulas for moment statistics that properly account for the presence of numerical errors in CFD calculations and numerical quadrature errors in the calculation of moment statistics have been previously presented in [8]. In this past work, hierarchical node-nested dense and sparse tensor product quadratures are used to calculate moment statistics integrals. In the present work, a framework has been developed that exploits the hierarchical structure of these quadratures in order to simplify the calculation of an estimate of the quadrature error needed in error bound formulas. When signed estimates of realization error are available, this signed error may also be used to estimate output quantity of interest probability densities as a means to assess the impact of realization error on these density estimates. Numerical results are presented for CFD problems with uncertainty to demonstrate the capabilities of this framework.

1 Introduction

Computational fluid dynamics (CFD) calculations often contain both numerical error arising from finite-dimensional approximation (e.g., grids, time steps, basis functions) and statistical uncertainty arising from the statistical characterization of model parameters and fields. Although the quantification of CFD numerical errors and the propagation of uncertainties have been individually studied in detail, the interaction of numerical realization errors within uncertainty propagation has not received sufficient attention, and is the focus of this work.

An important task in uncertainty quantification is the forward propagation of uncertainty information to determine the statistical uncertainty of output quantities of interest. This work considers the class of *non-intrusive* uncertainty quantification (UQ) methods that determine the uncertainty of output quantities of interest by performing CFD realizations for specific values of uncertain parameters and uncertain fields. Unfortunately, the accuracy of these computed statistics may be severely compromised by the presence of numerical errors of two types: (1) numerical errors occurring in

CFD realizations and (2) numerical errors occurring in the calculation of output statistics. A unified framework for combined non-intrusive uncertainty and error bound estimation has been developed in [8] such that:

- If the simulation has no uncertainty, then standard *a posteriori* error bound estimates are obtained.
- If the simulation has no numerical error, then standard uncertainty estimates are obtained.
- If the simulation has both uncertainty and numerical error, then uncertainty statistics with error bound estimates are obtained.

Particular attention is given to the calculation of expectation and variance moment statistics. For a function $f(x)$ and probability density $p(x)$, the expectation (mean) is calculated from the integral:

$$E[f] = \int f(x)\, p(x)\, dx \tag{1}$$

and the variance (standard deviation squared) is calculated from the integral:

$$V[f] = \sigma^2[f] = \int (f(x) - E[f])^2\, p(x)\, dx \tag{2}$$

In addition, often there is a keen interest in estimating the output probability density distribution denoted by p_f:

$$Pr[a \leq f \leq b] = \int_a^b p_f(x)\, dx \tag{3}$$

For many problems of interest, the function f can not be exactly evaluated and the required output statistics must be numerically approximated. Thus, error bound estimates for output statistics provide valuable information concerning the quality and reliability of computed statistics. More specifically, computable error bound estimates provide quantitative guidance when performing practical calculations with uncertainty:

- How accurate is a computed output statistic?
- How does realization error affect the accuracy of a computed output statistic?
- How does statistics integral quadrature error affect the accuracy of computed moment statistics?
- To improve the accuracy of computed output statistics, should additional resources be devoted to solving realizations more accurately or to improving the accuracy of computed statistics?

The ability to quantitatively answer these questions, together with error balancing strategies, is a powerful new capability in uncertainty quantification.

2 Background

2.1 The Non-Intrusive Forward Propagation of Uncertainty in PDE Models

The deterministic partial differential equations (PDEs) governing popular formulations of CFD often utilize models containing uncertain parameters that may be mathematically treated as random variables. This development and its impact on computed output quantities of interest is discussed next.

2.1.1 A Deterministic PDE Model

A starting point is the well-posed deterministic system of m conservation laws in d space dimensions that depends on M parameters, $\boldsymbol{\xi} \in \mathbb{R}^M$:

$$\begin{aligned} \partial_t \boldsymbol{u}(\boldsymbol{x},t;\boldsymbol{\xi}) + \sum_{i=1}^{d} \partial_{x_i} \boldsymbol{f}_i(\boldsymbol{u}(\boldsymbol{x},t;\boldsymbol{\xi});\boldsymbol{\xi}) &= 0 \\ \boldsymbol{u}(\boldsymbol{x},0;\boldsymbol{\xi}) &= \boldsymbol{u}_0(\boldsymbol{x};\boldsymbol{\xi}) \end{aligned} \tag{4}$$

with $\boldsymbol{x} \in \Omega \subset \mathbb{R}^d$ and $\boldsymbol{u}, \boldsymbol{f}_i \in \mathbb{R}^m$. The notation $u(\boldsymbol{x},t;\boldsymbol{\xi})$ has been chosen to highlight the dependence of the solution on parameters $\boldsymbol{\xi}$ in the problem. This system, together with suitable spatial boundary conditions (that may also depend on $\boldsymbol{\xi}$), is representative of many conservation law systems arising in computational science such as the equations of compressible flow utilized in CFD.

2.1.2 A Random Variable PDE Model

Let (Θ, Σ, P) denote the probability space of event outcomes, σ-algebra, and probability measure, respectively. Suppose the parameters $\boldsymbol{\xi}$ are now random variables depending on the random events $\omega \in \Theta$ that satisfy given probability laws. A random variable form of the conservation law system is now given by:

$$\begin{aligned} \partial_t \boldsymbol{u}(\boldsymbol{x},t,\omega;\boldsymbol{\xi}(\omega)) + \sum_{i=1}^{d} \partial_{x_i} \boldsymbol{f}_i(\boldsymbol{u}(\boldsymbol{x},t,\omega;\boldsymbol{\xi}(\omega)),\boldsymbol{\xi}(\omega)) &= 0 \\ \boldsymbol{u}(\boldsymbol{x},0,\omega;\boldsymbol{\xi}(\omega)) &= \boldsymbol{u}_0(\boldsymbol{x};\boldsymbol{\xi}(\omega)) \end{aligned} \tag{5}$$

The statistical behavior of $\boldsymbol{\xi}(\omega)$ is characterized here by a probability density $p_{\boldsymbol{\xi}}(\boldsymbol{\xi})$ such that $dP(\omega) = p_{\boldsymbol{\xi}}(\boldsymbol{\xi}) d\boldsymbol{\xi}(\omega)$. For simplicity, it is assumed in later examples that the random variables are independent so that the probability density is of product form:

$$p_{\boldsymbol{\xi}}(\boldsymbol{\xi}) = \prod_{i=1}^{M} p_{\xi_i}(\xi_i) \tag{6}$$

When the number of random variable parameters is large, a complete representation of the solution $\boldsymbol{u}(\boldsymbol{x},t,\omega;\boldsymbol{\xi}(\omega))$ is a high-dimensional object. Consequently, finding the solution to this problem either analytically or numerically can be cost prohibitive.

2.2 Output Quantities of Interest

For many practical problems, rather than explicitly representing the entire random variable solution, $\boldsymbol{u}(\boldsymbol{x},t,\omega;\boldsymbol{\xi}(\omega))$, there is often interest in quantifying uncertainty statistics of solution-derived output quantities of interest (QOI) denoted here by $J[u(\boldsymbol{x},t,\omega;\boldsymbol{\xi}(\omega))]$. Output QOIs may include random variable functionals, graphs, and fields. Recall that in forward uncertainty propagation, the probability laws associated with random variable parameters $\boldsymbol{\xi}$ are *given* and the primary task at hand is to characterize the probability law associated with output quantities of interest, $J[u(\boldsymbol{x},t,\omega;\boldsymbol{\xi}(\omega))]$. This probability law may be characterized either:

- Incompletely by the calculation of moment statistics such as expectation and variance for the QOI.
- Completely by the calculation of the probability density function for the QOI.

For problems with many sources of uncertainty, $J[u(\boldsymbol{x},t,\omega;\boldsymbol{\xi}(\omega))]$ is still a high-dimensional object so the estimation of these probability law characterizations is cost prohibitive. Rather than directly constructing high-dimensional random variable representations of the output quantities of interest, non-intrusive uncertainty propagation methods calculate a finite set of N decoupled deterministic numerical realizations of the PDE model, $u_h(\boldsymbol{x},t;\boldsymbol{\xi})$, for distinct parameter values:

$$\{\boldsymbol{\xi}^{(1)},\ldots,\boldsymbol{\xi}^{(N)}\} \tag{7}$$

yielding solution realizations

$$\{u_h(\boldsymbol{x},t;\boldsymbol{\xi}^{(1)}),\ldots,u_h(\boldsymbol{x},t;\boldsymbol{\xi}^{(N)})\} \tag{8}$$

and output quantities of interest from realizations

$$\{J[u_h(\boldsymbol{x},t;\boldsymbol{\xi}^{(1)})],\ldots,J[u_h(\boldsymbol{x},t;\boldsymbol{\xi}^{(N)})]\} \tag{9}$$

For the uncertainty propagation methods considered here, the values $\boldsymbol{\xi}^{(i)}$ are chosen as follows:

- At quadrature points when moment statistics of the QOI are sought, see Section 2.3.
- At interpolation points forming a response surface when a QOI probability density approximation is sought, see Section 5.

This approach is referred to as "non-intrusive" uncertainty propagation since it does not require modifications to an existing numerical method for calculating realizations other than the ability to change parameter values for each realization.

2.3 Numerical Quadratures for Moment Statistics

Let $I[f]$ denote a definite integral in d dimensions:

$$I[f] = \int_{[0,1]^d} f(x)\,dx \tag{10}$$

Recall that statistics integrals such as expectation and variance may be transformed to this form when the probability density function is an independent (separable) product form in d dimensions:

$$p(y) = p_{y_1}(y_1)p_{y_2}(y_2)\cdots p_{y_d}(y_d)$$

In this case, a transformed vector variable x can be found such that $dx = p(y)dy$, using the cumulative probability distribution function associated with each p_{y_i} for $i = 1,\ldots,d$. This permits a transformation of the integral to the unit cube, i.e.:

$$I[f] = \int_D f(y)\, p(y)\, dy = \int_{[0,1]^d} f(y(x))\, dx \tag{11}$$

Let $Q_N[f]$ denote an N-point numerical quadrature approximation to $I[f]$ with weights w_i and evaluation points $\boldsymbol{\xi}^{(i)}$:

$$Q_N[f] = \sum_{i=1}^{N} w_i f(\boldsymbol{\xi}^{(i)}) \tag{12}$$

with numerical quadrature error denoted by $R_N[f]$, i.e.:

$$R_N[f] = I[f] - Q_N[f] \tag{13}$$

Efficient quadratures $Q_N[\cdot]$ are considered in Section 2.5 that also provide straightforward estimates of the quadrature error $R_N[\cdot]$. These quadratures with quadrature error estimates are necessary ingredients in error bound formulas for moment statistics.

2.4 A Survey of Quadrature Methods

Several candidate methods exist for numerical quadrature of the integral (10). The locations of integrand evaluation for these methods are depicted in Figure 1 and quadrature error properties are

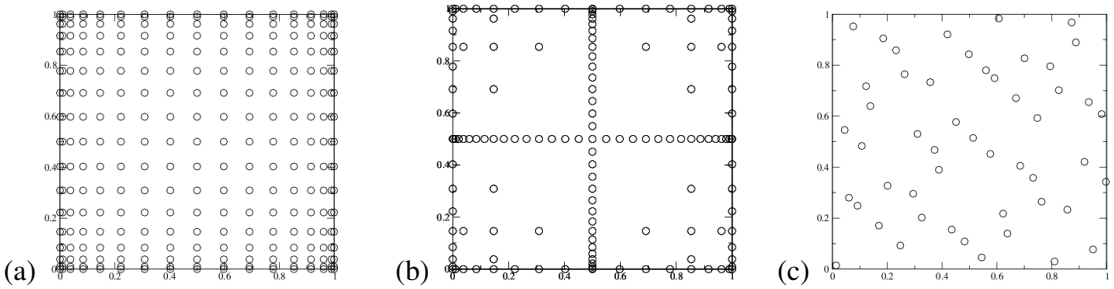


Figure 1. Quadrature point locations for (a) dense tensor product, (b) sparse tensor product, and (c) random sampling.

summarized in Table 1. For a small number of dimensions, dense tensor product methods based on global approximation are often used. More specifically, efficient dense tensor product quadrature evaluation based on the hierarchical node-nested Clenshaw-Curtis [14] and Gauss-Patterson [29]

Table 1. Summary of d-dimensional quadrature methods utilizing N evaluations.

Quadrature Method	Quadrature Error	Requirements
Global Dense Tensor Product	$\mathscr{O}(N^{-r/d})$	r bounded derivatives
Global Sparse Tensor Product	$\mathscr{O}(N^{-r}(\log N)^{(d-1)(r+1)})$	r bounded mixed derivatives
Piecewise Polynomial Tensor Product	$\mathscr{O}(N^{-(p+1)/d})$	p order local polynomials
Monte Carlo Sampling	$\mathscr{O}(N^{-1/2})$	bounded variance
Quasi-Monte Carlo Sampling	$\mathscr{O}(N^{-1}(\log N)^d)$	bounded variance

formulas are particularly useful in the present work because these quadratures also provide computable quadrature error estimates at little added cost. These quadratures are discussed in detail in Section 2.5. As the number of dimensions increases, dense tensor product quadratures become prohibitively expensive, and so sparse tensor product quadratures based on Smolyak [38] sparse grids are often then used instead. These sparse tensor product quadratures may also utilize Clenshaw-Curtis and Gauss-Patterson quadratures [27, 18] and thus retain the hierarchical node-nested structure needed for quadrature error estimates. As the number of dimensions increases further, sampling methods based on Monte Carlo random sampling [25] and low discrepancy quasi-random sampling [26] are preferred. Methods such as Monte Carlo random sampling have a well-known computable quadrature error estimate, $R_N[f] = \sqrt{V[f]/N}$. Thus, the error bound formulas described here apply without major modification to these methods as well. We now consider dense and sparse product quadratures and defer a discussion of the sampling methods to a separate work.

2.5 Hierarchical Node-Nested Quadratures with Quadrature Error Estimates

Hierarchical node-nested quadratures permit the efficient approximation of integrals while also providing a quadrature error estimate.

Let L denote a numbering of levels, N_L the number of univariate quadrature evaluation points at level L with $N_L < N_{L'}$ if $L < L'$, and $Q_L^{(1)}[\cdot]$ a quadrature at level L, i.e.:

$$Q_L^{(1)}[f] \equiv \sum_{i=1}^{N_L} w_i^{(j)} f(x_i^{(j)})$$

Of particular interest are hierarchical multi-level quadratures that satisfy the following:

$$\lim_{L\to\infty} \sup \left(N_L^r \| I^{(1)}[f] - Q_L^{(1)}[f] \| \right) < \infty \ , \quad \forall r \in \mathbb{N} \tag{14}$$

whenever f possesses r bounded derivatives. Satisfaction of (14) implies the quadrature error estimate for level L:

$$\| I^{(1)}[f] - Q_L^{(1)}[f] \| = \mathscr{O}(N_L^{-r}) \tag{15}$$

Brass [13] has shown that multi-level positive-weighted quadrature formulas that are exact for polynomials of degree strictly less than N_L satisfy Eq. (14) and the quadrature error estimate Eq. (15). Two particular multi-level quadrature formulas that satisfy these requirements are the Clenshaw-Curtis [14] quadrature formula and the Gauss-Patterson [29] quadrature formula. The node-nested

hierarchy in these two methods makes them very efficient; i.e., all quadrature evaluation points in level L are contained in level $L+1$. Evaluation points for these quadrature formulas are shown in Figure 2. Level L of the Clenshaw-Curtis quadrature requires $2^{L-1}+1$ evaluations and integrates

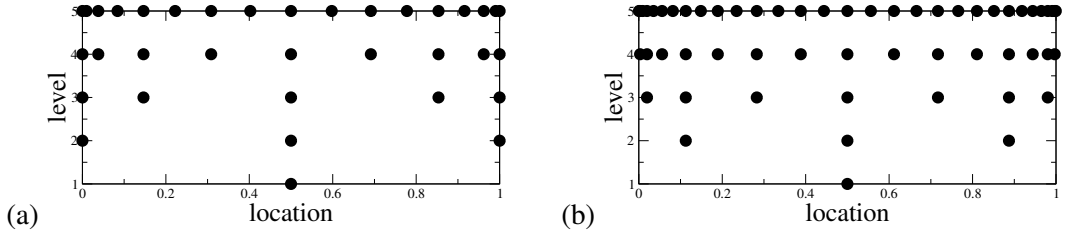


Figure 2. Node-nested quadratures point locations: (a) Clenshaw-Curtis quadrature with $N_{level} = 2^{(level-1)} + 1$, and (b) Gauss-Patterson with $N_{level} = 2^{level} - 1$ for $level > 1$ and $N_1 = 1$.

2^{L-1} polynomials exactly for $L \geq 2$. Level L of the Gauss-Patterson quadrature requires $2^L - 1$ evaluations for $L > 1$ and integrates $3 \cdot 2^{L-1} - 1$ degree polynomials exactly for $L \geq 2$. Because both quadratures satisfy the Brass conditions, they have a quadrature error estimate given by:

$$\|I^{(1)}[f] - Q_L^{(1)}[f]\| = \mathscr{O}(2^{-rL}) \tag{16}$$

2.5.1 Dense Tensor Product Clenshaw-Curtis and Gauss-Patterson Quadrature with Quadrature Error Estimate

Dense tensor product quadrature takes the form:

$$\begin{aligned} Q_L^{(d)}[f] &\equiv \left(Q_{L_1}^{(1)} \otimes \cdots \otimes Q_{L_d}^{(1)}\right)[f] \\ &= \sum_{i_1=1}^{N_{L_1}} \cdots \sum_{i_d=1}^{N_{L_d}} w_{i_1}^{(L_1)} \cdot w_{i_d}^{(L_d)} f(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d) \end{aligned} \tag{17}$$

with the total number of evaluations given by $N = \prod_{j=1}^{d} N_{L_j}$. When $Q_{L_j}^{(1)}$ corresponds to either the univariate Clenshaw-Curtis or the univariate Gauss-Patterson quadrature and $L_1 = L_2 = \cdots = L_d = L$, then the total number of evaluations is $\mathscr{O}(2^{dL})$ and the quadrature error reduces to (see for example [37]):

$$|I^{(d)}[f] - Q_L^{(d)}[f]| = \mathscr{O}(N^{-r/d}) = \mathscr{O}(2^{-rL})$$

Assuming that the unknown leading $\mathscr{O}(\cdot)$ constant does not oscillate when L is varied and remains approximately constant even for modest values of L, the following computable quadrature error formula is readily obtained:

$$R_L^{(d)}[f] \equiv I^{(d)}[f] - Q_L^{(d)}[f] \approx \frac{1}{2^r - 1}(Q_L^{(d)}[f] - Q_{L-1}^{(d)}[f]) \tag{18}$$

with

$$2^r = \frac{Q_{L-1}^{(d)}[f] - Q_{L-2}^{(d)}[f]}{Q_L^{(d)}[f] - Q_{L-1}^{(d)}[f]} \tag{19}$$

Although this is a parameter-free three-level quadrature error estimate, the performance of (19) can be unreliable when the underlying data lacks regularity. For this reason, lower and upper bounds on the smoothness parameter r are enforced, i.e., r is constrained to an interval:

$$r_{min} \leq r \leq r_{max} \tag{20}$$

with values $r_{min} = 1$ and $r_{max} = 6$ used in practical applications. Observe that if $r_{min} = r_{max}$, then the quadrature estimate (18) only depends on the two finest levels and the coarsest level is not used in the quadrature error estimate.

2.5.2 Sparse Tensor Product Clenshaw-Curtis and Gauss-Patterson Quadrature with Quadrature Error Estimate

Unfortunately, dense tensor product quadratures grow exponentially in complexity with respect to the number of dimensions. A quadrature of just two points in d dimensions requires:

$$N^{dense} = \mathscr{O}(2^d) \qquad \text{(dense product quadratures)} \tag{21}$$

evaluations. In contrast, complete polynomials of degree P in d dimensions require only the following number of evaluations:

$$N^{poly} = \binom{P+d}{d} \approx \frac{d^P}{P!} \qquad \text{(complete polynomials)} \tag{22}$$

This large gap indicates that dense product quadratures contain many unneeded evaluations. The sparse tensor product quadrature of Smolyak [38] offers a dramatic reduction in the number of evaluations required for a given precision P and dimension d when compared to dense product quadrature.

Sparse tensor product formulas are compactly written in terms of a multi-index, $\boldsymbol{i} \in \mathbb{N}^d$, so that a given product rule may be written as $Q^{(1)}_{N_{L_1}} \otimes \cdots \otimes Q^{(1)}_{N_{L_d}}$ with product level $|\boldsymbol{i}| = \sum_{j=1}^d i_j$. Using this compact notation, Smolyak sparse grid quadratures with maximum level L in d dimensions have the form:

$$Q^{(d)}_L[f] = \sum_{L-d+1 \leq |\boldsymbol{i}| \leq L} (-1)^{L-|\boldsymbol{i}|} \binom{d-1}{L-|\boldsymbol{i}|} (Q^{(1)}_{N_{L_1}} \otimes \cdots \otimes Q^{(1)}_{N_{L_d}})[f] \tag{23}$$

Novak and Ritter [27] have analyzed sparse tensor product quadratures using hierarchical multi-level Clenshaw-Curtis and Gauss-Patterson formulas. Sparse grid quadratures attain a polynomial precision P equal to $2L+1$ and require $\mathscr{O}\left(\frac{(2d)^P}{P!}\right)$ evaluations. Table 2 gives the number of quadrature evaluations required for sparse and dense forms of Clenshaw-Curtis and Gauss-Patterson quadrature for dimensions $d = 4$ and $d = 8$ and levels $L = 2,3,4,5$. The sparse tensor product Clenshaw-Curtis quadrature is a vast improvement in complexity over the dense product counterpart and differs from the use of complete polynomials by a factor 2^P.

The task of finding a computable quadrature estimate begins with a quadrature error estimate given by Wasilkowski and Woźniakowski [42] for $L \geq d$:

$$|I^{(d)}[f] - Q^{(d)}_L[f]| = \mathscr{O}(L^{(d-1)(r+1)} 2^{-rL}) \tag{24}$$

Table 2. Number of evaluations required for dense and sparse forms of Clenshaw-Curtis (CC) and Gauss-Patterson (GP) quadrature for dimensions $d = 4$ and $d = 8$ at levels $L = 2,3,4,5$.

L	Dense d=4 N(CC)	Dense d=4 N(GP)	Sparse d=4 N(CC)	Sparse d=4 N(GP)	Dense d=8 N(CC)	Dense d=8 N(GP)	Sparse d=8 N(CC)	Sparse d=8 N(GP)
2	81	81	9	9	6561	6561	17	17
3	625	2401	41	49	390625	5764801	145	161
4	6561	50625	137	209	$\mathscr{O}(10^8)$	$\mathscr{O}(10^{10})$	849	1121
5	83521	923521	401	769	$\mathscr{O}(10^{10})$	$\mathscr{O}(10^{12})$	3937	6401

Again assuming that the unknown leading $\mathscr{O}(\cdot)$ constant does not oscillate in sign when L is varied and remains approximately constant even for modest values of $L \geq d$, the following computable quadrature error estimate is readily obtained:

$$R_L^{(d)}[f] \equiv I^{(d)}[f] - Q_L^{(d)}[f] \approx \frac{1}{\left(\frac{L-1}{L}\right)^{(d-1)(r+1)} 2^r - 1} (Q_L^{(d)}[f] - Q_{L-1}^{(d)}[f]) \tag{25}$$

where r is calculated from the given parameters d and L by solving:

$$\frac{Q_{L-1}^{(d)}[f] - Q_{L-2}^{(d)}[f]}{Q_L^{(d)}[f] - Q_{L-1}^{(d)}[f]} = \frac{2^{2r} g(r;L-2,d) - 2^r g(r;L-1,d)}{2^r g(r;L-1,d) - g(r;L,d)} \tag{26}$$

with $g(r;L,d) \equiv L^{(d-1)(r+1)}$. Equation (25) reveals a strong dependence on the smoothness parameter r. In sharp contrast to the dense tensor products, the right-hand-side denominator term of the sparse error estimate may not be positive (or bounded) when the smoothness parameter r is too small for certain values of L and d. For example, when $L = 5$ and $d = 3$, then r must be greater than 1.81 for the denominator to be positive. Following the same strategy taken for the dense tensor product quadrature error estimate, the smoothness parameter r is constrained to an interval:

$$r_{min} \leq r \leq r_{max} \tag{27}$$

Depending on L and d, the lower bound may need to be larger than that used in the dense tensor product case. Note that for sparse tensor product quadrature calculations presented in Section 6.2 that correspond to $L = 4$ and $d = 2$, the formula (25) is well defined for all $r > 0.7$, so a value of $r_{min} = 1$ has been imposed.

2.5.3 Hybrid Quadrature (HYGAP) Using Piecewise Polynomial and Clenshaw-Curtis Quadrature

Hierarchical multi-level global quadratures such as Clenshaw-Curtis and Gauss-Patterson quadrature in dense tensor product form have $\mathscr{O}(2^{-rL})$ accuracy using L levels. The accuracy depends fundamentally on smooth integrands with r bounded derivatives. Sparse tensor product quadratures with $\mathscr{O}(L^{(d-1)(r+1)}2^{-rL})$ accuracy are even more demanding in requiring integrands with r bounded mixed derivatives. Unfortunately, output quantities of interest for hyperbolic problems may not have a smooth dependence on random parameters. Instead, one frequently encounters statistics

integrands that are *piecewise smooth* with a finite number of discontinuities in random variable dimensions. This observation motivated the construction of a new hybrid algorithm [7, 8] tailored for piecewise smooth integrand data. The basic idea behind the Hybrid Global Adaptive Polynomial (HYGAP) algorithm is relatively straightforward. Construct statistics quadratures in multiple dimensions using hierarchical multi-level Clenshaw-Curtis or Gauss-Patterson dense tensor products. Then for each dimension, examine integrand data to determine smoothness. When the integrand data is smooth in a dimension, use hierarchical multi-level Clenshaw-Curtis or Gauss-Patterson quadrature. When the integrand data is not smooth, construct a piecewise cubic polynomial interpolant in that dimension using data at the Clenshaw-Curtis or Gauss-Patterson points and integrate it using piecewise quadrature.

HYGAP(L)[8] Algorithm:

Step 1. Calculate quadrature point locations of an L-level Clenshaw-Curtis or Gauss-Patterson global quadrature.

Step 2. Evaluate output quantities of interest for those quadrature point values.

Step 3. Determine the smoothness of data in each dimension by estimating a one-dimensional Sobolev semi-norm measure using one-dimensional WENO piecewise polynomial approximations; see Jiang and Shu [22].

Step 4a. (smooth data) If the integrand data is smooth in a given dimension, then compute the moment statistics in that dimension using a conventional Clenshaw-Curtis or Gauss-Patterson quadrature formula.

Step 4b. (non-smooth data) If the integrand data is not smooth in a given dimension, then interpolate the integrand data at quadrature point locations using adaptive stencil piecewise polynomials. Further improve the piecewise polynomial approximation by applying a subscale resolution model [8]. Compute moment statistics by integrating the interpolated integrand data using a local piecewise quadrature.

Step 5. Compute the multi-dimensional quadrature using dense tensor products.

To demonstrate the dramatic improvement using HYGAP approximation, in [8] a random variable form of Burgers' equation with uncertain initial data was considered:

$$\frac{\partial}{\partial t}u(x,t,\omega)+\frac{\partial}{\partial x}(u^2(x,t,\omega)/2)=0 \tag{28}$$

with sinusoidal initial data with phase uncertainty:

$$u(x,0,\omega)=\sin(2\pi(x+g(\xi(\omega)))) \tag{29}$$

for $g:\mathbb{R}\mapsto\mathbb{R}$ a smooth function and $\xi(\omega)$ a random variable. The exact solution to this problem is readily constructed. At some later time, a discontinuity forms for each value of $\xi(\omega)$ so the global solution has a discontinuity that traverses obliquely through both physical and random variable dimensions as illustrated in Figure 3. The presence of discontinuities in random variable dimensions has enormous consequences in the performance of many classical techniques in UQ

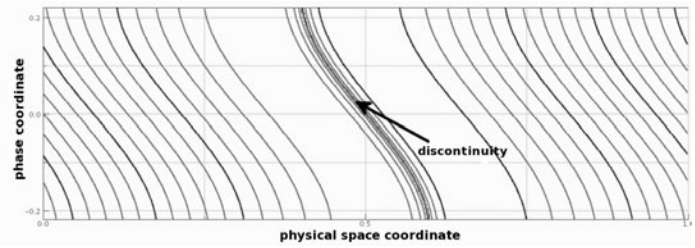


Figure 3. Solution contours for the Burgers' equation problem with phase uncertain sinusoidal initial data at time $t = .35$ with $g(\xi) = \frac{1}{10}\sin(2\pi\xi)$.

using dense and sparse tensor product quadrature. Figure 4 shows solution statistics for this Burgers' equation problem with imposed phase uncertainty satisfying a uniform probability density law, Uniform $[-.25, .25]$, with statistics approximated using Clenshaw-Curtis global quadrature. Note that similar results are obtained using a standard stochastic collocation method [39, 2]. The spuri-

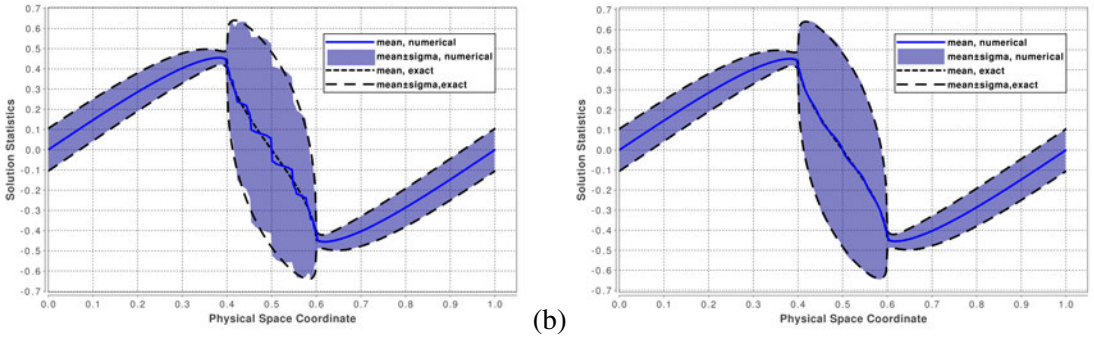


Figure 4. Expectation and standard deviation envelopes approximated from nine realizations using (a) stochastic collocation and (b) Clenshaw-Curtis HGAP approximation for the Burgers' equation problem (28) with phase uncertain initial data (29) at time $t = .35$.

ous oscillations are a result of using a global quadrature that spans a piecewise smooth integrand.

For smooth integrand data in all dimensions, applying the HYGAP algorithm at three successive levels L, $L-1$, and $L-2$ is sufficient to estimate the quadrature error using Eq. (18). When the integrand data is not smooth, the task of estimating quadrature error is difficult and not likely to be very sharp. A practical strategy is to assume a quadrature error formula of the form of Eq. (18) and then *prescribe* the value of smoothness parameter r. Keep in mind that when statistics are calculated at many spatial-temporal locations, the number of locations where r is prescribed in this way may be relatively small. In calculations shown here, $r = 1$ has been used for non-smooth integrands in the HYGAP method and satisfactory results were obtained.

2.6 Estimating Eealization Error in CFD Calculations

The error bound formulas for output moment statistics developed later in Section 3 require an estimate of the realization error:

$$\varepsilon_h \equiv J[u(\boldsymbol{x},t;\boldsymbol{\xi})] - J[u_h(\boldsymbol{x},t;\boldsymbol{\xi})] \tag{30}$$

for output quantities of interest. There are a wide variety of techniques for estimating this realization error. These include: extrapolation techniques such as Richardson's extrapolation [32] and Aitken's extrapolation [1]; patch postprocessing techniques [43, 16] that exploit superconvergence behavior of solutions; and error representation of functionals via dual problems [12, 17, 30, 23, 9, 10], among others. The numerical results of Section 6 utilize the Aitken extrapolation method or the error representation via dual problem method so these algorithms are briefly discussed.

2.6.1 Richardson and Aitken Extrapolations

Both the Richardson and Aitken extrapolation methods are sequence extrapolation/acceleration algorithms. Given a multi-level mesh hierarchy $\{\boldsymbol{u}_h, \boldsymbol{u}_{2h}, \boldsymbol{u}_{4h}, \dots\}$ where h denotes a mesh spacing parameter, assume that the output realization QOI has polynomial convergence properties that depend on the parameter h and a rate q, i.e.:

$$J(\boldsymbol{u}) - J(\boldsymbol{u}_h) = C\,h^q + \text{higher order terms} \tag{31}$$

for a constant C that does not depend on h or q. Evaluating this formula at three levels permits the elimination of the unknown constant C and the rate q, i.e.:

$$J(\boldsymbol{u}) - J(\boldsymbol{u}_h) \approx \frac{1}{2^q - 1}(J(\boldsymbol{u}_h - J(\boldsymbol{u}_{2h})) \tag{32}$$

Depending on how q is determined, the Aitken and Richardson extrapolation formulas can be obtained.

- If q is computed from
$$2^q = \frac{J(\boldsymbol{u}_{2h}) - J(\boldsymbol{u}_{4h})}{J(\boldsymbol{u}_h) - J(\boldsymbol{u}_{2h})} \tag{33}$$
then the extrapolation is referred to as Aitken's extrapolation [1] (although Aitken's extrapolation is often not written this way).
- If q is specified *a priori*, then the extrapolation is referred to as Richardson's extrapolation [32] and the extrapolation formula only uses 2 levels.

Equation (33) can be unreliable when the mesh sequence is not sufficiently resolved. Consequently, software implementations may impose lower and upper bounds on q:

$$q_{min} \le q \le q_{max} \tag{34}$$

Results presented in Section 6 have used $q_{min} = 1$ and $q_{max} = 6$ in realization QOI error estimates obtained using Aitken's extrapolation.

2.6.2 A Posteriori Error Estimation of Functionals Via Dual Problem

When an output quantity of interest $J(\boldsymbol{u}_h(\cdot;\boldsymbol{\xi});\boldsymbol{\xi})$ is a functional, the task of estimating the finite-dimensional approximation error is greatly simplified using the *a posteriori* error estimation theory developed by Eriksson *et al.* [17] and Becker and Rannacher [12]. This theory has been applied to finite-element methods [12, 17, 30, 23, 20] as well as Godunov finite-volume methods [10, 6].

The abstract *a posteriori* error estimation theory for Galerkin approximations of the model PDE in Eq. (4) consists of the following steps:

1. Solve the primal numerical problem using finite-dimensional approximation spaces $\mathscr{V}^h$. In the abstract formulation, $F(\boldsymbol{v}_h) : \mathscr{V}^h \mapsto \mathbb{R}^m$ is a forcing term (equal to zero in the present calculations) with boundary conditions assumed here to be enforced weakly via fluxes.

   Primal numerical problem: Find $\boldsymbol{v}_h \in \mathscr{V}^h$ such that:

   $$B(\boldsymbol{v}_h, \boldsymbol{w}_h) = F(\boldsymbol{w}_h) \quad , \quad \forall\, \boldsymbol{w}_h \in \mathscr{V}^h$$

2. Solve the mean-value linearized auxiliary dual problem $\overline{B}(\cdot,\cdot)$ using infinite-dimensional spaces $\mathscr{V}$ given a mean-value linearized functional $\overline{J}$.

   Linearized auxiliary dual problem: Find $\Phi \in \mathscr{V}$ such that:

   $$\overline{B}(\boldsymbol{w}, \Phi) = \overline{J}(\boldsymbol{w}) \quad , \quad \forall\, \boldsymbol{w} \in \mathscr{V}$$

3. Compute the error in a functional using the error representation formula.

   Error representation: Let π_h denote any projection into the Galerkin test space (e.g., L_2 projection, interpolation), the functional error is given by:

   $$J(\boldsymbol{u}) - J(\boldsymbol{u}_h) = F(\Phi - \pi_h\Phi) - B(\boldsymbol{u}_h, \Phi - \pi_h\Phi) \tag{35}$$

The mean-value linearization given by the error estimation theory requires knowledge of the infinite-dimensional primal solution. In addition, solutions of the dual problem are posed in infinite-dimensional spaces. These solutions are generally not available and so must be approximated. In the present computations, the mean-value linearization has been replaced by the Jacobian (tangent) linearization at the numerical solution state and the dual problem has been solved numerically using an approximation space that is one polynomial degree higher than the approximation space of the primal numerical problem. This permits the right-hand-side of Eq. (35) to be estimated.

The error representation formula (35) may be used to estimate the functional error. However, if the estimated error is too large, the formula does not provide information about how the mesh or approximation space should be modified to further reduce the error. Element-wise decomposition of the error representation formula provides a pathway for deriving element refinement indicators and a systematic means for reducing the error by refining the elements. To simplify the notation, let $Q^n = K \times I^n$ denote a space-time slab prism for a time interval, $I^n = [t^n, t^{n+1}]$. Observe that

without further approximation, the error representation formula can be written as a sum over spatial elements and N_T time intervals:

$$|J(\boldsymbol{u}) - J(\boldsymbol{u}_h)| = \left| \sum_{n=0}^{N_T-1} \sum_{Q^n} F_{Q^n}(\Phi - \pi_h\Phi) - B_{Q^n}(\boldsymbol{v}_h, \Phi - \pi_h\Phi) \right| \tag{36}$$

where $B_{Q^n}(\cdot,\cdot)$ and $F_{Q^n}(\cdot)$ are the restriction of $B(\cdot,\cdot)$ and $F(\cdot)$ to a single space-time element. Application of the generalized triangle inequality provides a localized estimate of the contribution of each space-time element to the total error in the functional:

$$|J(\boldsymbol{u}) - J(\boldsymbol{u}_h)| \leq \sum_{n=0}^{N_T-1} \sum_{Q^n} \underbrace{|F_{Q^n}(\Phi - \pi_h\Phi) - B_{Q^n}(\boldsymbol{v}_h, \Phi - \pi_h\Phi)|}_{\text{refinement indicator, } \eta(Q^n)} \tag{37}$$

These localized estimates serve as *refinement indicators* for mesh adaptivity. A commonly used strategy in mesh adaptivity, which is adopted here, is to refine a fixed fraction of element indicators $\eta(Q^n)$ that are too large and then coarsen a fixed fraction of element indicators that are too small.

3 Combined Uncertainty and Error Estimates

Recall from Section 2.2 that non-intrusive uncertainty propagation methods compute a finite number of realization output quantities of interest (QOI) for specific parameter values:

$$\{J_h^{(1)}, J_h^{(2)}, \cdots, J_h^{(N)}\} \tag{38}$$

In this formula, we have adopted the shorthand notation, $J_h^{(i)} \equiv J[u_h(\boldsymbol{x},t;\boldsymbol{\xi}^{(i)})]$. The output QOIs are calculated using a numerical method so they are finite-dimensional approximations of the exact output QOIs:

$$\{J^{(1)}, J^{(2)}, \cdots, J^{(N)}\} \tag{39}$$

where $J^{(i)} \equiv J[u(\boldsymbol{x},t;\boldsymbol{\xi}^{(i)})]$. For each realization QOI, we can assign a realization error:

$$\varepsilon_h^{(i)} \equiv J^{(i)} - J_h^{(i)} \tag{40}$$

Methods for estimating $\varepsilon_h^{(i)}$ are given in Section 2.6.

Next, recall from Section 2.3 that evaluation of moment statistics can be efficiently performed using hierarchical multi-level quadratures in dense and sparse tensor product form. For example, when applied to output QOIs, the expectation is given by:

$$E[J_h] = Q_L E[J_h] + R_L E[J_h] \tag{41}$$

with quadrature of the form

$$Q_L E[J_h] = \sum_{i=1}^{N_L} w_i J_h^{(i)} \tag{42}$$

and quadrature error $R_L E[J(\boldsymbol{u}_h(\boldsymbol{x},t;\boldsymbol{\xi}))]$ estimated from Eq. (18) for dense tensor product quadratures and Eq. (25) for sparse tensor product quadratures.

The ability to estimate quadrature errors and realization errors provides the components needed to construct error bound estimates for moment statistics.

3.1 Error Estimates for Moment Statistics Given Signed Realization and Quadrature Errors

Given the realization QOI error ε_h, a quadrature $Q_L E[\cdot]$, and quadrature error $R_L E[\cdot]$, the error in expectation can be calculated from the following formula:

$$E[J] - Q_L E[J_h] = Q_L E[\varepsilon_h] + R_L E[\varepsilon_h] + R_L E[J_h] \tag{43}$$

This formula expresses the difference between the exact expectation of the exact solution and a numerically approximated expectation of the approximated realization. This formula is purely motivational. In a realistic setting, the realization error may only be approximately known, as denoted by $\tilde{\varepsilon}_h$. Let $\tilde{J} \equiv J_h + \tilde{\varepsilon}_h$, the expectation and variance error can be estimated as follows:

• Expectation error estimate (Type I):

$$\begin{aligned} E[J] - Q_L E[J_h] &\approx E[\tilde{J}] - Q_L E[J_h] \\ &= Q_L E[\tilde{J}] + R_L E[\tilde{J}] - Q_L E[J_h] \end{aligned} \tag{44}$$

• Variance error estimate (Type I):

$$\begin{aligned} V[J] - Q_L V[J_h] &\approx E[\tilde{J}^2] - (E[\tilde{J}])^2 - (Q_L E[J_h^2] - (Q_L E[J_h])^2) \\ &= Q_L E[\tilde{J}^2] + R_L E[\tilde{J}^2] - (Q_L E[\tilde{J}] + R_L E[\tilde{J}])^2 - (Q_L E[J_h^2] - Q_L^2 E[J_h]) \end{aligned} \tag{45}$$

with equality in both formulas if $\tilde{\varepsilon}_h = \varepsilon_h$. In deriving the variance error, we have used the identity $V[J] = E[J^2] - E^2[J]$ and $Q_L V[J] = Q_L E[J^2] - Q_L E^2[J]$. Note that this latter identity does not strictly hold with equality if the quadrature weights depend on the integrand data, as in the HYGAP approximation.

3.2 Error Bound Estimates for Moment Statistics Given Realization and Quadrature Error Magnitudes

Error bound formulas for expectation and variance are derived in [8] assuming that only magnitudes of the quadrature error, $|R_N E[\cdot]|$, and magnitudes of the realization error, $|\varepsilon_h|$, are available. This situation arises often in practice. For example, the quadrature error estimate for Monte Carlo sampling given by:

$$|R_N E[J_h]| \approx \sqrt{V[J_h]/N} \tag{46}$$

only provides the magnitude of the quadrature error. As another example, the error estimate of Eriksson *et al.* [17] for functional output QOIs only estimates the magnitude. This estimate (which avoids the explicit calculation of a dual (adjoint) problem) is:

$$|\varepsilon_h| \equiv |J(u) - J(u_h)| \leq C_{int} C_{stab} \|h^s r_h\|, \quad s > 0 \tag{47}$$

where r_h is the discretization residual, C_{int} and C_{stab} are interpolation and stability constants, and h is the mesh spacing.

Assuming that only QOI realization error and quadrature error magnitudes are known, the following error bound formulas are given in [8] for expectation and variance:

• Expectation error bound (Type II):

$$\begin{aligned} |E[J(\boldsymbol{u})] - Q_L E[J(\boldsymbol{u}_h)]| &= |Q_L E[\boldsymbol{\varepsilon}_h]| + R_L E[\boldsymbol{\varepsilon}_h] + R_L E[J(\boldsymbol{u}_h)]| \\ &\leq |Q_L E[|\boldsymbol{\varepsilon}_h|]| + |R_L E[|\boldsymbol{\varepsilon}_h|]| + |R_L E[J(\boldsymbol{u}_h)]| \end{aligned} \tag{48}$$

• Variance error bound (Type II):

$$\begin{aligned} |V[J(\boldsymbol{u})] - Q_L V[J(\boldsymbol{u}_h)]| \leq\ & 2\,(|Q_L E[|\boldsymbol{\varepsilon}_h|^2]| + |R_L E[|\boldsymbol{\varepsilon}_h|^2]|)^{1/2} \\ & \times (|Q_L V[J(\boldsymbol{u}_h)]| + |R_L V[J(\boldsymbol{u}_h)]|)^{1/2} \\ & + |Q_L E[|\boldsymbol{\varepsilon}_h|^2]| + |R_L E[|\boldsymbol{\varepsilon}_h|^2]| + |R_L V[J(\boldsymbol{u}_h)]| \end{aligned} \tag{49}$$

3.3 Estimating the Standard Deviation Error

In Section 6, several example problems are presented that utilize the expectation error estimate Eq. (44) and variance error estimate Eq. (45). In graphing statistics data, the standard deviation (denoted by $\sigma \equiv \sqrt{V[\cdot]}$) is often reported, rather than the variance. When signed realization and quadrature errors are available, the signed standard deviation error can be directly estimated as follows (using the notation of Section 3.1):

$$\begin{aligned} \sqrt{V[J]} - \sqrt{Q_L V[J_h]} &\approx \sqrt{E[\tilde{J}^2] - (E[\tilde{J}])^2} - \sqrt{Q_L E[J_h^2] - (Q_L E[J_h])^2} \\ &= \sqrt{Q_L E[\tilde{J}^2] + R_L E[\tilde{J}^2] - (Q_L E[\tilde{J}] + R_L E[\tilde{J}])^2} - \sqrt{Q_L E[J_h^2] - Q_L^2 E[J_h]} \end{aligned} \tag{50}$$

When only the magnitudes of realization QOI error and the quadrature error are known, the following approximation can be used:

$$|\sqrt{V[J]} - \sqrt{Q_L V[J_h]}| \approx \frac{|V[J] - Q_L V[J_h]|}{2\sqrt{Q_L V[J_h]}} \tag{51}$$

where $|V[J] - Q_L V[J_h]|$ is calculated using Eq. (49), which only requires the magnitude of realization and quadrature errors.

4 A Software Framework for Non-Intrusive Uncertainty Propagation with Computable Error Bounds

A non-intrusive uncertainty propagation framework with optional error bounds is summarized in the Figure 5 flowchart. A user first specifies sources of uncertainty, provides realizations for outputs of interest, $J(u_h(\boldsymbol{x},t;\boldsymbol{\xi}^{(i)});\boldsymbol{\xi}^{(i)})$, and optionally provides an estimate of the error, $|J(u(\boldsymbol{x},t;\boldsymbol{\xi}^{(i)});\boldsymbol{\xi}^{(i)}) - J(u_h(\boldsymbol{x},t;\boldsymbol{\xi}^{(i)});\boldsymbol{\xi}^{(i)})|, i = 1,\ldots,N$.

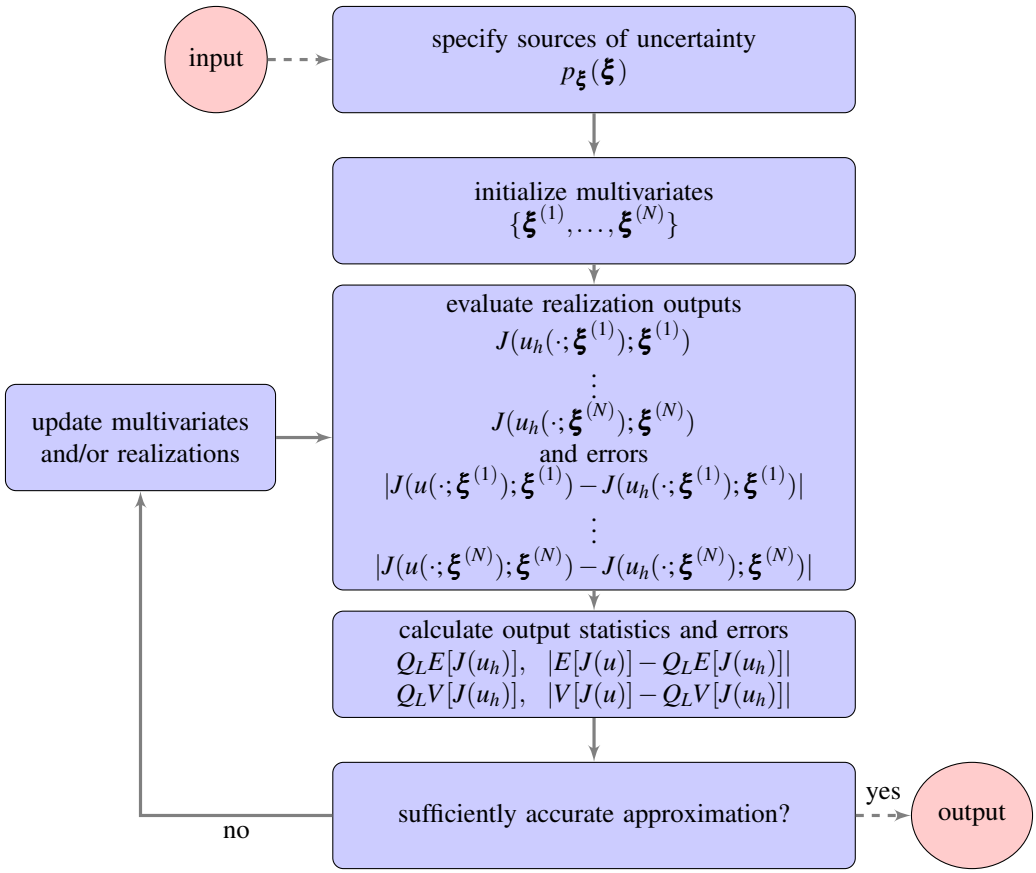


Figure 5. Flowchart of the generalized non-intrusive uncertainty propagation framework.

5 Estimating the Probability Density for Output Quantities of Interest

Output probability density distributions that deviate significantly from a normal distribution are not well characterized by expectation and variance. In this situation, it is often preferable to actually construct an approximation to the output QOI probability density distribution. Kernel density estimation is a standard technique in statistics that is used for this purpose.

5.1 Kernel Density Estimation

Kernel density estimation (KDE) is an algorithm attributed to Rosenblatt [33] and Parzen [28] for estimating the probability density distribution of an independent and identically distributed (i.i.d.) population of samples. Specifically, given a set of M i.i.d. samples $\{x_1, x_2, \ldots, x_M\}$, the probability density distribution is estimated from the sum:

$$p(x) = \frac{1}{Mh} \sum_{i=1}^{M} K\left(\frac{x - x_i}{h}\right) \tag{52}$$

where there is flexibility in the choice of kernel $K(\cdot)$ and bandwidth h. A popular choice is the Gaussian kernel:

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}|x|^2} \tag{53}$$

with bandwidth h calculated from the population standard deviation σ and cardinality M:

$$h = \left(\frac{4\sigma^5}{3M} \right)^{1/5} \tag{54}$$

which is optimal whenever the density being estimated is Gaussian. When M is small, results depend highly on the choice of the bandwidth parameter h and the representation of multi-modal probability density distributions is generally poor. As the sample population cardinality M increases, observe that the bandwidth parameter h given by Eq. (54) decreases. In this increasing limit, the resulting probability density distributions become relatively insensitive to h (even for multi-modal probability density distributions). Consequently, our strategy is to first construct a response surface giving the QOI response to changes in parameter samples. The response surface can then be i.i.d. sampled using a large number of samples (usually $M > 10,000$ or larger) at little cost.

5.2 Kernel Density Estimation for Output Quantities of Interest

Given Clenshaw-Curtis or Gauss-Patterson quadrature points at the finest level L:

$$\{\boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(2)}, \cdots, \boldsymbol{\xi}^{(N_L)}\}$$

and output quantities of interest:

$$\{J[u_h(\boldsymbol{x},t;\boldsymbol{\xi}^{(1)})], J[u_h(\boldsymbol{x},t;\boldsymbol{\xi}^{(2)})], \cdots, J[u_h(\boldsymbol{x},t;\boldsymbol{\xi}^{(N_L)})]\}$$

a response surface is constructed that provides fast approximate evaluation of $J[u_h(\boldsymbol{x},t;\boldsymbol{\xi})]$ using either of the following interpolants:

- Dense tensor product WENO polynomial interpolants [22, 8] using data at Clenshaw-Curtis or Gauss-Patterson quadrature points. These polynomials are already constructed in the HYGAP method so the cost can be minimal.
- Sparse tensor product interpolants [11] using data at sparse tensor product Clenshaw-Curtis or Gauss-Patterson quadrature points.

Once the response surface has been constructed, the cost associated with obtaining a large population of i.i.d. samples is minimal. Using a large population yields high-quality results even for multi-modal probability density distributions.

5.3 Correcting Kernel Density Estimates Using Realization Error

Results presented in Section 6 present probability density distributions for output QOIs that were calculated using KDE via a response surface sampling. Obtaining error bound estimates for probability density distributions that are calculated using KDE via response surface is a complex task

not discussed here. Instead, Section 6 results also show "realization error corrected" probability density distributions that are obtained by first constructing a response surface using output QOI data that has been corrected using an estimate of the realization error, i.e., $J[u_h(\boldsymbol{x},t;\boldsymbol{\xi}^{(i)})]+\varepsilon_h$, and then performing KDE using this corrected response surface. This provides an assessment of the impact of realization error on the output QOI probability density distribution.

Note that in adding realization error as a correction, one often observes "systematic biasing" effects, i.e., the corrected probability density distribution is shifted. This effect is easily understood since adding a *constant* ε_0 to expectation data merely shifts the expectation statistic, $E[J_h+\varepsilon_0]=\varepsilon_0+E[J_h]$, and does not change the variance, $V[J_h+\varepsilon_0]=V[J_h]$.

6 Numerical Applications

Several numerical examples have been selected to demonstrate the capabilities outlined in this article. These examples are briefly summarized in Table 3. Problems using Richardson ($r=2$) or Aitken extrapolation to estimate realization error require CFD simulations using 2 or 3 mesh resolutions, respectively. The multi-element airfoil problem uses a dual problem to estimate the realization error. For each problem, kernel density estimation (KDE), discussed in Section 5, was used to calculate a probability density distribution for the lift coefficient and a drag coefficient for the viscous flow problems.

Table 3. Summary of uncertainty calculations.

Section	Flow Problem	Uncertainty Type	Realization Error	Quadrature
6.1.1	2-D airfoil	boundary condition parameters	Aitken extrapolation	dense
6.1.2	2-D airfoil	turbulence model parameters	Aitken extrapolation	dense
6.2	3-D wing-body	slat and flap geometry	Aitken extrapolation	dense/sparse
6.3	3-D wing	boundary condition parameters	Aitken extrapolation	dense
6.4	3-D launch vehicle	thrust parameter	Richardson extrapolation	dense
6.5	2-D 3-element airfoil	boundary condition parameter	dual problem	dense
6.6	2-D airfoil	correlated random field	Aitken extrapolation	dense/sparse

6.1 NACA 0012 Airfoil Flow Computations with Uncertainty

Steady-state Reynolds-averaged Navier-Stokes (RANS) flow over a NACA 0012 airfoil geometry was calculated using a finite-volume method that achieves second-order spatial accuracy using linear polynomial MUSCL reconstruction [40, 41]. The one-equation Baldwin-Barth [3] turbulence model was used to model the effects of turbulence. Calculations were performed using a hierarchy of three structured meshes containing 513×65, 257×33, and 129×17 mesh points.

Experimental data from Harris [19] is available for comparison at a Mach number of 0.8, angle-of-attack (AOA) of 2.26°, and a Reynolds number of 9×10^6. These transonic flow conditions, as depicted in Figure 6, were chosen because computations utilizing the RANS equations with various

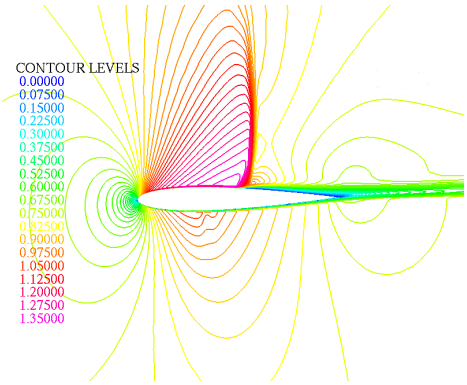


Figure 6. RANS flow over a NACA 0012 airfoil. Mach number contours from a single realization calculation at $M_\infty = 0.8$ and $AOA = 2.26^\circ$.

turbulence models show sensitivity to flow conditions and turbulence model parameters, yet no single value of these parameters fits the experimental data very well. This makes the flow problem an excellent candidate for uncertainty analysis. In subsequent calculations, three different forms of uncertainty for this flow problem are considered:

- Section 6.1.1: Uncertainty in the inflow boundary Mach number and angle-of-attack parameters.
- Section 6.1.2: Uncertainty in 3 Baldwin-Barth turbulence model parameters.
- Section 6.6: Uncertainty in the inflow boundary Mach number profile, modeled as a correlated random field.

6.1.1 NACA 0012 Airfoil Flow with Inflow Uncertainty

Uncertainty in the inflow Mach number and angle-of-attack (AOA) were imposed using Gaussian distributions truncated at four standard deviations and normalized to have unit total probability

- $M_\infty = \text{Gaussian}_{4\sigma}(m = 0.8, \sigma = .01)$
- $AOA = \text{Gaussian}_{4\sigma}(m = 2.26^\circ, \sigma = 0.1^\circ)$

Uncertainty statistics for output QOIs were then calculated using the Clenshaw-Curtis HYGAP approximation discussed in Section 2.5.3. Pressure coefficient statistics and Type I estimates of total error on the airfoil surface are graphed in Figure 7. The calculation using $L = 4$ Clenshaw-Curtis quadrature requires 81 CFD evaluations for each mesh resolution in the hierarchy of CFD meshes. Overall, the level of error in mean and standard deviation statistics is rather small except within the upper surface shock wave profile and immediately downstream of the shock wave. For the particular flow conditions chosen, this figure also shows the surprisingly large uncertainty in lower surface pressure coefficient values. This is apparently due to the presence of a weak lower surface shock wave that is sometimes present when the uncertainty parameters are varied. The presence of a lower surface shock wave significantly changes the solution and arguably explains

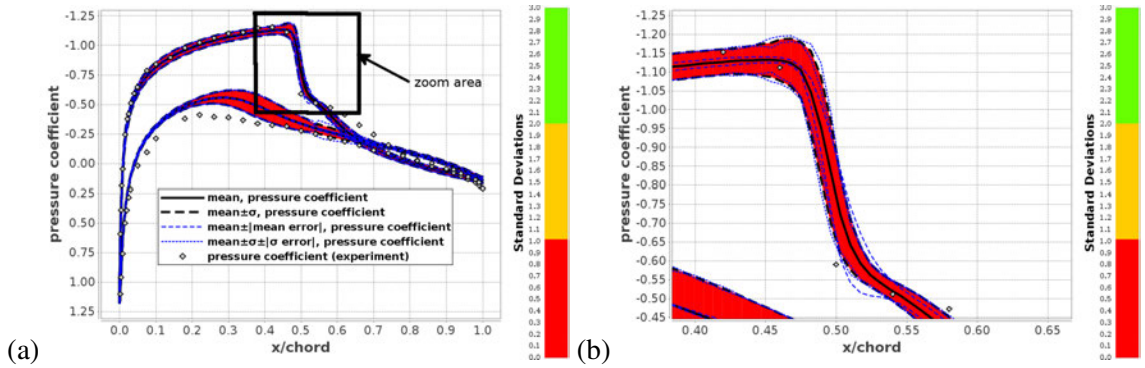


Figure 7. Surface pressure coefficient statistics on the NACA 0012 airfoil surface for transonic flow with inflow Mach number and AOA uncertainty. Shown are mean and standard deviation statistics as well as the estimates of mean and standard deviation errors for (a) the entire airfoil and (b) a closeup in the upper surface shock region.

the relatively large uncertainty in the lower surface pressure distribution. More fundamentally, the appearence of a strong shock wave on the upper surface and a weak lower surface shock wave that depend on values of the random parameters may cause the random variable data to be non-smooth in these locations. For non-smooth random variable data, the Clenshaw-Curtis HYGAP algorithm switches from standard global Clenshaw-Curtis quadrature to piecewise polynomial approximated quadrature using the Clenshaw-Curtis quadrature point data.

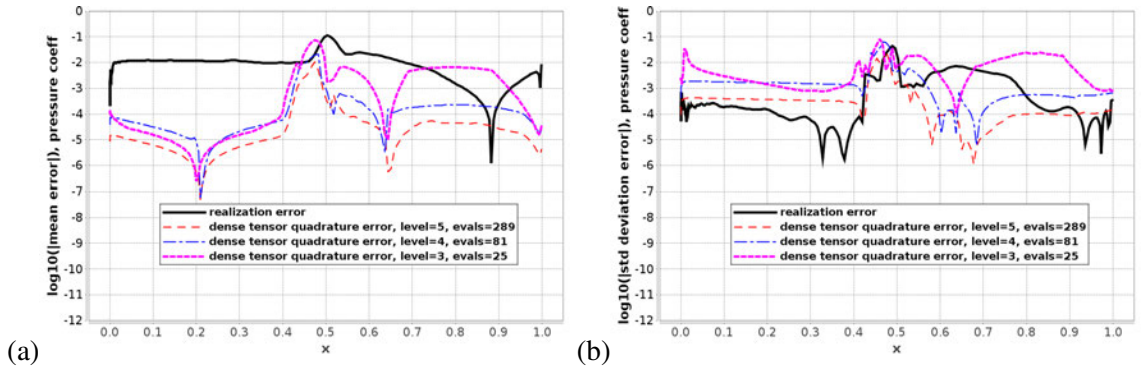


Figure 8. Statistics errors on the upper surface of the NACA 0012 airfoil for transonic flow with inflow Mach number and AOA uncertainty. Shown are Type I estimates of realization error and quadrature errors using $L = 3,4,5$ Clenshaw-Curtis HYGAP quadrature for the (a) mean and (b) standard deviation.

Figures 8 and 9 show graphs for Type I and Type II estimates of mean and standard deviation error for this uncertainty calculation. The Type I error estimates do not rely on inequality estimates, and consequently are generally assumed to be sharper than Type II error bound estimates. Realization and quadrature error curves for $L = 3,4,5$ HYGAP Clenshaw-Curtis quadrature are shown in figures 8 and 9 for comparison. The realization error curves are obtained by artificially setting the quadrature error equal to zero. Similarly, the quadrature error curves are obtained by artificially setting the realization error equal to zero. When compared to the Type I realization error estimates, the Type I quadrature error estimates from both $L = 4$ and $L = 5$ Clenshaw-Curtis HYGAP approximations appear acceptable. The criteria for acceptability used here assumes that the realization error from the CFD calculations is given and the quadrature error should balance or be less than (but perhaps

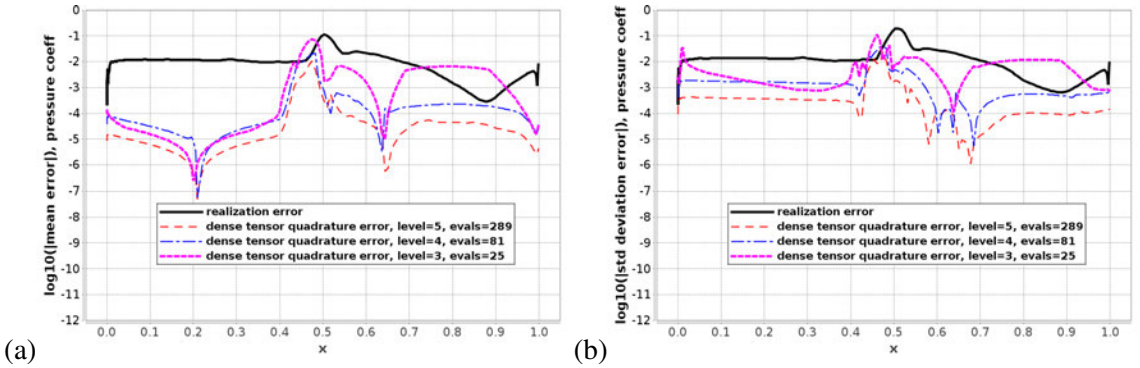


Figure 9. Statistics errors on the upper surface of the NACA 0012 airfoil for transonic flow with inflow Mach number and AOA uncertainty. Shown are Type II estimates of realization error and quadrature errors using $L = 3,4,5$ Clenshaw-Curtis HYGAP quadrature for the (a) mean and (b) standard deviation.

not strictly less than) the realization error. The computational savings in using $L = 4$ approximation is substantial since for two sources of uncertainty it requires only 81 CFD evaluations as compared to $L = 5$, which requires 289 CFD evaluations. Figure 10 graphs the probability density functions

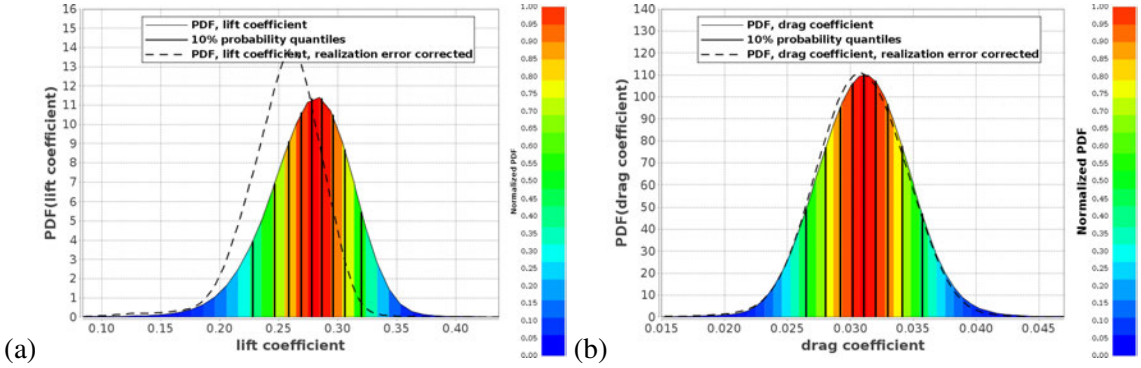


Figure 10. Probability density distributions for (a) lift coefficient and (b) drag coefficient for transonic flow over the NACA 0012 geometry with inflow Mach number and AOA uncertainty.

for lift and drag coefficients using the kernel density estimation procedure discussed in Section 5. In addition, this figure graphs the probability density function corrected by realization error, also discussed in Section 5. All distributions have a near-normal (Gaussian) shape. Surprisingly, the lift coefficient distribution seems most heavily impacted by the realization error with only a minor impact seen in the drag coefficient distribution.

6.1.2 NACA 0012 Airfoil Flow with Turbulence Model Parameter Uncertainty

The computations of the previous example are now repeated with uncertainty imposed in parameters of the Baldwin-Barth [3] turbulence model. Given a fluid density ρ, velocity components $u_i, i = 1,\ldots,d$ from the RANS approximation, and a wall distance y^+, the Baldwin-Barth turbulence model is a one-equation PDE of the form:

$$\frac{DR_T}{Dt} = (c_{\varepsilon_2} f_2(y^+) - c_{\varepsilon_1})\sqrt{R_T P} + (\nu + \frac{\nu_T}{\sigma})\nabla^2 R_T - \frac{1}{\sigma}(\nabla \nu_t)\cdot \nabla R_T \tag{55}$$

where

$$\mu_T = \rho \nu_T = \rho c_\mu D_1(y^+) D_2(y^+) R_T \quad , \quad P = \nu_T \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \frac{\partial u_i}{\partial x_j} - \frac{2}{3} \nu_T \left(\frac{\partial u_k}{\partial x_k} \right)^2$$

$$D_1 = 1 - \exp(-y^+/A^+) \quad , \quad D_2 = 1 - \exp(-y^+/A_2^+), \quad \sigma^{-1} = (c_{\varepsilon_2} - c_{\varepsilon_1}) \sqrt{c_\mu} / \kappa^2$$

and

$$\begin{aligned} f_2(y^+) &= \frac{c_{\varepsilon_1}}{c_{\varepsilon_2}} + (1 - \frac{c_{\varepsilon_1}}{c_{\varepsilon_2}})(\frac{1}{\kappa y^+} + D_1 D_2) \\ &\quad \times \left(\sqrt{D_1 D_2} + \frac{y^+}{\sqrt{D_1 D_2}} (\frac{1}{A^+} \exp(-y^+/A^+) D_2 + \frac{1}{A_2^+} \exp(-y^+/A_2^+) D_1) \right) \end{aligned}$$

with reference model parameters $\kappa = .41, c_\mu = .09, c_{\varepsilon_1} = 1.2, c_{\varepsilon_2} = 2.0, A^+ = 26$, and $A_2^+ = 10$ as given in Baldwin and Barth [3]. Uncertainty is imposed in three parameters of the model that are often observed to be most sensitive:

- $c_\mu = \text{Uniform}[.081, .099]$
- $c_{\varepsilon_1} = \text{Uniform}[1.08, 1.32]$
- $c_{\varepsilon_2} = \text{Uniform}[1.8, 2.2]$

Uncertainty statistics for output QOIs were then calculated using the Clenshaw-Curtis HYGAP approximation discussed in Section 2.5.3. The calculation using the $L = 4$ Clenshaw-Curtis quadrature

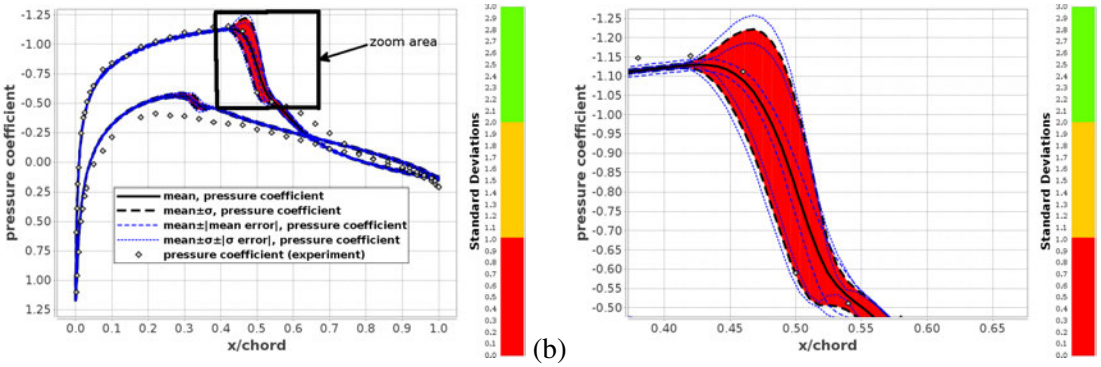


Figure 11. Surface pressure coefficient statistics on the NACA 0012 airfoil surface for transonic flow with Baldwin-Barth turbulence model uncertainty. Shown are mean and standard deviation statistics as well as the estimates of mean and standard deviation errors for (a) the entire airfoil and (b) a closeup in the upper surface shock region.

with three sources of uncertainty requires 729 CFD evaluations for each mesh resolution in the hierarchy of CFD meshes. Pressure coefficient statistics and Type I error estimates on the airfoil surface are graphed in Figure 11. In contrast to the previous example, the uncertainties remain somewhat confined to the upper shock profile region with only very small uncertainties seen on the lower surface. Type I and Type II pressure coefficient error estimates are shown in figures 12 and 13. Using the criteria of Section 6.1.1, examination of the errors in mean and standard deviation graphed in

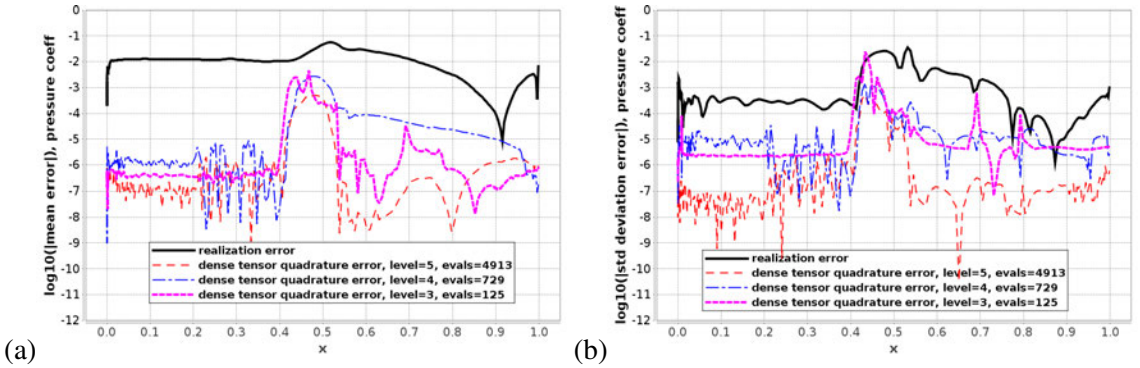


Figure 12. Statistics errors on the upper surface of the NACA 0012 airfoil for transonic flow with Baldwin-Barth turbulence model uncertainty. Shown are Type I estimates of realization error and quadrature errors using $L = 3,4$ Clenshaw-Curtis HYGAP quadrature for the (a) mean and (b) standard deviation statistic.

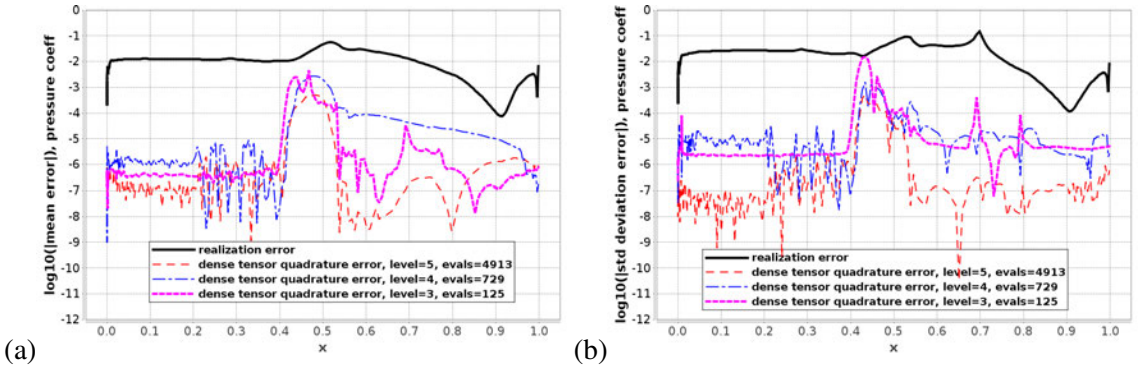


Figure 13. Statistics errors on the upper surface of the NACA 0012 airfoil for transonic flow with Baldwin-Barth turbulence model uncertainty. Shown are Type II estimates of realization error and quadrature errors using $L = 3,4$ Clenshaw-Curtis HYGAP quadrature for the (a) mean and (b) standard deviation statistic.

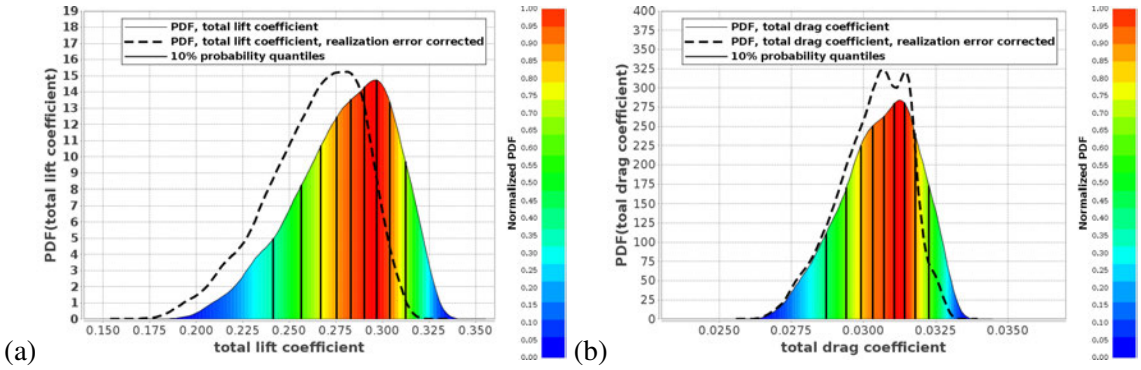


Figure 14. Probability density distributions for (a) lift coefficient and (b) drag coefficient for transonic flow over the NACA 0012 geometry with Baldwin-Barth turbulence model uncertainty.

Figure 12 indicate that the $L = 3$ Clenshaw-Curtis HYGAP approximation using just 125 CFD evaluations at each CFD mesh resolution may be sufficient. Figure 14 graphs the probability density

distributions for lift and drag coefficients with and without realization error correction, as discussed in Section 5. These graphs do show a non-negligible impact of realization error on the probability density distributions for both lift and drag. Observe that although the input uncertainty distributions for turbulence model parameters are uniform distributions, the resulting output distributions have a more peaked Gaussian shape. This has significant meaning. The input probability laws for model parameters were all chosen as uniform probability density distributions. This implies for each parameter that all possible values in an interval have equal probability. The observed output probability density is a peaked distribution, which implies that this output is relatively insensitive to the uncertain model parameters. This form of input-output response is often sought in the design of model systems.

6.2 High-Lift Wing-Body Flow with Slat and Flap Geometric Uncertainty

Steady-state Reynolds-averaged Navier-Stokes flow over a high-lift wing-body geometry at Mach 0.2 and angle-of-attack 13° was computed using the OVERFLOW [21] CFD solver. This test case is taken from the 1st AIAA CFD High Lift Prediction Workshop [36] held in 2010. Details of the OVERFLOW calculations presented at that workshop can be found in Sclafani *et al.* [35]. Experimental data from NASA Langley Research Center wind tunnel testing is available for comparison.

In the present calculations, uncertainty in the slat and flap angles has been imposed:

- $\alpha_{\text{slat}} = 30^\circ + \text{Gaussian}_{4\sigma}(m = 0.0^\circ, \sigma = .75^\circ)$
- $\alpha_{\text{flap}} = 25^\circ + \text{Gaussian}_{4\sigma}(m = 0.0^\circ, \sigma = .75^\circ)$

A hierarchy of refined CFD mesh resolutions was used in calculations so that realization errors can be estimated using Aitken extrapolation. The finest resolution meshes contain approximately 90 million mesh points. Figure 15 shows surface pressure contours from a single OVERFLOW

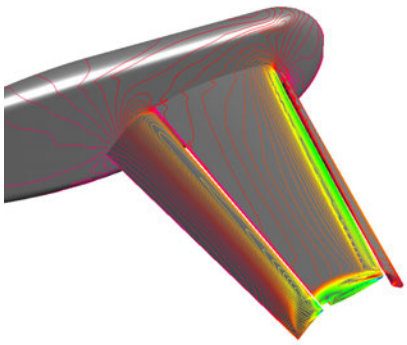


Figure 15. Surface pressure coefficient contours on the NASA Trap Wing high-lift model with inflow conditions $M_\infty = 0.2$, $AOA = 13^\circ$ with slat deployed 30° and slat deployed 25°. Green denotes low values and red denotes high values.

calculation with slat deployed 30° and slat deployed 25°. Uncertainty statistics for output QOIs

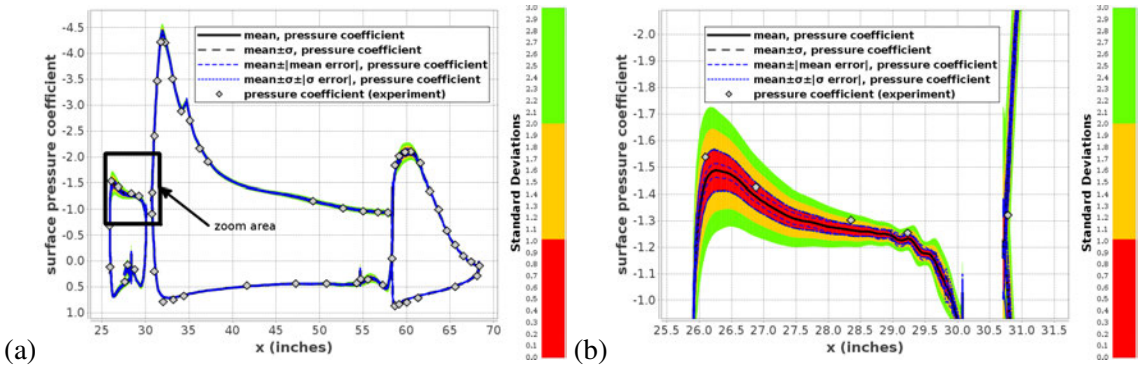


Figure 16. Surface pressure coefficient statistics at the 50% wing span location for the NASA Trap Wing high-lift model with slat and flap angle uncertainty. Shown are mean and standard deviation statistics as well as the estimates of mean and standard deviation errors for (a) entire wing section and (b) a closeup in the slat region.

were calculated using Clenshaw-Curtis quadrature points in dense and sparse tensor product form. The calculation using $L = 4$ Clenshaw-Curtis dense tensor product quadrature requires 81 CFD evaluations and the sparse tensor product quadrature requires 29 CFD evaluations for each mesh resolution in the hierarchy. Surface pressure statistics and errors at the 50% wing span location are shown in Figure 16. The uncertainties appear to be quite small everywhere except the leading

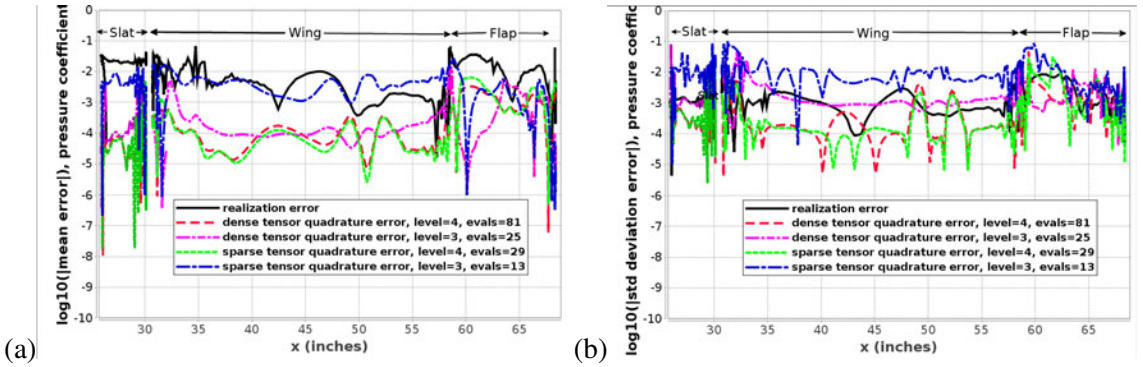


Figure 17. Statistics errors on the upper surface of the NASA Trap Wing high-lift model with slat and flap angle uncertainty. Shown are realization error and quadrature errors using $L = 3,4$ sparse and dense tensor product forms of Clenshaw-Curtis quadrature for the (a) mean and (b) standard deviation.

edge slat and near the leading edge of the flap. Figure 17 shows realization error and quadrature error for the mean and standard deviation statistics on the upper surface of the wing at 50% span using both sparse and dense tensor product forms of Clenshaw-Curtis quadrature for $L = 3$ and $L = 4$ approximation. When compared to the realization error, all the quadratures except the $L = 3$ sparse tensor product quadrature have acceptable levels of error. The $L = 4$ sparse tensor product quadrature requires just 29 CFD realization evaluations for each mesh resolution and seems to be a good choice for this problem.

Figure 18 shows the probability density distributions for lift and drag coefficients with and without realization error correction, as discussed in Section 5. These graphs show the non-negligible effect

of realization error on the probability density distributions for both lift and drag. When compared to the lift probability density distribution, the flatness in the drag probability density distribution indicates that the drag is slightly more sensitive to changes in slat and flap angle.

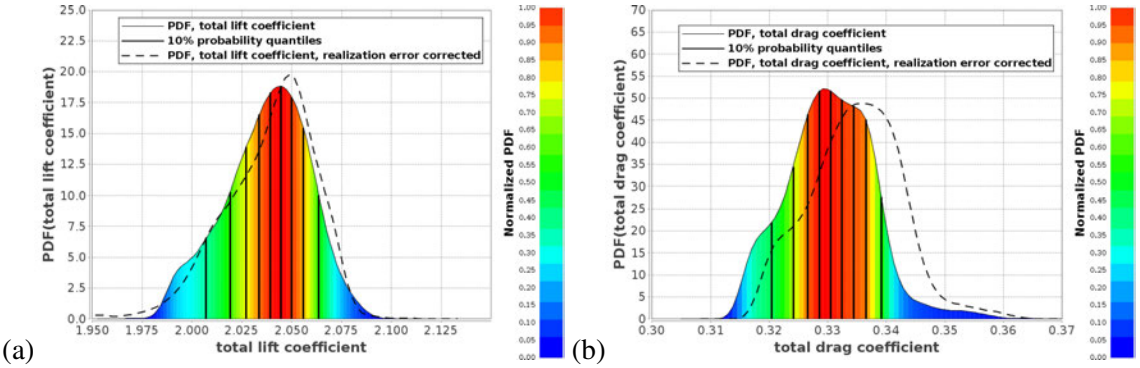


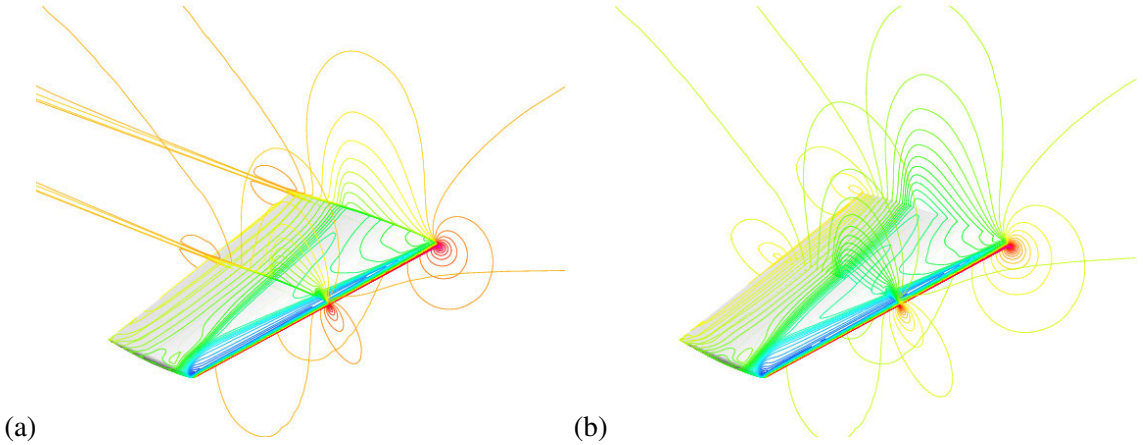
Figure 18. Probability density distributions for (a) lift coefficient and (b) drag coefficient for the NASA Trap Wing high-lift model with slat and flap angle uncertainty.

6.3 ONERA M6 Wing Flow with Uncertain Inflow Conditions

Steady-state Reynolds-averaged Navier-Stokes flow over an ONERA M6 wing at a Reynolds number of 11.72×10^6 was computed using the OVERFLOW [21] CFD solver. Experimental data by Schmitt and Charpin [34] is available for comparison. Uncertainty in the inflow Mach number and angle-of-attack (AOA) were imposed, i.e.:

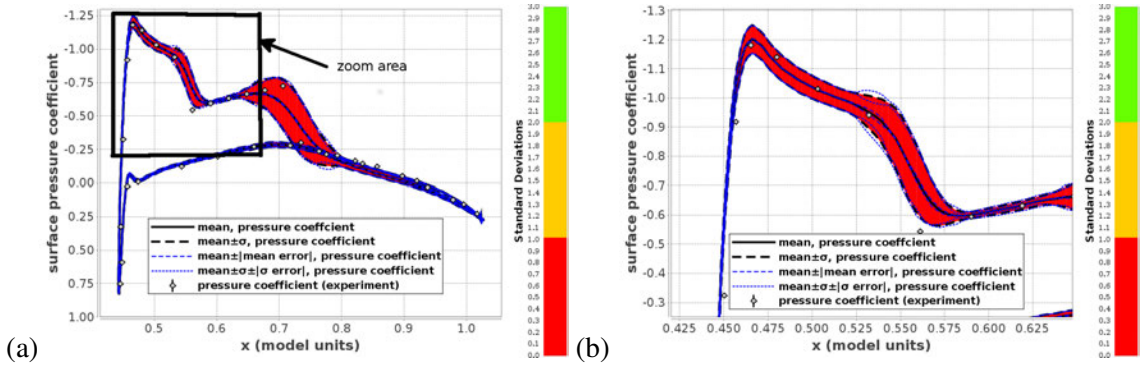
- $M_\infty = \text{Gaussian}_{4\sigma}(m = .84, \sigma = .02)$
- AOA = $\text{Gaussian}_{4\sigma}(m = 3.06^\circ, \sigma = .075^\circ)$

Calculations were performed on a hierarchy of refined meshes containing approximately 5 million mesh points at the finest resolution. Uncertainty statistics were calculated using $L = 4$ Clenshaw-Curtis HYGAP approximation, defined in Section 2.5.3, which uses Clenshaw-Curtis quadrature for smooth random variable data and cubic piecewise polynomials for non-smooth random variable data. Density and pressure coefficient contours from a single numerical realization at $M_\infty = .84$, $AOA = 3.06^\circ$ are presented in Figure 19. The well-known lambda shock pattern is clearly seen on the upper wing surface. Pressure coefficient statistics and Type I error estimates on the wing surface at the 65% span location are graphed in Figure 20. Significant uncertainties are observed on the upper surface of the wing at the lambda shock and leading edge locations with almost no uncertainty seen in the lower surface. Type I and Type II realization and quadrature error estimates for mean and standard deviation are shown in figures 21 and 22 for $L = 3, 4, 5$ Clenshaw-Curtis HYGAP approximations. Based on results from Figure 21, $L = 4$ approximation using 81 CFD evaluations for each mesh resolution results in a level of error comparable to the realization error and appears to be an acceptable choice for this calculation. To understand these results in more detail, kernel density estimation was used to construct the actual probability density distribution for surface



(a) (b)

Figure 19. Transonic Reynolds-averaged Navier-Stokes flow over the ONERA M6 wing at Mach .84 and AOA = 3.06°. Shown are single realization (a) density contours and (b) pressure coefficient contours. Blue denotes low values and red denotes high values.



(a) (b)

Figure 20. Surface pressure coefficient statistics at the 65% wing span location for the ONERA M6 wing problem with Mach number and AOA uncertainty. Shown are mean and standard deviation statistics as well as the estimates of mean and standard deviation errors for (a) entire wing section and (b) a closeup in the slat region.

pressure coefficient at each location of the wing surface at the 65% wing span location. Figure 23(a) shows the surface pressure coefficient probability density distribution over the entire wing section, but this probability distribution only has meaning when traversed vertically for a particular x location on the upper or lower surface. Figure 23(b) shows the resulting probability density distribution near the leading edge at $x = 0.5$ on the upper surface. This distribution is very well approximated by a Gaussian distribution, which justifies the use of mean and standard deviation to characterize it. Figure 23(c) shows the surface pressure coefficient probability density distribution at the location of the rear shock wave at $x = 0.722$. The bi-modal shape of this probability density distribution gives high probability to the pre-shock and post-shock states observed in realizations. The standard deviation σ at this location is approximately one-half the distance *between* peaks in the bi-modal distribution and does not characterize the width of any peak. Consequently, any model optimization procedure that seeks to minimize σ for this QOI may be drastically misled. Figure 24 graphs the probability density distributions for lift and drag coefficients with and without realization error correction. The left shifting of corrected distributions is due to the systematic bias that is expected

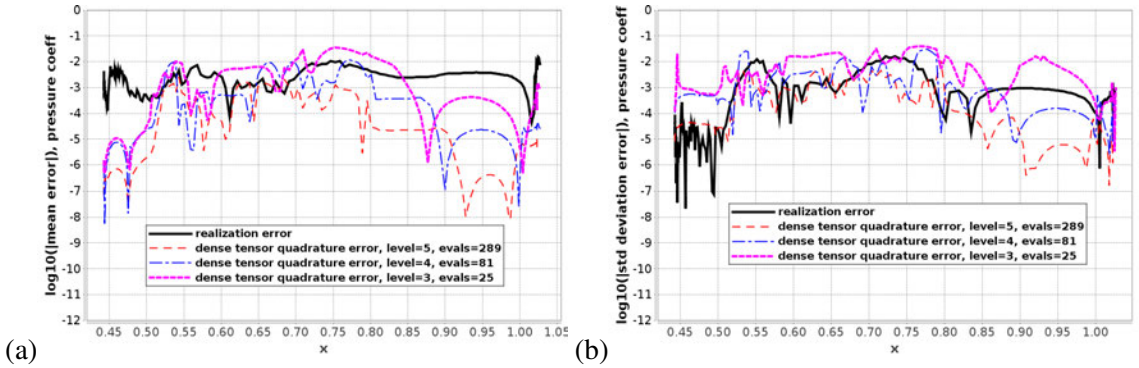


Figure 21. Statistics errors on the wing upper surface for the ONERA M6 wing problem with Mach number and AOA uncertainty. Shown are Type I estimates of realization error and quadrature errors using $L = 3,4,5$ dense tensor product forms of Clenshaw-Curtis quadrature for the (a) mean and (b) standard deviation.

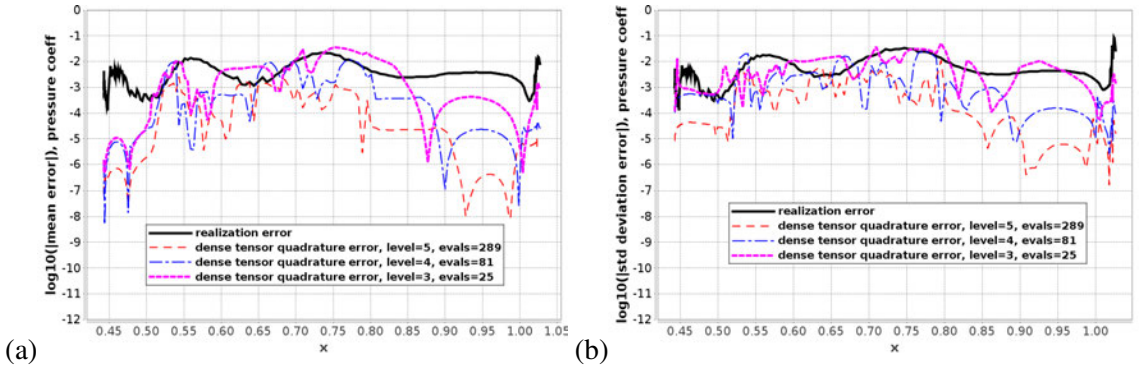


Figure 22. Statistics errors on the wing upper surface for the ONERA M6 wing problem with Mach number and AOA uncertainty. Shown are Type II estimates of realization error and quadrature errors using $L = 3,4,5$ dense tensor product forms of Clenshaw-Curtis quadrature for the (a) mean and (b) standard deviation.

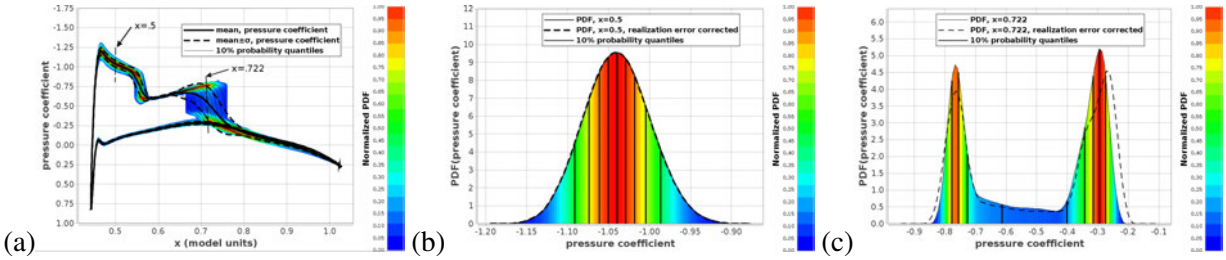


Figure 23. Surface pressure coefficient statistics at the 65% wing span location for the ONERA M6 wing problem with Mach number and AOA uncertainty. Shown are a color shaded probability density function with 10% quantile partititions for (a) the entire wing section, (b) at the location $x = .5$ on the upper surface, and (c) at the shock wave location $x = .722$ on the upper surface.

from realization error corrections, as discussed in Section 5. Given that the input probability laws for Mach number and angle-of-attack where chosen to be Gaussian distributions, the near Gaussian shape of lift, and the skewed Gaussian shape of drag may be interpreted as meaning that lift is

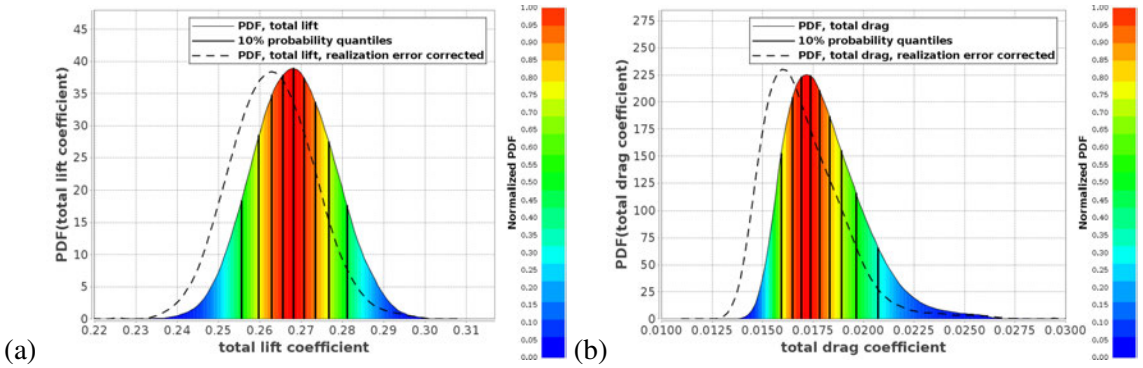


Figure 24. Probability density distributions for (a) lift coefficient and (b) drag coefficient for the ONERA M6 wing problem with Mach number and AOA uncertainty.

responding nearly linearly to variations in the uncertain parameters while drag is not.

6.4 Launch Vehicle Plume Analysis with Uncertain Thrust

In this example, uncertainty propagation for a launch vehicle simulation with exhaust plume modeling is considered. Launch vehicle designs must cope with very large forces, volatile chemicals,

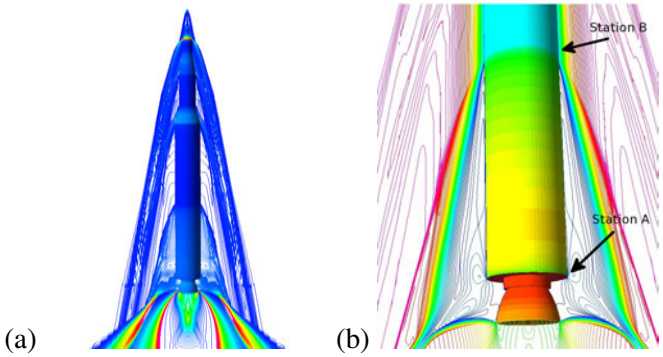


Figure 25. Mach 6.7 flow over a launch vehicle configuration with rocket plume modeling. Shown are (a) pressure contours over the entire launch vehicle and (b) contours of flow field Mach number and surface color shading of flow field density in the lower rocket region. Blue denotes low values and red denotes high values.

and extremely hot rocket plume gases. A phenomenon sometimes encountered in these designs is Plume Induced Flow Separation (PIFS). As an example of this phenomenon, Figure 25(a) shows Mach 6.7 flow over a launch vehicle with rocket plume modeling approximated using the OVERFLOW [21] CFD solver (courtesy Goetz Klopfer, NASA Ames). As the launch vehicle accelerates and ascends into the atmosphere, the rocket plume expands and eventually causes the flow to separate near Station B on the rocket body with reversed flow occurring between Stations A and B in Figure 25(b). This reversed flow may carry hot plume gases in close proximity to the rocket body, resulting in material failure unless additional thermal protection is provided. As a historical

note, this phenomenon was encountered in the NASA Apollo Saturn-V launches during 1967-1973. Consequently, during those launches one of the five Saturn-V engines was intentionally powered off at a prescribed altitude to reduce the severity of the PIFS phenomenon. For the present simplified single-engine configuration, calculations were performed to quantify the extent of the PIFS flow reversal with respect to uncertain flight Mach number and thrust conditions, i.e.:

- $M_\infty = \text{Gaussian}_{3\sigma}(m = 6.7, \sigma = .067)$
- $\text{thrust} = \text{Uniform}[\text{thrust}_{80\%}, \text{thrust}_{100\%}]$

where $\text{thrust}_{80\%}$ and $\text{thrust}_{100\%}$ denote thrust at 80% and 100% power settings, respectively. One way to characterize the extent of this reversed flow is via a skin friction coefficient that changes sign when the vertical (streamwise) component of velocity changes direction.

Figure 26 graphs skin friction coefficient statistics and Type I error estimates between Stations A and B using $L = 4$ Clenshaw-Curtis HYGAP approximation using 81 CFD evaluations.

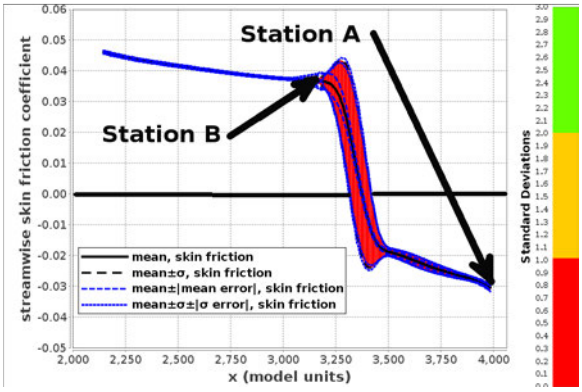


Figure 26. Mach 6.7 flow over a launch vehicle configuration with Mach number and thrust uncertainty. Shown are skin friction coefficient uncertainty statistics and Type I error estimates using $L = 4$ Clenshaw-Curtis HYGAP uncertainty approximation.

The mean streamwise skin friction component becomes negative at approximately $x = 3,375$, which indicates flow reversal and flow separation; but there is significant uncertainty in this position due to Mach number and thrust uncertainty. This uncertainty in flow separation and reversal position is the information sought by engineers. Figures 27 and 28 provide graphs of Type I and Type II estimates of mean and standard deviation error in the region between Station A and Station B for this calculation. These figures indicate that the accuracy in these output statistics appears to be dominated by realization error in the CFD computations. This is useful information if more accurate statistics are required because it indicates that additional resources should be first used to improve the accuracy of the CFD realizations and *not* used to improve the accuracy of the statistics integral quadratures until error balancing is achieved.

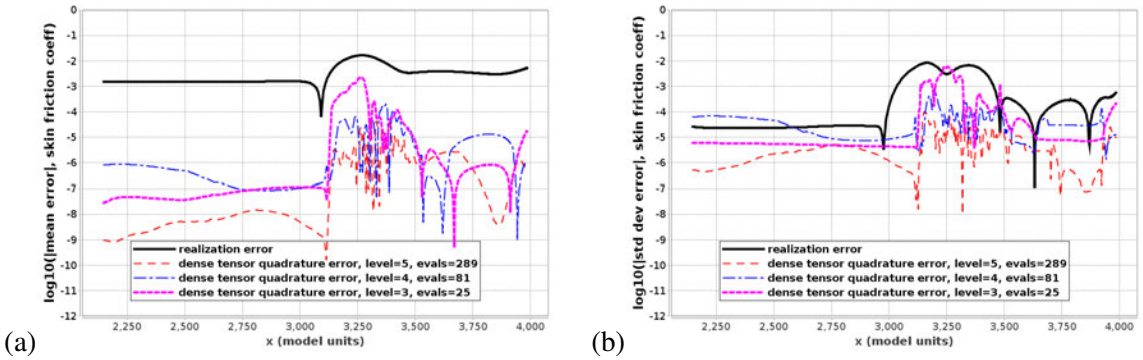


Figure 27. Mach 6.7 flow over a launch vehicle configuration with thrust uncertainty calculated using $L = 3,4,5$ Clenshaw-Curtis HYGAP approximation. Shown are graphs of Type I estimates of skin friction coefficient (a) mean error and (b) standard deviation error between Station A and Station B.

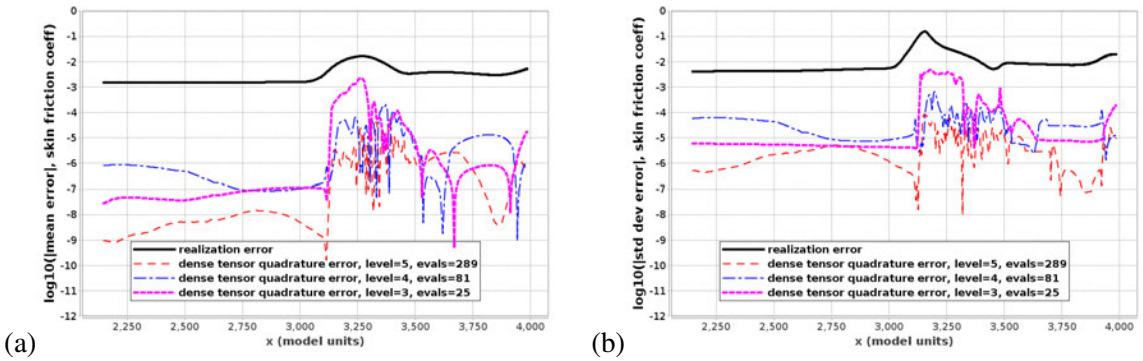


Figure 28. Mach 6.7 flow over a launch vehicle configuration with thrust uncertainty calculated using $L = 3,4,5$ Clenshaw-Curtis HYGAP approximation. Shown are graphs of Type II estimates of skin friction coefficient (a) mean error and (b) standard deviation error between Station A and Station B.

6.5 Inviscid Euler Flow Over a Multi-Element Airfoil with Uncertainty

In this example, steady-state Euler flow over a multi-element airfoil at Mach 0.1 has been computed with angle-of-attack (AOA) uncertainty:

$$AOA = \text{Gaussian}_{4\sigma}(m = 5^\circ, \sigma = 0.1^\circ)$$

Using the theory outlined in Section 2.6.2, a dual problem is used to estimate realization error for the lift coefficient QOI using the error representation equation (35). These dual problems are also used to adaptively refine the mesh using the adaptive refinement indicators in Eq. (37) to reduce the realization error in the lift coefficient QOI. The flow problem was approximated using a discontinuous Galerkin finite-element method [31, 24, 15] formulated in entropy stable symmetric variables as proposed in [4, 5] and used in solving dual problems arising in *a posteriori* error estimation in Larson and Barth [23]. Primal flow problems were computed using linear elements and dual problems were computed using quadratic elements. Figure 29(a) shows Mach number contours of the primal Euler flow problem during the adaptive mesh refinement process and Figure 29(b) shows contours of the dual problem (x-momentum) for the lift coefficient functional. A graph of the error reduction in the lift coefficient during the adaptive meshing process is shown in Figure 30(a) and

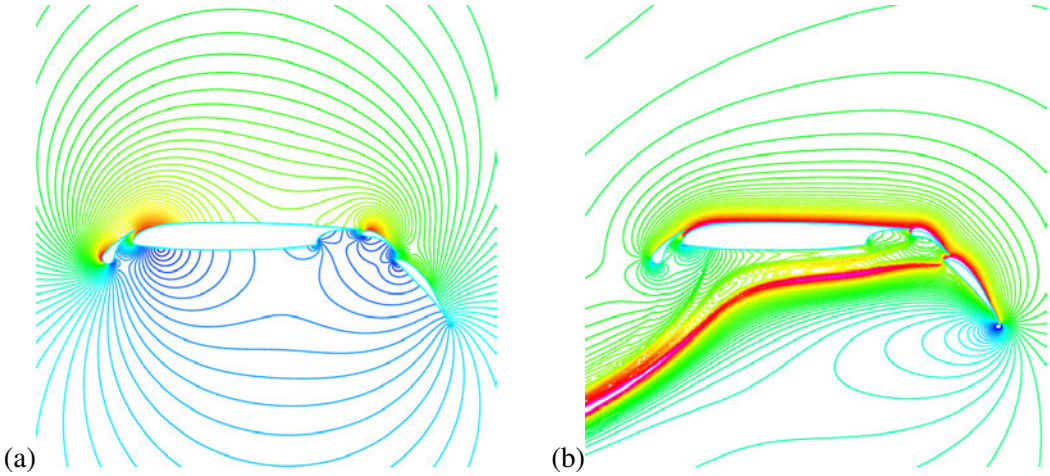


Figure 29. A single realization of Mach 0.1 Euler flow over a multi-element airfoil geometry at 5° AOA. Shown are (a) Mach number contours associated with the primal solution and (b) contours of the x-momentum associated with the dual solution for a lift functional. Blue denotes small values and red denotes large values.

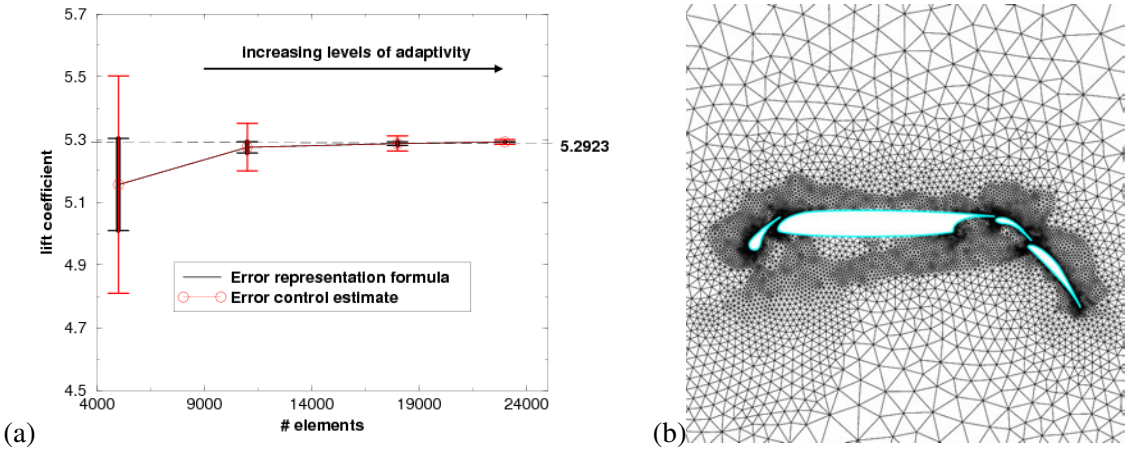


Figure 30. Adaptive mesh refinement for the 5° realization. Shown are (a) graphs of the estimated error in lift coefficient functional using estimates in equations (36) and (37) and (b) the resulting adapted mesh with 18,000 elements after two levels of adaptive refinement.

the adaptive mesh obtained after two levels of refinement is shown in Figure 30(b). Meshes with 11,000 and 18,000 elements were created after 1 and 2 levels of adaptive meshing, respectively. Using the 18,000 element mesh, $L = 5$ Clenshaw-Curtis quadrature yields a lift coefficient mean value of $.52940 \pm .00650$ and standard deviation of $.00704 \pm .000002$. In this calculation, the error representation equation (35) was used to estimate the realization QOI error. Some computational saving can be achieved by reusing the dual problem solution for different realizations but this assumes that the primal problem linearization error can be neglected. One should weigh the cost and complexity of solving a dual problem against the alternative of simply solving the primal problem with a more accurate method and estimating the realization error from it. This latter approach has a certain appeal but if the estimated realization error is found to be too large, lacking a dual prob-

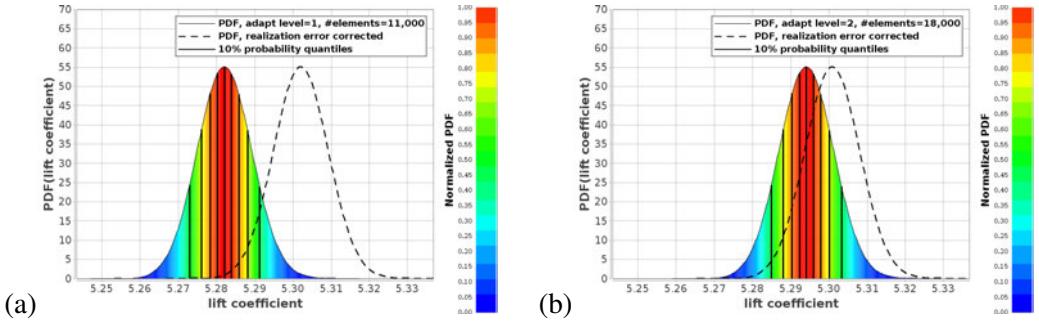


Figure 31. Probability density distributions for the lift coefficient using (a) 11,000 elements and (b) 18,0000 elements for Euler flow over a multi-element airfoil with AOA uncertainty.

lem, the approach does not provide information on how the primal should refined to make the error smaller.

Kernel density estimation in Section 5 has been used to calculate the lift coefficient probability density distribution with and without realization error correction, as shown in Figure 31. Observe in Figure 31 that adding a realization correction does not appear to change the shape, but merely shifts the probability density distribution. This is an expected systematic bias that results from realization error, as discussed in Section 5.3.

6.6 NACA 0012 Airfoil Flow with Correlated Random Field Inflow

Correlated random fields in fluid flow simulations are often introduced to model microscopic spatial or temporal flow perturbations idealized as distributed Wiener processes. A finite correlation between random perturbations at nearby spatial or temporal locations results in a macroscopic scale at which the fluid flow behavior is smooth. A well-known example is the stochastic modeling of a fluid flow in a porous media. A correlated random field is used in these simulations to model the microscale perturbations introduced by the porous material that result in a smooth macroscopic flow field. Random fields also naturally arise in fluid flow simulations as a model of initial and boundary data subject to correlated random perturbations. The next example demonstrates this use in simulating transonic flow over an airfoil with random field inflow data.

6.6.1 Karhunen-Loeve Representation of Correlated Random Fields

The Karhunen-Loeve (K-L) expansion provides an efficient basis for representing correlated random fields. The K-L expansion represents independent Wiener processes at different locations by a single process modulated by a function depending on location. The K-L expansion for a function $f(t,\omega)$ takes the form:

$$f(t,\omega) = \sum_{i=1}^{\infty} \sqrt{\lambda_i}\, \xi_i(\omega)\, \psi_i(t) \tag{56}$$

with $\psi_i(t)$ orthogonal eigenfunctions and λ_i eigenvalues of a correlation function $C(t_1,t_2)$ with spectral representation:

$$C(t_1,t_2) = \sum_{i=1}^{\infty} \lambda_i \psi_i(t_1)\psi_i(t_2) \tag{57}$$

satisfying the Fredholm integral equation of the second kind for $i = 1,2,\ldots,\infty$

$$\int_D C(t_1,t_2)\psi_i(t_1)\,dt_1 = \lambda_i \psi_i(t_2) \tag{58}$$

From orthogonality of the eigenfunctions ψ_i, it immediately follows that:

$$\xi_i(\omega) = \frac{1}{\sqrt{\lambda_i}} \int_D f(t,\omega)\psi_i(t)\,dt \tag{59}$$

For Gaussian processes, the ξ_i are centered, mutually uncorrelated, random variables with unit variance. Consequently, an n-term K-L expansion results in n uncorrelated random variable dimensions.

6.6.2 Airfoil Flow with Random Field Inflow Data

As an example of random field uncertainty, a 2-D transonic Reynolds-averaged Navier-Stokes flow

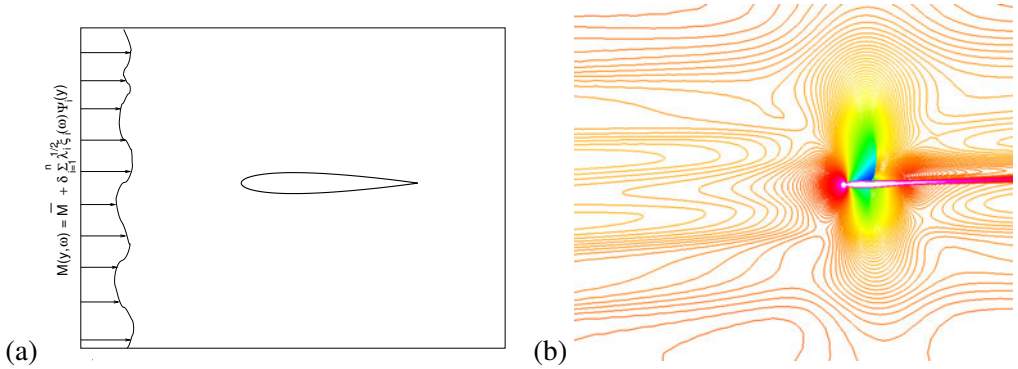
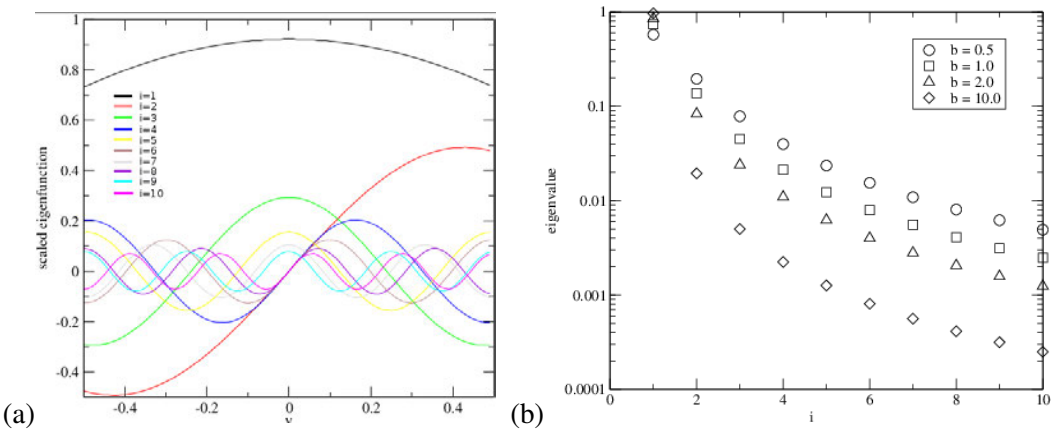


Figure 32. Flow over an airfoil with correlated random field inflow data. Shown are (a) closeup schematic of flow over an airfoil with random field inflow Mach number data, and (b) horizontal velocity component contours for a single Reynolds-averaged Navier-Stokes realization.

field over an airfoil at 2.26° angle-of-attack, with correlated random field inflow data, has been numerically approximated; see the schematic in Figure 32(a) and a sample Reynolds-averaged Navier-Stokes realization in Figure 32(b). The inflow Mach number profile in the vertical dimension y is a random field with exponential correlation:

$$C(y_1,y_2) = \varepsilon^2 e^{-|y_1-y_2|/b} \tag{60}$$

for a specified correlation length b and parameter ε. The associated Fredholm equation (58) using this correlation has a simple analytical solution. This exact solution is readily obtained by twice integrating the Fredholm equation with respect to the independent variable y_2 and applying boundary conditions. Scaled eigenfunctions $\sqrt{\lambda_i}\psi_i(y)$ are graphed in Figure 33(a) for $\sigma = 1.0$, $a = b = 1/2$, and $i = 1,\ldots,10$. For this same problem, the eigenvalues λ_i are graphed in Figure 33(b) for several



(a) (b)

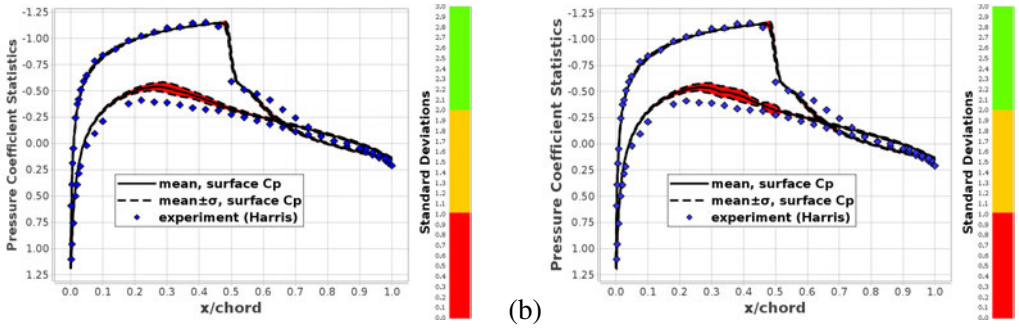
Figure 33. (a) scaled eigenfunctions, $\sqrt{\lambda_i}\psi_i(y)$, for $i = 1,\ldots,10$ and (b) eigenvalues λ_i for correlation lengths $b \in \{0.5, 1.0, 2.0, 10.0\}$.

different correlation lengths b. Observe that the eigenvalues decay rapidly (more so as b increases). This rapid decay motivates the use of the following n-term truncated approximation with mean $\overline{M}$ and scale ε:

$$M(y,\omega) \approx \overline{M} + \varepsilon \sum_{i=1}^{n} \sqrt{\lambda_i}\,\xi_i(\omega)\,\psi_i(y) \tag{61}$$

In the present example, a unit correlation $b = 1$ equal to the airfoil chord length was chosen, $\overline{M} = 0.8$, $\varepsilon = .01$, and the random field approximated by a four-term truncated $K-L$ expansion.

Uncertainty calculations with four random variable dimensions were performed using both sparse and dense tensor product quadratures. Calculations using $L = 4$ sparse Clenshaw-Curtis quadrature required 137 realizations, and calculations using $L = 4$ dense tensor product Clenshaw-Curtis HYGAP approximation required 6,561 CFD evaluations. Figure 34 compares surface pressure coefficient statistics computed using both dense and sparse tensor product quadratures. The sparse tensor product statistics in Figure 34(b) show small oscillations on the lower surface. These spuri-



(a) (b)

Figure 34. Reynolds-averaged Navier-Stokes flow over a NACA 0012 airfoil with correlated random field inflow data. Shown are surface pressure coefficient mean and standard deviation statistics using (a) $L = 4$ dense tensor product Clenshaw-Curtis HYGAP quadrature and (b) $L = 4$ sparse tensor product Clenshaw-Curtis quadrature.

ous oscillations are due to non-smoothness in the random variable dimensions which significantly

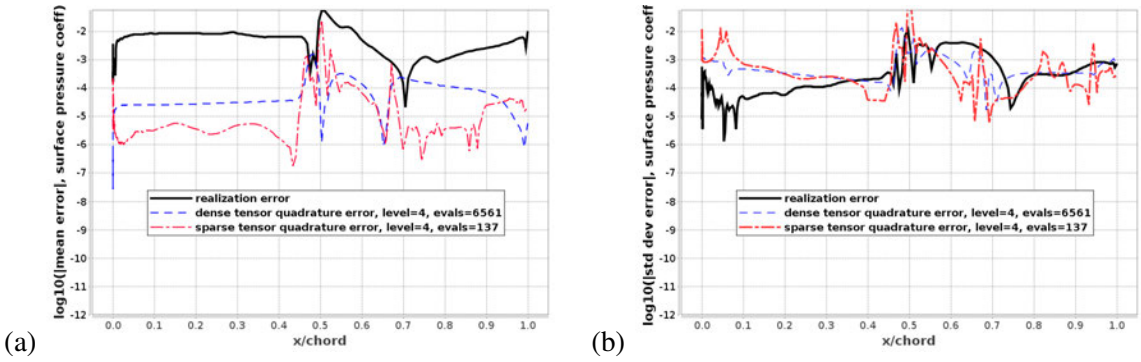


Figure 35. Navier-Stokes flow over the NACA 0012 airfoil with correlated random field inflow data. Shown are Type I estimates of surface pressure coefficient realization error and sparse/dense quadrature error for (a) mean and (b) standard deviation statistics on the airfoil upper surface.

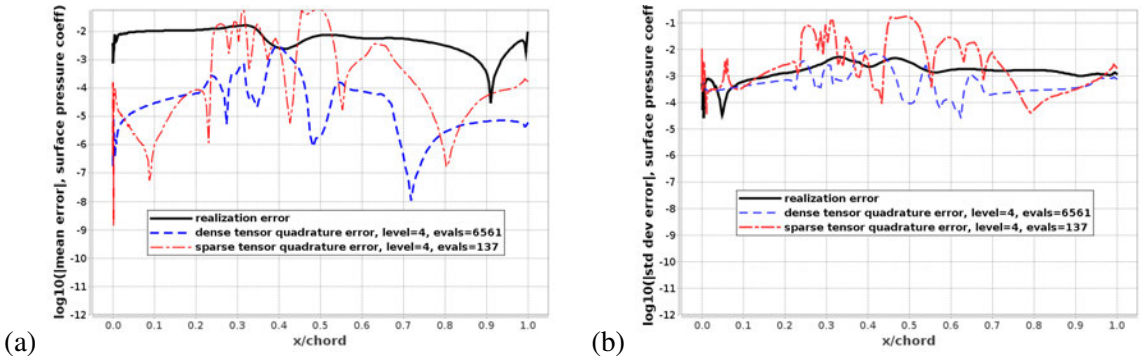


Figure 36. Navier-Stokes flow over the NACA 0012 airfoil with correlated random field inflow data. Shown are Type I estimates of surface pressure coefficient realization error and sparse/dense quadrature error for (a) mean and (b) standard deviation statistics on the airfoil lower surface.

deteriorates the accuracy of the sparse tensor product quadrature. These spurious oscillations are absent in the Figure 34(a) dense tensor product result because the HYGAP algorithm switches to piecewise polynomial approximation whenever the data becomes non-smooth in random variable dimensions. Examination of the Type I mean and standard deviation error graphs for dense and sparse tensor product quadratures for the upper airfoil surface in Figure 35 and the lower airfoil surface in Figure 36 further confirms the poor accuracy of the sparse tensor product quadrature due to non-smoothness of the underlying statistics integrands. In contrast, the dense Clenshaw-Curtis HYGAP quadrature results shown in figures 35 and 36 show acceptable quadrature error levels, but the computational cost is significantly higher. These calculations show considerable room for improvement in efficiency.

7 Concluding Remarks

Combined uncertainty and error bound estimates provide a quantitative guide when performing practical CFD calculations with uncertainty by:

- Quantifying the overall accuracy of computed output moment statistics.
- Quantifying the impact of statistics integral quadrature error on the accuracy of computed moment statistics.
- Quantifying the impact of realization error on the accuracy of computed moment statistics.
- Providing a systematic procedure for determining whether additional resources should be devoted to solving realizations more accurately (finer grids) or improving the accuracy of computed moment statistics quadratures (more parameter evaluations).

References

1. A. Aitken. On Bernoulli's numerical solution of algebraic equations. *Proc. Soc. Edinburgh*, 46:289–305, 1926.
2. I. Babuska, F. Nobile, and R. Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM J. Numer. Anal.*, pages 1005–1034, 2007.
3. B. S. Baldwin and T. J. Barth. A one-equation turbulence transport model for high Reynolds number wall-bounded flows. Technical Report TM-102847, NASA Ames Research Center, Moffett Field, CA, August 1990.
4. T. J. Barth. Numerical methods for gasdynamic systems on unstructured meshes. In Kröner, Ohlberger, and Rohde, editors, *An Introduction to Recent Developments in Theory and Numerics for Conservation Laws*, volume 5 of *Lecture Notes in Computational Science and Engineering*, pages 195–285. Springer-Verlag, Heidelberg, 1998.
5. T. J. Barth. Simplified discontinuous Galerkin methods for systems of conservation laws with convex extension. In Cockburn, Karniadakis, and Shu, editors, *Discontinuous Galerkin Methods*, volume 11 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Heidelberg, 1999.
6. T. J. Barth. A-posteriori error estimation and mesh adaptivity for finite volume and finite element methods. In T. Plewa, T. Linde, and V. Weirs, editors, *Adaptive Mesh Refinement: Theory and Applications*, volume 41 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Heidelberg, 2004.
7. T. J. Barth. On the propagation of statistical model parameter uncertainty in CFD calculations. *Theor. Comput. Dyn.*, 26:435–457, 2012.
8. T. J. Barth. Non-intrusive uncertainty propagation with error bounds for conservation laws containing discontinuities. In *Uncertainty Quantification*, volume 92 of *Lecture Notes in Computational Science and Engineering*, pages 1–55. Springer-Verlag, Heidelberg, 2013.
9. T. J. Barth and M. Larson. A posteriori error estimation for adaptive discontinuous Galerkin approximations of hyperbolic systems. Technical Report NAS-99-010, NASA Ames Research Center, 1999.

10. T. J. Barth and M. Larson. A-posteriori error estimation for higher order Godunov finite volume methods on unstructured meshes. In Herbin and Kröner, editors, *Finite Volumes for Complex Applications III*, pages 41–63. Hermes Science Pub., London, 2002.

11. V. Barthelmann, E. Novak, and K. Ritter. High dimensional polynomial interpolation on sparse grids. *Advances in Computational Mathematics*, 12:273–288, 2000.

12. R. Becker and R. Rannacher. Weighted a posteriori error control in FE methods. In *Proc. ENUMATH-97, Heidelberg*. World Scientific Pub., Singapore, 1998.

13. H. Brass. Error bounds based on approximation theory. In T. Espelid and A. Genz, editors, *Numerical Integration*, pages 147–163. Kluwer Academic Pub, Dordrecht, 1992.

14. C. W. Clenshaw and A. R. Curtis. A method for numerical integration on an automatic computer. *Numer. Math.*, 2:197–205, 1960.

15. B. Cockburn, S. Hou, and C. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case. *Math. Comp.*, 54:545–581, 1990.

16. B. Cockburn, M. Luskin, C.-W. Shu, and E. Süli. Enhanced accuracy by postprocessing for finite element methods for hyperbolic equations. *Math. Comp.*, 72:577–606, 2003.

17. K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. Introduction to numerical methods for differential equations. *Acta Numerica*, pages 105–158, 1995.

18. T. Gerstner and M. Griebel. Numerical integration using sparse grids. *Numerical Algorithms*, 18(3–4):209–232, 1998.

19. C. Harris. Two-dimensional aerodynamic characteristics of the NACA 0012 airfoil in the Langley 8-foot transonic pressure tunnel. Technical Report TM-81927, NASA Langley Research Center, Hampton, VA, 1981.

20. R. Hartmann and P. Houston. Adaptive discontinuous Galerkin methods for the compressible euler equations. *J. Comp. Phys.*, 182(2):508–532, 2002.

21. D. Jespersen, T. Pulliam, and P. Buning. Recent enhancements to OVERFLOW. Technical Report 97-0644, AIAA, Reno, NV, 1997.

22. G. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. *J. Comp. Phys.*, pages 202–228, 1996.

23. M. Larson and T. Barth. A posteriori error estimation for adaptive discontinuous Galerkin approximations of hyperbolic systems. In Cockburn, Karniadakis, and Shu, editors, *Discontinuous Galerkin Methods*, volume 11 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Heidelberg, 1999.

24. P. LeSaint and P. Raviart. On a finite element method for solving the neutron transport equation. In C. de Boor, editor, *Mathematical Aspects of Finite Elements in Partial Differential Equations*, pages 89–145. Academic Press, 1974.

25. N. Metropolis and S. Ulam. The Monte Carlo method. *J. Amer. Stat. Assoc.*, 44(247):335–341, 1949.

26. H. Niederreiter. Low-discrepancy and low-dispersion sequences. *J. Num. Th.*, 30:51ąV70, 1988.

27. E. Novak and K. Ritter. High dimensional integration of smooth functions over cubes. *Numer. Math.*, 75(1):79–97, 1996.

28. E. Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33:1065–1076, 1962.

29. T. Patterson. The optimum addition of points to quadrature formulae. *Math. Comp.*, 22:511–520, 1993.

30. S. Prudhomme and J. Oden. On goal-oriented error estimation for elliptic problems: application to the control of pointwise errors. *Comp. Meth. Appl. Mech. and Eng.*, pages 313–331, 1999.

31. W. H. Reed and T. R. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos National Laboratory, Los Alamos, New Mexico, 1973.

32. L. Richardson and J. Gaunt. The deferred approach to the limit. *Trans. Royal Soc. London, Series A*, 226:299–361, 1927.

33. M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *Annals of Mathematical Statistics*, 27:832–837, 1956.

34. V. Schmitt and F. Charpin. Pressure distributions on the ONERA M6 wing at transonic mach numbers. Technical Report AGARD AR-138, Advisory Group for Aerospace Research and Development, 1979.

35. T. Sclafani, J. Slotnick, J. Vassberg, T. Pulliam, and H. Lee. Overflow analysis of the nasa trap wing model from the first high lift prediction workshop. Technical Report 2011-0866, AIAA, Orlando, FL, 2011.

36. J. Slotnick, J. Hannon, and M. Chaffin. Overview of the first aiaa cfd high lift prediction workshop. Technical Report 2011-0862, AIAA, Orlando, FL, 2011.

37. R. Smith. *Uncertainty Quantification*. SIAM Pub, Philadephia, 2014.

38. S. Smolyak. Quadrature and interpolation formulas for tensor products of centain classes of functions. *Dok. Akad. Nauk SSSR*, 4:240–243, 1993.

39. M. A. Tatang. *Direct Incorporation of Uncertainty in Chemical and Environmental Engineering Systems*. PhD thesis, MIT, 1994.

40. B. van Leer. Towards the ultimate conservative difference schemes V. A second order sequel to Godunov's method. *J. Comp. Phys.*, 32:101–136, 1979.

41. B. van Leer. Upwind-difference schemes for aerodynamics problems governed by the Euler equations. volume 22 of *Lectures in Applied Mathematics*. AMS Pub., Providence, Rhode Island, 1985.

42. G. Wasilkowski and H. Woźniakowski. Explicit cost bounds on algorithms for multivariate product problems. *J. Complexity*, 11:1–56, 1995.

43. O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery and a posteriori error estimates. Part I: the recovery technique. *Int. J. Numer. Meth. Engrg.*, 33:1331–1364, 1992.

The Modern Design of Experiments

Richard DeLoach

NASA Langley Research Center, Hampton, VA 23681

Nomenclature

Alternative Hypothesis	Proposition that some specified effect is large enough to be of interest: "Differences between response estimates by a complex CFD code and by a simpler proposed candidate response surface model are too great to be acceptable."
CFD	Computational Fluid Dynamics
CLT	Central Limit Theorem
CPU	Central Processing Unit
Critical Path	The most time-consuming sequence of tasks that must be performed sequentially to complete a project
Design Matrix	Extension of a test matrix to include a column for each term in the response model
Design Space	A coordinate system in which each axis corresponds to a different factor.
Explained Variance	Measure of response variation induced by known changes in factor levels
Factor	An independent variable, typically varied in an experiment
Graduating Function	Formula for predicting system responses over a constrained range of factor levels
Hidden Replication	A property of p-parameter response models fitted to $N > p$ data points that improves precision through partial cancelation of model residuals
Hierarchy	Condition in which higher order terms are accompanied by component lower-order terms
Inference	Decision to reject either a null hypothesis or its alternative
Independent Variable	Factor upon which a given response depends; intentionally varied in an experiment
Inference Space	Another term for design space
Lack-of-fit	Systematic computational response error
Level	Specific value of a factor
MDOE	Modern Design of Experiments
Null Hypothesis	An unproved proposition to be used as a basis for the argument: "There is no difference between response estimates by a complex CFD code and by a simpler response surface model."
OFAT	One-factor-at-a-time method, a common conventional design strategy
Population	An infinite (or complete) set of response estimates; usually a theoretical abstraction
Regressors	Response model terms beyond intercept; there are $p - 1$ regressors for a p-term model
Repeat Points	Duplicate points acquired with relatively little change in experimental error from point to point; contrast with Replicate Points

Replicate Points	Duplicate points acquired with as much chance of a change in experimental error from point to point as possible; contrast with Repeat Points
Residual	Difference between a response estimate and a specified reference
Residual Variance	See Unexplained Variance
Response	A metric that describes some aspect of a system when factors are set at prescribed levels
Response Surface Methods	Process by which responses are represented as functions of specified factors
Response Surface Model	A mathematical function describing a given system response in terms of factor levels
RSM	Response Surface Model(s) or Modeling, Response Surface Methods
RTV	Room Temperature Vulcanized
Sample	A finite set of response estimates
Significance Level	Probability of wrongly rejecting a null hypothesis
Site	A point in a design space corresponding to a unique combination of factor levels
Site Selection	Process of defining which specific points to include in a test matrix
Statistically Significant	Detectable with a prescribed probability
Sum of Squares	A measure of variance; computed by adding all the squared residuals of a data sample
TAEM	Terminal Area Energy Management; Space Shuttle landing regime
Test	Synonymous with "experiment"
Test Matrix	Array of factor levels for which responses are to be estimated in an experiment
Unexplained Variance	Response variability that cannot be attributed to known factor level changes.
Variance	Measure of degree to which individual points in a sample differ from some reference

Symbols

df	Degrees of freedom
$df_{Explained}$	Explained degrees of freedom
df_{Total}	Total degrees of freedom
$df_{Unexplained}$	Unexplained degrees of freedom
L	Least Squares Function; sum of squared model prediction residuals
N	Sample size
SS	Sum of squares
$SS_{Explained}$	Explained sum of squares
SS_{Total}	Total sum of squares
$SS_{Unexplained}$	Unexplained sum of squares
y_i	Value of the i^{th} response estimate
$\bar{y}$	Mean of a sample of response estimates
α	Maximum acceptable probability of erroneously rejecting the null hypothesis
β	Maximum acceptable probability of erroneously rejecting the alternative hypothesis

λ	Smallest residual that can be detected with a probability of $1-\beta$, given a probability no greater than α of erroneously rejecting the null hypothesis
λ_0	Smallest residual that can be detected with specified inference error probabilities, absent residual degrees of freedom
σ	Standard error
σ^2	Variance

1. Introduction

This chapter summarizes basic tenants of professional experiment design that can be applied to computational experiments as readily as they can be applied to the physical experiments for which they were originally developed. The principal distinction between computational and physical experiments is that system responses are estimated by calculation using an underlying computer code in the former and by direct measurement in the latter. This chapter assumes that formal experiment design is largely outside the common experience of the reader, and it therefore features more of a tutorial flavor than other chapters in this document. In order to exploit the topic fully, the reader will need to consult the rich literature of formal experiment design, from which selected references are cited at the end of this chapter. The objective of this chapter is simply to provide a general roadmap for how to proceed, and to suggest that efforts to develop a greater familiarity with the formal design of experiments are likely to pay dividends.

Central to a computational experiment is an underlying computer code describing some system of interest. We assume the code will map a vector of factor levels (values of a set of independent variables) into corresponding values of one or more response variables. We also assume that the code is sufficiently complex that there is no cost-effective way to readily reveal the effects and interactions of all of the independent variables. It is therefore useful to establish some approximating pattern of associations between factor levels and response levels to achieve this end.

A tabular approach is the most conceptually straightforward way of establishing these relationships between factor levels and corresponding responses. In this approach, tables are constructed that feature relevant combinations of factor levels, for which corresponding response estimates are recorded in a database. The expense of estimating each response need only be incurred one time, after which the database can be consulted by multiple users to support numerous applications, none of which need be specified before the database is constructed.

For the sake of illustration, let us say that we have a CFD code describing the aerodynamics of a particular aircraft, and that the code is capable of predicting lift, drag, pitching moment, etc. for specified levels of six factors: angle of attack, angle of sideslip, Mach number, Reynolds number, flap deflection, and aileron deflection. We wish to know the forces and moments for all combinations of ten levels of each of these six factors. There are 10^6 such combinations, which we describe as sites in a design space consisting of a coordinate system for which each axis represents a factor, or independent variable. The design space spans the range of factor levels over which responses are to be estimated, and is also known as an inference space.

Resource constraints would probably preclude running the code as many times as this example would require, but one might run it for a smaller number of factor combinations that are strategically selected to enable adequate response estimates for all factor combinations of interest. A regression model could be fitted to the smaller number of points and used to predict responses elsewhere. We use the term “response surface modeling” (RSM) to refer to such a process of

establishing mathematical relationships between factor levels and response estimates over a given design space. While there are other types of designed experiments, this chapter will focus on response surface modeling methods. Linear regression, a common technique for developing RSM models, is reviewed in a subsequent section in this chapter.

An RSM experiment can be organized into a design phase, an execution phase, and an analysis phase. In the design phase, a candidate model is specified that is believed suitable for making adequate predictions for a given response. The selection of the candidate model is generally based on a combination of subject matter expertise and experience. The model will be evaluated in the analysis phase and either augmented or replaced as necessary.

A minimum number of data points is estimated in the design phase that is sufficient to fit the candidate response model and to ensure precision requirements for model predictions. Precision requirements are also specified in the design phase, as well as maximum acceptable inference error risk probabilities, which will be discussed further later in the chapter. The distribution of these points within the design space of the experiment is then specified in order to achieve desirable effects, such as minimizing the uncertainty of the response model regression coefficients, ensuring relatively low prediction errors in preferred regions of the design space, or minimizing how changes in one factor level are correlated with changes in another.

During the execution phase of the experiment, data are acquired at the sites specified during the design phase. In a physical experiment, a number of quality assurance tactics are normally implemented during execution. This is because there are numerous sources of time-varying bias error in a physical experiment; due to temperature effects, instrument drift, operator fatigue or operator learning effects, atmospheric effects, desiccation effects, and so on. The basic principle behind these quality assurance tactics is to decouple the intentionally induced response level changes of the test plan from changes that occur as a result of circumstances that are not controlled in the experiment. These tactics are beyond the scope of the present document but are described in any elementary text on formal experiment design.

In the analysis phase, the response model is fitted to the data. The model might be optimized by transforming the response variable or one (or more) of the independent variables. Based on the analysis, a different model may be fitted to the data—say, a higher order model. The model is also validated in the analysis phase, by ensuring that there is an acceptably high probability that it predicts adequately at given sites in the design space.

Such models can be developed for a fraction of the cost of directly filling an entire database, and the response predictions they generate normally feature less uncertainty than single-point response estimates directly generated through high-volume data collection. This quality improvement results from a partial cancelation of errors that occurs when multiple data points are used to fit the model. Absent costly replication in a physical experiment, the error from a single-point response estimate is not similarly reduced when a database is simply filled. Of course, the uncertainty at a given site in the design space of a computational experiment cannot be reduced—or even detected—with replication. There are resulting special considerations that are treated in Section 2.

Some specific distinctions between conventional and designed experiments can be illustrated with a variation of an example attributed to Hotelling [2] and summarized in Cuthbert Daniel's book on experiment design [1]. Assume you are charged with weighing two objects, *A* and *B*, using a double-pan balance. We will demonstrate a little later how measurement precision is related to data volume, but for now let us simply stipulate that two measurements are required to provide

enough precision to meet the objectives of this test. That is to say, each result must be expressed as the average of two measurements. How should we proceed?

The conventional experimentalist would place *A* in one pan and record the counterweight needed to balance the pans, repeat this measurement a second time, and average the two results. He would then do the same with *B*, which would result in the required precision at the cost of four runs.

A designed experiment might begin by weighing *A* and *B* together, resulting in an estimate of the sum of *A* and *B*. Call this sum *S*. In a second weighing, the heavier object is placed in one pan and the lighter object in the other, with enough of the counterweights added to the latter to balance the pans. The added weight represents the difference between *A* and *B*. Call this difference *D*. It is now possible to estimate the weights of *A* and *B* with the same two-measurement precision but at half the cost—two runs instead of four. Let us say that $A > B$ so that $S = A + B$ and $D = A - B$. Then $A = \frac{1}{2}(S + D)$ and $B = \frac{1}{2}(S - D)$. The precision requirement is satisfied by the fact that the weight of each object has been determined by averaging two measurements, *S* and *D* in one case and *S* and *-D* in the other.

This simple example illustrates one of the principal advantages of formal experiment design over conventional data acquisition; namely, that it packs more information into each data point so that fewer points are needed to achieve a given objective. In this example, each data point in the conventional experiment extracted information about only one factor, while in the designed experiment, each point extracted some information about each of two factors. Since information on twice as many factors was embedded in each measurement, only half the measurements were needed in this case. Data volume reductions of a factor of two or more are common in more elaborate designed experiments as well. Since the acquisition of data consumes resources—direct operating costs and time—a significant reduction in data volume can translate directly into substantial cost savings.

The double-pan weighing experiment also illustrates three general characteristics of formally designed experiments that may temper the enthusiasm that conventional experimentalists might otherwise feel for the cost savings and the other benefits available with this approach. The first such characteristic is that formal experiment design generally requires more planning than conventional experimentation. The resource savings achieved in the double-pan experiment depended on the sums and differences scheme used to analyze the data, which had to be worked out in advance.

Secondly, no intermediate results were available in the designed experiment before all the planned points were acquired. With the conventional data acquisition approach, a preliminary estimate of the weight of one of the objects would have been available after the first measurement. Depending on the order, by the second measurement either the final estimate of the weight of the first object would have been in hand, or a preliminary estimate of each of them would have been available. By the third measurement, the final result for one object would have been available and a preliminary estimate would have been available for the other. Some idea of the direction of the experiment could have been had at every step of the way.

The designed experiment provided no insight into the weight of either object until all of the planned data were in hand. Results of designed experiments are obtained from samples of data rather than from individual data points. A more complex experiment might entail the acquisition of several individual samples of data, for which preliminary results are unavailable until a

complete sample is acquired. More patience is therefore needed to execute a designed experiment than a conventional test focused on data collection.

The third characteristic of designed experiments that may give conventional experimentalists pause is that some mathematical analysis of the data is inevitably required to unpack the higher density of information imparted to each measurement when more than one factor at a time is examined. In the conventional variation of the pan-balance experiment, each measurement involved *A* or *B* alone, and the desired information was manifested in the raw data itself. Each measurement in the designed experiment entailed both *A* and *B* together, so information on each object separately was not available until some mathematical analysis was performed on the data. More complex experiments require more advanced analytical techniques such as regression and analysis of variance to unpack the information in a designed experiment. This adds to the cognitive workload, and may require the experimentalist to acquire additional skills.

To summarize, the double-pan weighing experiment demonstrates how tangible benefits can be achieved with formal experiment design. In addition to the cost savings illustrated with this example, designed experiments can also ensure a prescribed level of precision. While not illustrated in such a simple experiment as this, designed experiments also provide greater insights into the physical phenomenon under study than are readily available with high-volume data collection methods. However, it should come as no surprise that these benefits are only realized with some cost. This cost takes the form of additional planning in the design phase of the experiment, additional patience and discipline during execution, and additional cognitive workload in the analysis.

Computational experiments differ from physical experiments in that the unexplained variance in response estimates cannot be detected by replication. This is true at least for common deterministic (non-stochastic) codes, because for such codes, there is no variance among replicates as there is in physical experiments in which random error imposes itself with each new point. This simply means that in the case of deterministic computational experiments, some other method must be used to estimate it. Section 2 reviews some sources of unexplained variance in a computational experiment and considers practical ways of estimating it.

Section 3 describes how accuracy and precision specifications drive minimum data volume requirements in a formally designed experiment.

Section 4 reviews site selection, the process of specifying particular factor/level combinations in order to impart certain desirable characteristics to the distribution of prediction errors within the design space of a formally designed experiment. Representative site distributions are displayed that make the distribution of prediction errors radially or azimuthally symmetric, or that minimize the average variance of the response model coefficients, reduce correlations among independent variables, or minimize the integral of the prediction variance over the design space. The advantages of these outcomes are briefly reviewed.

Section 5 reviews selected highlights from linear regression and introduces a useful relationship between response model prediction variance and the number of points fitted to a given regression model.

Quality assurance is described in Section 6. In this section, the volume of data specified in Section 3, distributed within the Design Space in Section 4, and fitted to a response model in Section 5 culminates in results that are evaluated in Section 6.

Concluding remarks are offered in Section 7.

2. Variance and Quality in a Computational Experiment

Every sample of experimental data, either from a physical experiment or a computational experiment, exhibits variance except for the uninteresting case in which all points are identical. Most of the total variance is induced intentionally by changing independent variables during the normal course of executing the experiment. It is suggested that such intentionally induced variance is explained by changes made according to the test matrix. Ideally, all the variance would be explained, but some residual or unexplained variance always remains after the intentional factor changes are taken into account. It is this unexplained variance that is responsible for uncertainty in experimental results.

Traditionally, stability of data—that is, low variance of data in genuine replicates—has been regarded as a measure of quality in physical experiments. Uncertainty in the estimate of a response at a given site in the design space is typically quantified by some metric that is proportional to a standard error, σ, which is the square root of the variance at that site.

Quality in a designed experiment is associated less with variance in the raw data and more with variance in response predictions. A high-quality designed experiment results in system response predictions that are consistently within some tolerance that is specified during the design of the experiment. This interval is assumed to contain an estimate of truth, although it might be biased. For a fixed volume of data, the interval is greater when there is greater variance in the raw data, but for a given degree of variance in the measurement environment, its width can be reduced by acquiring more data. Data quality is therefore more of a cost issue than a quality driver in a designed physical experiment, since greater precision can generally be achieved by expending the additional resources needed to acquire extra data.

Considerable cost savings can be achieved by understanding and exploiting this connection between quality and data volume in a designed experiment. The basic idea is to define specific precision requirements early in the design process, to compute the volume of data necessary to achieve those requirements, and to halt the expenditure of additional resources when that volume of data has been acquired plus a little more, perhaps, for comfort's sake. The precision associated with the data in hand is then assessed, and additional data are acquired if necessary, but only if necessary.

The process of scaling the experiment is described in the Section 3 of this chapter. Scaling refers to the process of estimating the volume of data required to fit a specified model in a given measurement environment. The volume of data must be great enough to ensure that prescribed precision requirements are achieved and that there will not be an unacceptably high risk that significant lack-of-fit error will go undetected. These concepts are developed more fully in the next section. Before discussing scaling, we describe a few categories of unexplained variance.

2.1. Coding Variance

Since quality is so often associated with low data variance conventional physical experiments, there can be some confusion about the role that quality assessment plays in the design of a computational experiment featuring deterministic codes, in which replicates do not display any variance. (We neglect the uninteresting case in which fluctuations in the least significant bit might induce some variance in replicates.) The fact that no variance is observed in computational

replicates acquired at a given design space site cannot be construed to mean that there is no uncertainty in the response estimated at that site. It is important to distinguish between a state in which there is no variance at a given site in the design space, and one in which the variance simply cannot be conveniently detected there. That is, it is important to distinguish between the uncertainty that is always present, and evidence of uncertainty, such as variance in replicates, that is not observed in a computational experiment.

Recall that a random variable such as a system response estimate is characterized by location and dispersion. Location quantifies the value of the response, while dispersion quantifies the uncertainty in that location estimate. One can quantify location with a single data point or estimate dispersion with respect to a specified reference with one data point. However, it requires at least two points to quantify both location and dispersion. A non-replicated response estimate has only one degree of freedom, which is consumed in estimating its location. There are no additional degrees of freedom available in a single-point measurement to estimate dispersion, so the uncertainty in the location estimate cannot also be quantified from that single point. However, the fact that dispersion cannot be estimated by an unreplicated measurement does not imply that there is no uncertainty associated with that measurement. It means only that one must use some other method besides replication to estimate the unexplained variance—perhaps an appeal to historical data or to information external to the experiment. The same is true in a computational experiment.

The concept of "*N* genuine replicates" that permits random error to be assessed conveniently in a physical experiment could be accommodated in principle in a computational experiment by considering a large number of independently developed response models, each producing a somewhat different response estimate for the same site in the design space. This sample of "*N* different computational response estimates" can be regarded as having been drawn at random from a population of computational estimates with some distribution. Sample statistics such as the standard deviation could be used to estimate the variance of such a population, just as in a conventional physical experiment.

We do not evaluate uncertainty in this way in a computational experiment because it is impractical to do so, but surely there would be a distribution of different response estimates produced by a sample of different codes if we did do this. This distribution would allow a direct estimate of the uncertainty associated with what might be called "random coding variance," a component of the total response prediction variance that would be analogous to the random error that replication reveals in a physical experiment. This coding variance is present whether or not we can conveniently quantify it.

One might estimate the coding variance in a computational experiment based on experience and subject matter expertise, then use the underlying code to generate an estimate of the response that is assumed to be unbiased and add to that a random error estimate drawn from the expected distribution of coding errors. Absent any information to the contrary, one might appeal to the Central Limit Theorem to use a Gaussian distribution that has a mean of zero and a specified variance. Response estimates used to fit the response model can be constructed as the algebraic sum of this simulated random error and the response prediction of the underlying code.

2.2. Set-Point Variance

Set-point variance is another component of total response prediction variance that exists despite the lack of variance in computational replicates. Response surface models developed in computational experiments can also be used to propagate set-point errors into uncertainty in the

response estimates with considerable efficiency. Consider a response, y, which depends on k factors:

$$y = f\left(x_1, x_2, \cdots x_k\right) \tag{1}$$

If errors in estimating the k factors are independent of each other, the response variance attributable to set-point uncertainty can be estimated as follows:

$$\sigma_y^2 = \left(\frac{\partial y}{\partial x_1}\right)^2 \sigma_{x_1}^2 + \left(\frac{\partial y}{\partial x_2}\right)^2 \sigma_{x_2}^2 + \cdots + \left(\frac{\partial y}{\partial x_k}\right)^2 \sigma_{x_k}^2 \tag{2}$$

where σ_{x_i} is the standard error in estimating the i^{th} factor level. Consider as an example a rectangle of length L and width W, as in Fig. 1.

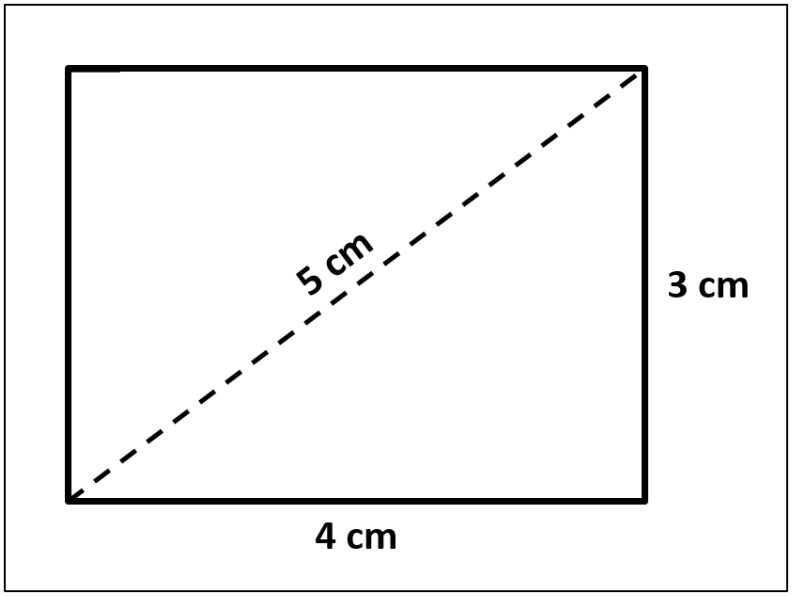


Figure 1. Rectangle 3 cm × 4 cm. Area is 12 cm^2. If the standard error of the length and width is 0.1 cm, then the standard error of the area is 0.5 cm^2.

The uncertainty in the area (A) estimate due to the uncertainties in estimating L and W is computed using Eq. 1, as follows:

$$A = LW \rightarrow \sigma_A^2 = \left(\frac{\partial A}{\partial L}\right)^2 \sigma_L^2 + \left(\frac{\partial A}{\partial W}\right)^2 \sigma_W^2 = \left(W^2 + L^2\right)\sigma^2 \rightarrow \sigma_A = \sigma\sqrt{W^2 + L^2} \tag{3}$$

where the subscripts have been dropped from the factor variances since the errors in estimating length and width are assumed to be the same. Thus, the standard error in estimating the area is just the standard error in estimating either linear dimension, scaled by the diagonal of the rectangle.

Assume that the standard error in estimating each dimension is 0.1 cm. In this example with W = 3 cm, L = 4 cm, and σ = 0.1 cm, the standard error in the area of 12 cm^2 is 0.5 cm^2, or about

4%, compared to standard errors of about 3.3% and 2.5% for *W* and *L*, respectively. As noted, polynomial response models commonly encountered in response surface modeling experiments are especially well suited to this type of factor error propagation.

A chicken-and-egg problem associated with set-point error estimation should be acknowledged here. From an experiment design perspective, the driving motivation for estimating total response variance is to properly scale the experiment by specifying an ample volume of data to drive prediction uncertainty low enough, given that variance in the sample of fitted data. However, to propagate set-point error by the methods described in this document it is necessary to begin with a response model, which must be developed from a sample of response estimates of some size. Thus, the sample must be collected before its optimum size can be precisely determined.

The solution to the chicken-and-egg problem is to develop an initial model from a sample that is scaled (by methods described in Section 3 of this chapter) using a conservative estimate of the contribution of propagated set-point error to the total sample variance. Such an estimate must rely on experience and judgment. Use that model to propagate set-point error into prediction variance, and compare with the initial estimate. If the initial set-point error estimate was larger than necessary, it simply means that the experiment design will have been over-scaled by specifying more data than necessary to satisfy precision requirements. The response model prediction uncertainty would then be smaller than required to meet specifications—not really a problem except that an adequate result could have been achieved with fewer resources. If, on the other hand, the set-point variance propagated with Eq. 2 is significantly greater than the estimate used to scale the experiment, the scaling should be repeated with the larger total variance estimate to determine how many additional points should have been acquired. The experiment can then be augmented by acquiring additional data.

2.3. Spatial Variance

The concept of temporal variance, by which unexplained response variations occur in a physical experiment over time at a fixed site in the design space, is a familiar one. Each point in Fig. 2 is a measurement of what is ostensibly the same response. These are therefore ordinary replicates, with all point-to-point variation attributed to ordinary random error. The net random error of each point is assumed to be the algebraic sum of multiple errors attributable to various mechanisms, none of which is necessarily known. However, given very mild conditions that are met in most practical circumstances, the Central Limit Theorem asserts that such random errors will behave as if they have been drawn from a normal distribution.

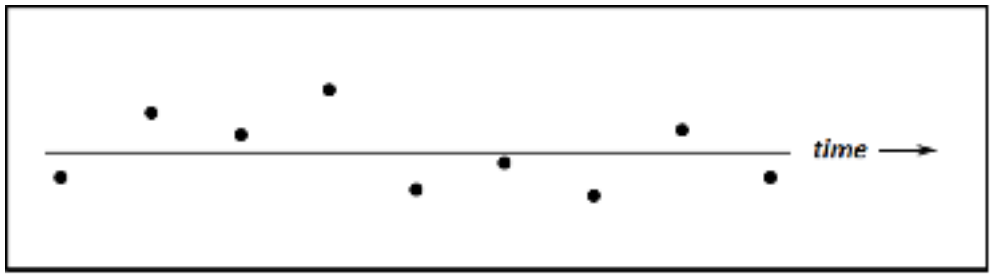


Figure 2. Ordinary temporal variance. Response estimates at a fixed site in the design space acquired at various times in a physical experiment. Solid line represents the mean.

Variance in the data of Fig. 2 is quantified by squaring the difference between each point and some constant reference, usually the sample mean. Those squared differences are added to generate a sum of squares (*SS*). The sum of squares is then divided by the number of degrees of

freedom (*df)*, which is the minimum number of data points required to estimate the *SS*. When the sample mean is known, the number of degrees of freedom is just the sample size minus one, since the known sample mean can be multiplied by the known sample size to determine the sum of all the responses, and any individual response can be determined by subtraction if *N – 1* of them are known. The total variance of an *N*-point sample of physical replicates is, then

$$\sigma^2 = \frac{SS}{df} = \frac{\sum_{i=1}^{N} \left(y_i - \bar{y} \right)^2}{N - 1} \tag{4}$$

where $\bar{y}$ is the sample mean and y_i is the i^{th} response estimate.

In a computational experiment there is no variance in a sample of data acquired over time at a fixed site in the design space, but there will be variance in a sample of data acquired over space, (i.e., over the design space). In theory, this spatial variance could be explained with a perfectly fitted regression model, but in the absence of a perfect model there is always some residual unexplained variance even if the fit is generally adequate, whether the experiment is physical or computational.

This unexplained variance can be attributed to a number of random and systematic factors. The approximate nature of the regression model will invariably introduce some degree of systematic lack-of-fit bias error, but even if the overall fit is adequate, the fitted response model might somewhat overstate the underlying code predictions at some sites in the design space and somewhat understate them elsewhere. These localized imperfections, however small, can produce residuals similar to those attributed to a combination of systematic modeling error and random experimental error in a physical experiment.

Consider a sample of data from a computational experiment consisting of *N* non-replicated points that have been fitted by linear regression to the following first-order response model in one factor:

$$y = b_0 + b_1 x \tag{5}$$

Assume that the fit is adequate; that is, that there are no systematic departures of the response model from the data. The quantities *x* and *y* are the usual independent and dependent variables, respectively, and the b_i are regression coefficients determined by methods that will be reviewed in Section 5 of this chapter. Figure 3 displays the plotted data, plus the fitted response model of Eq. 5.

The data in Fig. 3 bear the same relationship to the fitted first-order response model as the replicated points in Fig. 2 bear to the sample mean. That is to say, the fitted data in Fig. 3 display variance with respect to a designated reference just as in Fig. 2, even though no individual data point is replicated in Fig. 3. This is known as hidden replication. To illustrate hidden replication and the similarity between temporal and spatial variance, Fig. 3 was constructed by simply rotating Fig. 2. Residuals in Fig. 3 are represented by the vertical distance from each data point to the plotted response model used as a reference, rather than the perpendicular distance to the same reference. Except for this detail, they would be identical to the residuals of Fig. 2.

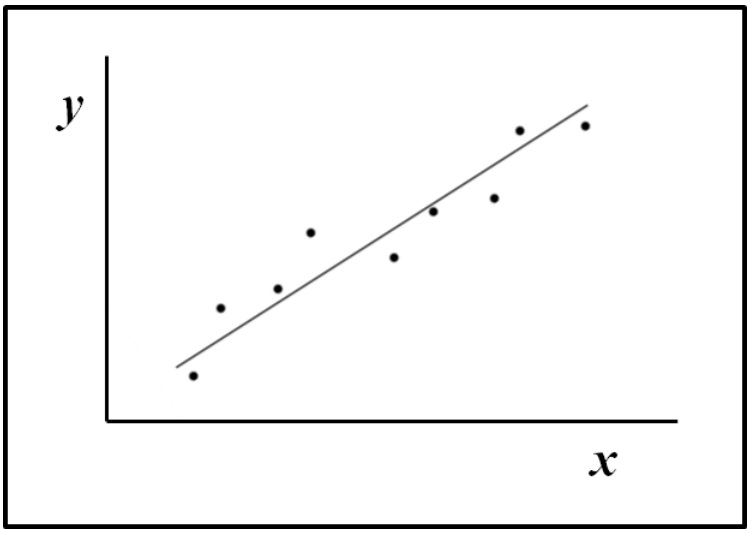


Figure 3. Spatial Variance. Points represent non-replicated response estimates at multiple sites in the design space. Solid line represents a response model fitted to the data.

Hidden replication occurs whenever N points are fitted to a model with p parameters with $N > p$. Hidden replication improves fitting precision through partial cancelation of model residuals, even when there is no replication of the individual points. The fitted response model utilizes information from response estimates at multiple sites in the design space, and the errors from these multiple points partially cancel even though no individual point is replicated. It can be shown that the average prediction variance of a fitted response model is reduced by a factor of p/N from the variance of single-point response estimates that are fitted, because of this hidden replication. This is true for computational experiments where replication is not possible, and it is also true in physical experiments where replication is possible, but not performed.

The unexplained variance displayed in Fig. 3 is computed by a formula analogous to Eq. 4:

$$\sigma^2_{Unexplained} = \frac{SS_{Unexplained}}{df_{Unexplained}} = \frac{\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2}{N - p} \tag{6}$$

where $\hat{y}_i$ is a response model prediction at the site of the i^{th} fitted data point. Just as one degree of freedom is consumed in estimating the mean of a sample of replicates so that only $N - 1$ degrees of freedom remain available to estimate the variance, so in this case are there $N - p$ degrees of freedom available to quantify the unexplained variance, since a response model with p terms consumes one degree of freedom for each term in the model. The intercept is included as one of the p terms.

The sum of squares term for the unexplained variance of a fitted sample of data is analogous to the sum of squares term for a sample of physical replicates. The reference used to compute the i^{th} residual in the former case is $\overline{y}$ and in the latter it is $\hat{y}_i$, representing the best response estimate in either case (the straight lines in Figs. 2 and 3).

2.4. Composite Variance Estimate

Various sources of uncertainty in a computational experiment have been discussed. The experimenter has some flexibility in deciding which of these to consider. It is not necessarily wrong to ignore some error sources, but the consequences should be understood.

Often the objective of a computational response surface experiment is simply to model the underlying code, and not necessarily the true response. This is the case when the assumption can be reasonably made that the underlying code is an adequate representation of the phenomenon under investigation. In such a case, one might be justified in ignoring uncertainty in the response predictions of the underlying code.

Factor set-point errors can likewise be ignored, along with the contribution they make to response model prediction variance. Again, this is not necessarily wrong, but it might be more prudent to invoke Eq. 2 to estimate the component of response variance due to set-point uncertainty, or to use some of the Monte Carlo methods described elsewhere in this volume to estimate it. Doing so can help justify the decision not to consider set-point uncertainty in the design of a computational experiment.

The spatial variance reflected in residuals at sites distributed throughout the design space can be quantified by Eq. 6. This captures the variance attributable to imperfections in how well the fitted response model corresponds to the underlying code. Care should be taken with fitted models for which the residuals are not normally distributed, indicating a statistically significant lack-of-fit error. If such an error is large compared to one's tolerance for error, the model should be rejected and another model fitted. If the model is a polynomial, then adding higher-order terms might produce a better fit.

If multiple types of unexplained variance such as those described above are estimated, they can be pooled by methods described in standard references [3, 4]. However, for the purposes of designing an experiment, it is not necessary—nor is it likely to be possible—to precisely estimate the total unexplained variance in a computational experiment for which direct estimates by replication are not available. One might simply model the unexplained variance instead. This approach is consistent with the notion that computational response estimates are in fact simulations of how we expect natural phenomena to behave. It is a relatively small extension of this concept to acknowledge that all response estimates are random variables that have associated with them some uncertainty, and that it is just as legitimate to simulate this uncertainty as it is to simulate the responses themselves.

By this approach, the underlying code can be used to make what is assumed to be an unbiased estimate of the true response at a given site in the design space. We might reasonably assume from the Central Limit Theorem that the net response error at that site will be drawn from a distribution that is normally distributed with a mean of zero and some prescribed standard deviation. One can use experience and subject matter expertise to estimate this standard deviation, or one can simply assume various levels of experimental error, designing separate experiments for different simulated measurement environments. This would shed light on the differences between inferences that can be supported from experiments conducted in relatively quiescent or volatile measurement environments. In any case, the simulation of experimental error in this way would simply entail estimating a measured response plus an error value drawn from a normal distribution chosen to represent the measurement environment, then summing the two for each site in the design space that is specified by the test matrix.

The entire machinery of formally designed physical experiments can then be brought to bear in a computational experiment. Variance in the measurement environment plays a central role in a designed experiment. The volume of data required to meet precision requirements and to ensure that the effects of lack-of-fit errors are detected is directly proportional to this variance, as will be seen in Section 3 of this chapter. Variance in the measurement environment also plays a key role in assessing the quality of a response model. By simulating the measurement environment, inferences can be made conditional on the degree of unexplained variance, which is in fact a realistic limiting factor.

3. Quantifying Data Volume Requirements: Scaling a Design Experiment

Unexplained variance, whether conveniently detectable through replication or not, is of interest in the planning stages of a designed experiment because it impacts minimum data volume requirements. The smaller the unexplained variance is, the smaller the sample of data required to satisfy specific precision objectives.

Test matrices can be developed with the fewest points (thus lowest cost) necessary to fit a specified response model within a given tolerance when the data are characterized by some unexplained variance, σ^2. For the purpose of designing a computational experiment, it makes no difference that σ^2 cannot be computed directly from measured replicates, since it can be estimated by pooling the variance estimates of the various error components noted in Section 2, or it can simply be simulated using subject matter expertise and experience to specify a reasonable degree of dispersion.

One can only justify the cost of acquiring the next data point if the value it adds exceeds the resources needed to acquire it. Average per-point costs can be estimated by conventional accounting methods that prorate labor and direct operating costs over the data sample, but it is not immediately obvious how to consider the value of the next point.

We can begin by noting that the next data point adds value only to the extent that it decreases the probability of making an erroneous inference about the system under investigation. In a response surface modeling experiment, this means that it is only necessary to acquire additional data if the prediction variance is not yet small enough to estimate responses within a specified tolerance and with a prescribed level of confidence. When enough data are in hand to predict responses within tolerance with the required level of confidence, and to ensure that lack-of-fit errors do not go undetected, we may say that we can adequately predict system behavior, and the cost of further data acquisition is no longer justified.

Such cost/benefit considerations are complicated by the fact that the value of the next new data point changes with the volume of data already in hand. The first few points acquired in an experiment are especially valuable because there is so little else upon which to base any inference about the behavior of the system under study. Additional data always adds value, but points added to an existing sample from the same design space will generally contribute less to the value of the experiment than the early points.

There is an exploitable consequence of this change in the value of data with the volume in hand. Since the value of new data decreases with the volume of data already in hand, the value added by the next point must eventually drop below the cost of acquiring it. In other words, there is a point of diminishing returns beyond which the net benefit of each new point is negative. One has certainly acquired ample data by the time the volume of the data at hand causes the next new

point to be worth less than the cost of acquiring it. Knowing this point of diminishing returns is useful in formulating a cost-minimal exit strategy for the experiment. Whenever model prediction variances exceed acceptable tolerance levels documented during the design of the experiment, additional data can always be acquired in an augmentation to the original test matrix. There is no real danger of acquiring too little data by an overly aggressive cost control focus that halts data acquisition too early.

This section will show that a surprisingly small sample of data is usually sufficient to achieve specified precision goals. An important corollary is that the traditional productivity strategy of maximizing data volume within a given resource budget simply ensures that costs are maximized. It is not productive to acquire data in such a high volume that much of the sample will cost more than the value added by the data. There may be experiments in which the cost of data acquisition is so low that a very large volume of data would have to be acquired before the value of additional data could be driven below the cost of acquiring it, but such cases are not the norm in large-scale aerospace research.

Specifying the minimum volume of data needed to meet precision goals in an important element of what we describe as quality assurance. Data volume considerations also play an important role in quality assessment, where the adequacy of a candidate response model is tested. To test the adequacy of a candidate model, it is necessary to compare its predictions with the true response, which is not generally known. We must therefore define some acceptable surrogate. For example, we may decide to use the mean of some number of unbiased measurements as a truth surrogate in a physical experiment, or possibly some prior result in which we have high confidence. As we will see later in this section, the volume of data necessary to develop an adequate response model depends on the nature of this truth surrogate.[1]

Neglecting bad data as the explanation (such as data from improperly calibrated instruments), lack-of-fit by an inadequate model is the most likely cause of errors that are too large to attribute to random variations in the data. For example, we might assume that the unknown underlying form of the response function can be adequately approximated by a Taylor series of some order that we specify from a combination of experience and subject matter expertise. Among other possible reasons, systematic deviations of model predictions from the data would occur if too few higher-order terms were included in such an approximation. If this deviation is too large to be acceptable, the response model is inadequate.

Obtaining a response model by fitting a sample of data to what seems to be a plausible candidate function is not very difficult, especially with widely available software packages that automate this function. The result is typically a “best fit” of the data to the specific candidate model, in a least-squares sense. This means it will be the fit that produces the smallest sum of squared residuals and thus the smallest residual or unexplained variance. Unfortunately, the best fit will not be a very good fit unless the candidate model adequately represents how the modeled response varies with changes in the independent variables.

[1] The term “surrogate” is often used to describe the entire fitted response model, which serves as a surrogate for the more complex underlying code. Here, however, surrogate (specifically, truth surrogate) refers to an unbiased estimate of the true response to which a prediction by the fitted response model is compared in order to assess the adequacy of that prediction.

Assessing the adequacy of such a model comprises the most intensive work in either a physical or a computational response surface modeling experiment. This task is complicated by the fact that accuracy requirements can be so stringent in modern aerospace testing that lack-of-fit bias errors small enough in absolute terms to elude cursory detection can still be unacceptable. We can ensure that a bias error of a given magnitude is detected with a prescribed level of confidence at some site in the design space by acquiring enough data that random errors cancel sufficiently for bias errors of that magnitude to emerge from the noise.

If a bias error is too large to be acceptable at a given site in the design space, the model is said to be out of tolerance at that site. If the model is out of tolerance at a large enough number of sites within the design space, the model can be said to be inadequate. The following subsections provide a specific framework for objectively inferring whether a candidate response model is out of tolerance at given sites in the design space selected for such an evaluation.

3.1. Hypothesis Testing

Formally designed experiments generally have at their core a test of two hypotheses to facilitate objective inferences about some important element of the experiment: a null hypothesis and a corresponding alternative hypothesis. Readers with little experience in professionally designed experiments may regard this as an unnecessary academic construction, but there are practical engineering advantages to this approach, not the least of which are insights into the crucial question of how much costly data has to be acquired for a successful result.

The null hypothesis in a response surface modeling experiment asserts that at a given site in the design space there is no significant difference between the response one would predict with the fitted model developed to represent the underlying code, and the true response at that site, which is typically taken to be what the underlying code predicts. That is, the null hypothesis asserts that the response model adequately represents the underlying code, so that any prediction errors are random and not systematic.

The alternative hypothesis asserts for the same site that a significant difference does exist between the true response and the response one would predict with the model developed in the experiment. By “significant” we mean “statistically significant,” which is to say too large to be attributed to ordinary random error. This implies that a systematic bias error is in play, which we usually attribute to lack-of-fit of in the response model. We also mean “significant” in the sense that the error is large enough to be detected unambiguously, and also to be of practical concern. Lack-of-fit errors that are too small to matter do not invalidate the response model.

In a well-designed experiment, the null and alternative hypotheses are mutually exclusive, so testing them consists of acquiring and analyzing the data necessary to reject one or the other. We focus on the null hypothesis by convention and for practical reasons: it is easier to detect when the null hypothesis is violated than when the alternative is violated. The decision either to reject the null hypothesis or not to reject it formally constitutes an inference. We commit an inference error if we reject the null hypothesis erroneously, or if we fail to reject it when we should have done so.

In a poorly designed experiment, it is possible to acquire so much data that both the null and the alternative hypotheses can be simultaneously rejected in a response surface modeling experiment if we neglect engineering significance. This occurs when the precision of the experiment is so great due to excessive data volume that bias errors too small to be of concern are unambiguously

detectable. It is not necessary to pay for enough data to resolve bias errors too small to be of concern, even if they truly exist.

It is likewise possible to acquire so little data that neither the null nor the alternative hypothesis can be rejected without assuming more inference error risk than is acceptable. This occurs when a bias error that is large enough to be of concern is still too small to be unambiguously resolved, given how much random error there is in the measurement environment. Increasing the volume of data would permit more random error to cancel in this case, providing a clearer view of any systematic bias error that may be in play.

The null and alternative hypotheses have been cast in terms of residuals, which are the differences between response model predictions and the best estimate of the true response at a given site in the design space. There are four possible inferences in evaluating this difference: (1) We can reject the null hypothesis when a residual truly is out of tolerance, (2) we can decide not to reject the null hypothesis when the residual is small enough, (3) we can reject the null hypothesis where the response model is actually adequate, or (4) we can fail to reject the null hypothesis where the response model is truly inadequate. Stated succinctly, we can either reject the null hypothesis or not, and in either case we can be right or wrong. In a well-designed experiment, the four combinations of these two possible decisions and two possible outcomes exhaust all possibilities. The first two cases are proper inferences, either correctly indicting the model at a given site, or correctly validating it there. The latter two are improper inferences, falsely indicting the model at a given site, or falsely validating it.

The improper inferences of cases 3 and 4 arise because of experimental error. They are called "alpha" and "beta" inference errors, respectively; also known as "Type I" and "Type II" inference errors. Theoretically, we can drive the probability of committing these errors to any arbitrarily low level (except zero) by acquiring enough data. To reach zero, we would require an infinite volume of data, although in most practical circumstances a finite and surprisingly small volume of data is sufficient to approach zero closely enough to satisfy even the most risk-averse experimentalist. Controlling the trade-off between inference error probability and data volume is the essence of scaling a designed experiment.

During the experiment design, we document the highest acceptable probabilities of committing Type I and Type II inference errors. These decisions are entirely subjective; they simply reflect the investigator's degree of risk aversion and there is no incorrect answer. A common industry specification for the largest acceptable Type I error probability in an aerospace experiment is 0.05, corresponding to "95% confidence" that a given response prediction is within a specified tolerance, or any other risk specification is equally valid.

There is no industry consensus for the largest acceptable Type II error probability. Maximum acceptable probabilities of Type I and Type II inference errors can be (and should be) different whenever the consequences of committing them are different. For example, the selection of $\alpha = 0.05$ and $\beta = 0.01$ would reflect a subjective assessment that validating an inadequate model (Type II error) is a more serious error than failing to recognize an adequate model (Type I error). The latter case might result in wasting additional resources while seeking an adequate model that was already in hand. The former would result in a model that produces the wrong answer. Neither result is desirable, but to the extent that one considers it a more serious offense, the investigator may wish to specify more inference error protection against Type II errors than against Type I errors.

3.2. Impact of Quality Requirements on Data Volume Needs

The first step in estimating data volume requirements in a response surface modeling experiment is to specify precision/accuracy requirements for the response model to be fitted. These requirements have associated with them a tolerance level and a probability.

It is useful to nail down the difference between precision and accuracy in response surface modeling. Precision can be visualized as a "fuzz band" within which the response model is centered. Consider a response model for drag coefficient that has a precision of ±0.5 drag count with 95% confidence. This is meant to convey the idea that there is a 95% probability that the true response lies between ±0.5 drag counts of the model's response prediction. The precision specification is a measure of aleatory uncertainty, due to chance variations (random error) in the fitted data. These chance variations cause uncertainty in the regression coefficients of the fitted model, which are propagated as prediction variance. So even if the model were itself a perfect representation of how drag depends on all of the independent variables of the experiment, there would still be some error in the model predictions that reflect the imperfections in each of the finite number of individual response estimates that were fitted to the model. The precision specification assumes no significant bias errors are in play. For a response model, this means that any lack-of-fit error that may exist is negligible compared to the random prediction variance.

An accuracy specification is expressed as the smallest systematic bias error that is too large to be acceptable. The accuracy of a candidate response model is assessed during a designed experiment and the model is rejected if there is not a sufficiently high probability that the accuracy specification will be met at randomly selected sites in the design space. There is a probability associated with an accuracy specification that is analogous to the confidence level of a precision specification. That probability is known as the resolving power of the experiment. It represents the probability that a systematic bias error with a specified magnitude can be detected, given the precision of the response model. These concepts will be considered in more detail below.

3.2.1. General Data Volume Consideration

Under the null hypothesis, each residual would be zero except for random experimental error. Random error results in a probability distribution for each residual, with a mean of zero and a level of variance that depends on the volume of data used to estimate the variance, among other things. Under the alternative hypothesis, there is also some distribution. The dispersion is the same as for the null hypothesis because it is caused by the same factors, but the dispersion would occur about a mean that is displaced from zero by the systematic lack-of-fit bias error, λ, that is assumed to be in play under the alternative hypothesis.

Figure 4 illustrates how the residuals are distributed under the null hypothesis, H_0, and the alternative hypothesis, H_A. The distributions are assumed to be normal by the Central Limit Theorem. They are separated by λ, the smallest unacceptable bias error. If lack-of-fit bias errors are much larger than λ, they are easy to detect and are therefore relatively uninteresting from a quality assessment point of view. Figure 4 illustrates the more interesting case in which the lack-of-fit bias error is just large enough to be unacceptable. This value of λ represents the accuracy specification for which we will scale the experiment. If we acquire enough data to detect this level of lack-of-fit without incurring unacceptable inference error risks, we will have acquired enough data to detect more blatant bias errors as well. We therefore limit our consideration of the alternative hypothesis to this threshold bias error, and each residual is assumed to be drawn from one or the other of the distributions in Fig. 4.

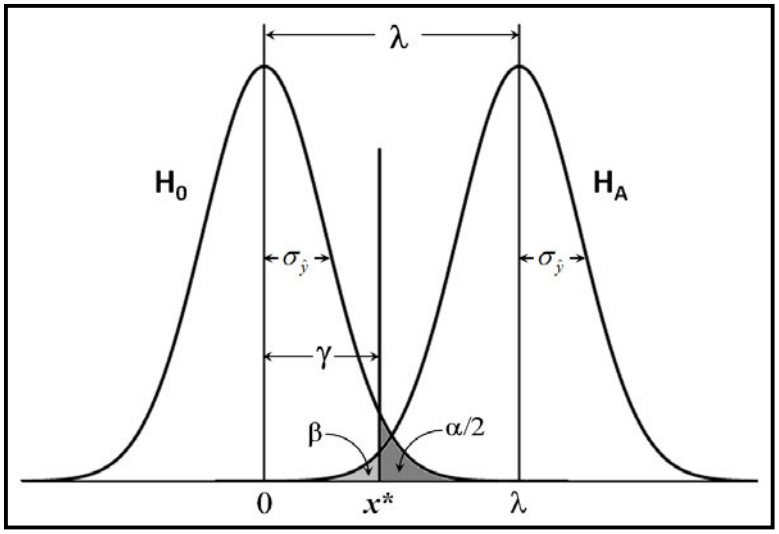


Figure 4. Distribution of response estimates $\hat{y}$ under the null and alternative hypotheses H_0 and H_A with tolerance, λ; acceptable inference error risks, α and β; and hypothesis rejection criterion, x^*.

Even though λ is the smallest bias error that is unacceptable, it is not sufficient for the residual to simply be less than λ for assessing model adequacy. The reason is that the magnitude a single residual drawn from the H_A distribution in Fig. 4 will be less than λ half of the time by definition of the mean of the distribution, notwithstanding the fact that the model is out of tolerance. We would therefore erroneously validate models with marginally unacceptable bias errors half of the time. Depending on the size of λ, random experimental error could even drive the residual under the H_0 distribution in Fig. 4 so that it is misinterpreted as evidence of an adequate fit, even if it is actually drawn from the H_A distribution. This would be a Type II inference error, as described earlier, which results in an erroneous model validation. There can also be occasions when random error causes the magnitude of a residual to be so large that it is misinterpreted as evidence of lack-of-fit, even when the overall model fits the data adequately. This is the Type I inference error introduced earlier.

We cope with the vagaries that random errors impose by defining a cut-off criterion to declare when a residual can be said to be in or out of tolerance. Such a criterion is labeled x^* in Fig. 4. If the magnitude of the residual is less than x^*, the residual is assumed to be drawn from the H_0 distribution on the left; otherwise it is assumed to be drawn from the H_A distribution on the right.

The shaded area under the H_0 distribution to the right of x^* represents the probability of committing a Type I inference error by erroneously rejecting the null hypothesis due to random experimental error. In such a case, random error would be responsible for erroneously indicting an adequate response prediction.

This shaded area is labeled $\alpha/2$ in Fig. 4 because, in this instance, the null hypothesis is two-sided. This means the null hypothesis asserts that the response model prediction is neither larger nor smaller than the truth surrogate. There is another symmetric area under the H_0 distribution to the left of negative x^* that has the same $\alpha/2$ area, but that area is not shown in the figure to reduce clutter. The total probability that a residual lies outside $\pm x^*$, given the null hypothesis, is therefore α.

The H_0 decision criterion, x^*, is a distance γ away from the mean of the distribution corresponding to the null hypothesis. This difference is a measure of precision in the experiment. The range of $\pm\gamma$ is the $(1-\alpha) \times 100\%$ confidence interval. Under the null hypothesis, we would expect the response model prediction to differ from the truth surrogate by no more than $\pm\gamma$ with a probability of $(1-\alpha) \times 100\%$.

The shaded area labeled β under the H_A distribution to the left of x^* represents the probability of erroneously rejecting the alternative hypothesis. This is a Type II inference error that validates a biased model prediction. It is the probability that an error actually drawn from the biased H_A distribution will be less than the x^* decision criterion simply due to chance variations in the data, and therefore is erroneously assumed to have been drawn from the H_0 distribution. The resolving power of the experiment is said to be $(1-\beta) \times 100\%$, representing the minimum probability that a bias error as small as λ will be detected.

The x^* criterion is a distance from the mean of each distribution in Fig. 4 that can be represented as a multiple of the standard deviation that characterizes each of the distributions. We say that x^* is z_α standard deviations to the right of the H_0 mean, and z_β standard deviations to the left of the H_A mean. The value of z_α depends on α. It is larger when α is smaller (corresponding to an x^* location farther to the right). Likewise, the value of z_β depends on β, and is larger when β is smaller (corresponding to an x^* location farther to the left).

When the H_0 and H_A distributions in Fig. 4 are normal, the quantities z_α and z_β are standard normal deviates that are tabulated in readily available statistical tables. They can be replaced by t-statistics when the sample variance is estimated with a small sample (typically not more than 10 points). Various commonly available software packages such as Excel can also be used to determine both the standard normal deviates and the t-statistics for specified values of α and β.

Estimates of the sample variance are typically based on samples large enough to use standard normal deviates instead of t-statistics, and we will assume that to be the case in this chapter. For example, $z_\alpha = 1.960$ for the two-sided null hypothesis with $\alpha = 0.05$, and $z_\beta = 2.327$ for the alternative hypothesis (always one-sided) with $\beta = 0.01$. From Fig. 4 we have:

$$\lambda = \left(z_\alpha + z_\beta\right)\sigma_{\hat{y}} \tag{7}$$

The standard deviation of the distributions of residuals in Fig. 4 depends on the volume of data used to estimate the response; the greater the volume of data, the narrower the distributions, and vice versa. The optimum sample size will be the one that establishes equality in Eq. 7 when λ is the smallest unacceptable bias error.

Depending on the circumstances, either constants or random variables can serve as truth surrogates to assess the adequacy of a response model at various sites in the design space. For example, in a physical experiment in which response model predictions are compared to "known" historical results (results for which there is high confidence), the truth surrogates are often treated as constants. This is also the case in certain certification experiments in which response model predictions are compared with some specific response goal. However, for the general case in which measured responses are used as truth surrogates, they must be treated as random variables for which their variance is explicitly taken into account. This is necessary in order for the

response model to be evaluated with respect to a best estimate of the true response. If variance in the truth surrogate is neglected, the response model can only be evaluated with respect to the specific validation points used for the surrogates.

The standard error in response estimates that appears in Eq. 7 and defines the widths of the distributions in Fig. 4 will have a different dependence on data volume depending on whether the truth surrogate is a constant or a random variable characterized by variance. For this reason, the minimum data volume requirements will be different in these two cases. We will address the case of constant truth surrogates and variable truth surrogates separately.

To ignore variance in the truth surrogate is to assume that each validation point represents the true response precisely, rather than as a mere estimate of the true response that is drawn from some distribution of such estimates. This was a rather common practice when response surface modeling was first introduced to wind tunnel testing in the 1990s. Skeptics charged with evaluating the method regarded direct measurements as "truth" and were inclined to regard response model predictions with suspicion. Any difference between measurement and model prediction was attributed to imperfections in the model. This meant that the de facto objective of a response surface modeling experiment was not necessarily to develop a model that reliably predicted the true response, but to develop a model that reliably reproduced the measured data. Variance in the truth surrogates used in wind tunnel testing has been recognized over time, but the effect this has on the scaling of designed experiments has not been widely understood. This distinction will be highlighted shortly.

Variance in the validation points of computational experiments is often neglected simply because replication fails to reveal it. A computational experiment is therefore much like a physical experiment in which the variance of validation points is ignored. In such an experiment, the most one can reasonably expect is to develop a response model that reproduces the underlying code within some specified tolerance without regard to the accuracy of the code itself. This is not necessarily "wrong," as long as it is understood that the regression model in such a case is meant to predict only the underlying code and not the true response, except in the unlikely event that the underlying code is error-free. However, if response variance is modeled in a computational experiment in the same way that response levels are (see Section 2), and if this variance is also modeled for the truth surrogates, then the fitted response model is more likely to represent the true underlying response if the variance estimates are realistic.

3.2.2. Truth Surrogate with No Variance

We now consider the minimum volume of data needed to ensure that inference errors are within specified limits when the truth surrogate is treated as a constant. The assumption of no variance in the truth surrogate is not very realistic in most experiments, but it is nonetheless of interest for two reasons. The first is that this assumption is often implicitly made by those inclined to attribute non-zero residuals entirely to imperfections in the response surface model. The second is that in computational experiments, the truth surrogate is generally a response prediction produced from the underlying code, for which variance at a given site in the design space is not directly observed by replication.

Regardless of the nature of the truth surrogate used as a reference in quantifying residuals, the RSM scaling problem is complicated by the fact that the standard deviation of the residual probability distributions in Fig. 4 is site-dependent. That is to say, the regression model prediction variance depends on the specific combination of independent variables for which the variance is to be evaluated, which will not be determined until the number of points to be fitted to the model

is determined. However, the average prediction variance across all sites used to fit a polynomial linear regression model can be computed in advance, by a simple formula derived in Section 5. It is presented here for convenience as Eq. 9, and is a function of the number of terms in the response model, p, as well as the number of fitted points, N.

A full d^{th}-order linear regression model in k independent variables has p terms, including the intercept, where

$$p = \frac{(d+k)!}{d!k!} \tag{8}$$

If N data points have been fitted to such a model, the standard error associated with the average prediction variance over all fitted sites is shown in Section 5 of this chapter to be

$$\bar{\sigma}_{\hat{y}} = \sigma_0 \sqrt{\frac{p}{N}} \tag{9}$$

where σ_0 is the ordinary standard deviation of the data. Since a minimum of $N = p$ data points is required to fit a polynomial with p terms, we see from Eq. 9 that the standard prediction error of a fitted response model is never more than the standard deviation associated with unreplicated direct response estimates. For the common case in which N exceeds p, the average uncertainty associated with a regression model prediction is less than the uncertainty associated with a direct response estimate at the same site in the design space. This is because of the hidden replication illustrated in Fig. 3 and discussed in the surrounding text.

Insert Eq. 9 into Eq. 7 and solve for N:

$$N = p\left[\frac{(z_\alpha + z_\beta)\sigma_0}{\lambda}\right]^2 \tag{10}$$

Equation 10 indicates that in a response surface modeling experiment, the minimum volume of data required to generate a math model capable of adequately predicting system responses is directly proportional to p, the number of terms in the model (given in Eq. 8 for a d^{th}-order polynomial response model in k factors), and the variance of the measurement environment (square of the standard deviation, σ_0). It also depends on the inference error risks, α and β, that one is willing to assume in validating the model, and it is inversely proportional to the tolerance for bias error, λ.

It is not uncommon for the investigator to have only a vague initial idea of what value to specify for the tolerance, λ; there is often only a qualitative perception that the response model should fit the data well, with no particular quantitative tolerance value in mind. The standard deviation, σ_0, likewise can be imperfectly understood at the start of an experiment. In such cases, one might exploit the fact that it is not necessary to specify either λ or σ_0 individually in order to quantify minimum data volume requirements by Eq. 10; one need only specify their ratio. That is, for scaling purposes, one need only express the tolerance as some multiple of the standard deviation.

For example, one might specify the 95% Least Significant Difference (95% LSD) as a tolerance. This is the smallest difference between two response estimates that can just be resolved with 95%

confidence, given that both are random variables with a standard deviation of σ_0. Absent any requirement or preference for a different tolerance, the 95% LSD is not an unreasonable specification for experiment design purposes. It can be quantified as follows:

$$95\%\ \text{LSD} = 2\sqrt{2}\sigma_0 \tag{11}$$

For λ = 95% LSD, we can insert Eq. 11 into Eq. 10:

$$\lambda = 95\%\ \text{LSD} = 2\sqrt{2}\sigma_0 \rightarrow N = \frac{1}{2}p\left(\frac{z_\alpha + z_\beta}{2}\right)^2 \tag{12}$$

For $\alpha = 0.05$ and $\beta = 0.01$, so $z_\alpha = 1.960$ and $z_\beta = 2.327$, we have from Eq. 12 that $N = 2.30p$. Let us assume that we are fitting some fourth-order polynomial response model as a function of three independent variables; say angle of attack, angle of sideslip, and Mach number. By Eq. 8, we have for $d = 4$ and $k = 3$ the number of terms, $p = 35$. Therefore, $N = 2.30p = 81$ for $\alpha = 0.05$ and $\beta = 0.01$, after rounding N up to the next highest integer.

Let us assume that this is a response model for drag coefficient, and the standard deviation in direct estimates of drag coefficient, σ_0, is 0.00005, or a half drag count. The average standard prediction error by Eq. 9 is $\sqrt{\frac{35}{81}}\sigma_0$ or $0.657\sigma_0$, in this case, so hidden replication has reduced the prediction uncertainty to less than 2/3 of the uncertainty of a direct response estimate. The criterion for rejecting the null hypothesis, x^*, is $z_\alpha\sigma = 1.960 \times 0.657 \times 0.5$ drag counts = 0.64 counts. Therefore, in this example, if a given residual is greater than 0.64 counts, we would reject the null hypothesis and infer with no more than a 5% chance of error that the response model does not predict within 0.64 counts. If the residual is less than 0.64 counts, we would not reject the null hypothesis, and would infer with no more than a 1% chance of error that any bias error at that site in the design space is not greater than the 95% LSD.

We can gain some additional insight into the relationship between data volume, residual tolerance, and inference error risk by considering the special case in which the response model is fitted with the smallest number of points possible. A polynomial response model with p terms can be fitted with as few as $N = p$ points, but no fewer. In such a case, there are no extra (residual) degrees of freedom to permit random variations to cancel about the best fit, and by Eq. 9 the average model prediction variance is no smaller than the sample variance of the raw data. The probability distributions in Fig. 4 then have a standard deviation of σ_0. Let us designate λ as λ_0 for this special case of no residual degrees of freedom. Since the bracketed term in Eq. 10 has a value of 1 when $N = p$, we have

$$\lambda_0 = \left(z_\alpha + z_\beta\right)\sigma_0 \tag{13}$$

The quantity λ_0 represents the smallest bias error that can be detected without residual degrees of freedom when the measurement environment is characterized by a standard deviation of σ_0, given that α and β are the greatest acceptable risks of erroneously declaring a bias error of magnitude λ, or erroneously declaring a bias error to be less than λ, respectively. For the example just considered in which σ_0 is a half drag count and α and β are 0.05 and 0.01 respectively, we use Eq. 13 to understand that $\lambda_0 = 2.1$ drag counts. Without residual degrees of freedom, systematic bias

errors must have a magnitude as large as 2.1 drag counts to be detected with no more than a 5% chance of erroneously declaring an inadequate model fit (Type I inference error), or a 1% chance of erroneously validating the model (Type II inference error).

Equation 10 can then be expressed as:

$$N = p\left(\frac{\lambda_0}{\lambda}\right)^2 \tag{14}$$

Equation 14 shows that for prescribed levels of inference error risk α and β, corresponding via Eq. 13 to a specified λ_0, tighter tolerances ($\lambda < \lambda_o$) can only be achieved by increasing N. This means adding residual degrees of freedom contributes to quality assurance, which is in harmony with intuitive expectations.

We can also rearrange Eq. 14 to show how tolerance, λ, depends on N:

$$\lambda = \lambda_0\sqrt{\frac{p}{N}} \tag{15}$$

Acquiring more and more data improves the precision of response estimates via Eq. 9. The result of this increase in precision is to permit smaller and smaller bias errors to be detected within specified inference error risk limits, as shown by Eqns. 14 and 15. There will be a volume of data for which a bias error specified as the smallest of concern, λ, can be detected without exceeding prescribed limits of acceptable inference error risk, α and β. It is possible to detect even smaller bias errors—bias errors too small to be of concern—by expending the resources necessary to acquire enough additional data to do so, but such an unnecessary expense would be difficult to justify.

3.2.3. Truth Surrogate with Variance

The scaling calculations have not as yet taken into account any variance in the truth surrogate used as a reference in constructing the residuals. Variance in the distributions of Fig. 4 was associated only with uncertainty in the fitted response model. We now assume that the truth surrogate in either a physical or a computational experiment is characterized by variance, without regard to how it is estimated. We will also assume that this variance in the truth surrogate is the same as in the sample of data used to fit the response model, since both the truth surrogate and the fitted data sample will have been drawn from the same population of response estimates.

It can be shown that under commonly occurring circumstances, the variance of the difference in two random variables is the sum of the variances of each of them. Therefore, the variance in a sample of residuals consists of the sum of the response model's prediction variance as given in Eq. 9 and the variance of the truth surrogate. Assume that the truth surrogate has a variance σ_0^2. Then the distribution of residuals has a variance as follows:

$$\sigma_r^2 = \sigma_0^2 + \frac{p}{N}\sigma_0^2 \rightarrow \sigma_r = \sigma_0\sqrt{1+\frac{p}{N}} \tag{16}$$

Recall that Eq. 7 is the basic relationship between residual tolerance, λ; the acceptable risks, α and β, of erroneously rejecting the null and alternative hypotheses, respectively; and the standard error in residuals when the truth surrogate is a constant. In such a case the variance in the residuals is just the response model prediction variance, $\sigma_{\hat{y}}$ but this notation is generalized in Eq. 16 for the case in which the truth surrogate is a random variable with non-zero variance. Insert Eq. 16 into Eq. 7, respecting this change in notation:

$$\lambda = \left(z_\alpha + z_\beta\right)\sigma_0\sqrt{1+\frac{p}{N}} \tag{17}$$

Square both sides of Eq. 17:

$$\lambda^2 = \left(z_\alpha + z_\beta\right)^2\left(\frac{N+p}{N}\right)\sigma_0^2 \tag{18}$$

Multiply both sides by N:

$$N\lambda^2 = N\left(z_\alpha + z_\beta\right)^2\sigma_0^2 + p\left(z_\alpha + z_\beta\right)^2\sigma_0^2 \tag{19}$$

Gather terms in N:

$$N\left[\lambda^2 - \left(z_\alpha + z_\beta\right)^2\sigma_0^2\right] = p\left(z_\alpha + z_\beta\right)^2\sigma_0^2 \tag{20}$$

Solve for N:

$$N = p\left[\frac{\left(z_\alpha + z_\beta\right)^2\sigma_0^2}{\lambda^2 - \left(z_\alpha + z_\beta\right)^2\sigma_0^2}\right] \tag{21}$$

We can combine this with Eq. 13 as in the case of no variance in the truth surrogate, to generate a compact representation of the data volume requirement in terms of λ_0, for the case in which the truth surrogate displays variance:

$$N = p\left[\frac{\lambda_0^2}{\lambda^2 - \lambda_0^2}\right] \tag{22}$$

Compare Eq. 22 with Eq. 14 to see how data volume requirements change when there is variance in the truth surrogate. Rearrange to compute the magnitude of the bias error that can be detected within inference error risk specifications and compare with Eq. 15 to also see how bias error resolution changes when there is variance in the truth surrogate.

$$\lambda = \lambda_0\sqrt{1+\frac{p}{N}} \tag{23}$$

3.2.4. Effect of Variance in the Truth Surrogate

To further compare the effect of a constant truth surrogate to the effect of a truth surrogate with variance, it is instructive to consider for both cases the limits in which *N* is a minimum and in which *N* increases without bound. Consider first the case in which *N* is a minimum. There must be at least one fitted point for every term in a *p*-term polynomial response model, so $N = p$ when *N* is a minimum. (All *N* sites must also be unique. In general, in order to fit a *p*-term model, the data must be drawn from at least *p* unique sites in the design space). When $N = p$, all available degrees of freedom are consumed in estimating the response model coefficients, with no extra degrees of freedom available to quantify variance at a given site in the design space.

When $N = p$, the terms in square brackets in Eq. 21 and Eq. 10 must each have a value of 1. For the case in which the truth surrogate of a residual has no variance, which corresponds to Eq. 10, we have:

$$\text{Constant Truth Surrogate, } N = p \rightarrow \frac{\left(z_\alpha + z_\beta\right)\sigma_0}{\lambda} = 1 \rightarrow \left(\frac{\lambda}{\sigma_0}\right)_{N=p} = z_\alpha + z_\beta \tag{24}$$

Or, by Eq. 13,

$$\text{Constant Truth Surrogate, } N = p \rightarrow \lambda_{N=p} = \lambda_0 \tag{25}$$

Equation 21 describes the case in which the truth surrogate displays variance. For $N = p$ we have

$$\frac{\left(z_\alpha + z_\beta\right)^2 \sigma_0^2}{\lambda^2 - \left(z_\alpha + z_\beta\right)^2 \sigma_0^2} = 1 \tag{26}$$

and therefore

$$\left(z_\alpha + z_\beta\right)^2 \sigma_0^2 = \lambda^2 - \left(z_\alpha + z_\beta\right)^2 \sigma_0^2 \rightarrow 2\left(z_\alpha + z_\beta\right)^2 \sigma_0^2 = \lambda^2 \tag{27}$$

or

$$\text{Variable Truth Surrogate, } N = p \rightarrow \left(\frac{\lambda}{\sigma_0}\right)_{N=p} = \sqrt{2}\left(z_\alpha + z_\beta\right) \tag{28}$$

That is to say,

$$\text{Variable Truth Surrogate, } N = p \rightarrow \lambda_{N=p} = \sqrt{2}\lambda_0 \tag{29}$$

The response model is forced through every fitted data point when $N = p$, and by Eq. 9, the prediction standard error is just the ordinary standard error of the measurement environment. For constant and variable truth surrogates, λ/σ_0 in Eq. 24 and Eq. 28 represents the smallest response difference that can be resolved without Type I and Type II inference error probabilities exceeding

α and β, respectively, when $N = p$. This minimum difference is expressed as a multiple of σ_0, the irreducible standard error of the measurement environment.

Consider, for example, an experiment in which 95% confidence is required to infer that a residual is out of tolerance, and 99% confidence is required to infer that it is within tolerance. These confidence levels are selected by the investigator to reflect his inference error risk tolerance and can be larger or smaller, but for this example $\alpha = 0.05$ and $\beta = 0.01$, for which $z_\alpha = 1.960$ and $z_\beta = 2.327$. In this case, the smallest residual that can be resolved when $N = p$ is $4.287 \times \sigma_0$ if the truth surrogate is a constant, and $6.063 \times \sigma_0$ if the truth surrogate is a random variable.

In a high precision wind tunnel test the standard deviation corresponding to irreducible drag measurement error might be 0.25 drag counts. Assume that the response model has been fitted with the smallest possible sample of data. Residuals would then have to be at least 1.1 counts to be unambiguously detected if the truth surrogate is a constant. If the truth surrogate is a random variable, residuals would have to be larger to be detected with the same inference error risk—at least 1.5 counts.

Having considered the minimum-volume situation in which $N = p$, consider now the opposite extreme in which the volume of data increases without bound. Equation 10 describes the case in which the truth surrogate is constant. This can be rearranged as follows:

$$\frac{\lambda}{\sigma_0} = \sqrt{\frac{p}{N}}\left(z_\alpha + z_\beta\right) \tag{30}$$

Therefore,

$$\text{Constant Truth Surrogate, } N = \infty \rightarrow \lambda_{N=\infty} = 0 \tag{31}$$

Absent any contribution from variance in the truth surrogate, all of the uncertainty in the distribution of residuals is attributable to variance in the response model. Since the model prediction variance approaches zero as N increases without bound using Eq. 9, and if there is no variance in the truth surrogate, then Eq. 31 indicates that any lack-of-fit error, λ, no matter how small, can be detected if a sufficient volume of data is acquired. Note in Eq. 30 that this is true even for arbitrarily small inference error probabilities, α and β.

Consider a case in which the truth surrogate displays variance and the sample of data used to fit the response model is large. Equation 21 indicates that N will increase without bound as the denominator in the bracketed term approaches zero, as follows:

$$\text{Variable Truth Surrogate, } N = \infty \rightarrow \lambda^2 - \left(z_\alpha + z_\beta\right)^2 \sigma_0^2 = 0 \rightarrow \left(\frac{\lambda}{\sigma_0}\right)_{N=\infty} = z_\alpha + z_\beta \tag{32}$$

or, with Eq. 13:

$$\text{Variable Truth Surrogate, } N = \infty \rightarrow \lambda_{N=\infty} = \lambda_0 \tag{33}$$

Comparing Eq. 32 with Eq. 30 reveals that λ/σ_0 approaches $z_\alpha + z_\beta$ for two extreme cases: (1) when the truth surrogate is variable and N is unbounded; and (2) when the truth surrogate is constant and $N = p$, the minimum sample size needed to fit a polynomial with p terms. This is because with Eq. 9 and Eq. 16, the distribution of residuals has the same standard deviation in each case, namely, σ_0.

Formulas for average model prediction standard errors given in Eq. 9 and Eq. 16 for constant and variable truth surrogates are displayed in Table 1. They are evaluated for the special cases of minimum and maximum fitted data volume.

Table 1. Average standard error in distribution of residuals, σ_r, for constant and variable truth surrogates, given minimum and maximum samples of fitted data.

σ_r	Truth Surrogate is a CONSTANT	Truth Surrogate Exhibits VARIANCE
$N = p$	$\sigma_0\sqrt{\frac{p}{N}} = \sigma_0$	$\sigma_0\sqrt{1+\frac{p}{N}} = \sqrt{2}\sigma_0$
$N = \infty$	$\sigma_0\sqrt{\frac{p}{N}} = 0$	$\sigma_0\sqrt{1+\frac{p}{N}} = \sigma_0$

Table 2 summarizes results derived in Eqns. 25, 29, 31, and 33. It displays the smallest systematic bias errors that can be resolved with inference error probabilities, α and β, for the four cases in Table 1.

Table 2. Smallest systematic bias errors resolvable with specified inference error risk for constant and variable truth surrogates, given minimum and maximum data volume. $\lambda_0 = (z_\alpha + z_\beta) \times \sigma_0$.

λ	Truth Surrogate is a CONSTANT	Truth Surrogate Exhibits VARIANCE
$N = p$	λ_0	$\sqrt{2}\lambda_0$
$N = \infty$	0	λ_0

Table 2 reveals that increasing the data volume has a significantly greater effect if the truth surrogate is constant, than if the truth surrogate is variable. A constant truth surrogate enables one to resolve arbitrarily small residuals with a specified inference error risk, by acquiring a sufficiently large volume of data. This is because the average residual variance is inversely proportional to N. New data points enable one to detect smaller and smaller imperfections in the model, approaching zero asymptotically. In practice, there is always some tolerance level that represents a prediction error small enough to be acceptable, so there is no need to incur the

expense or to take the time to acquire so much additional data that even smaller errors can be detected.

Table 2 also shows that if the truth surrogate displays variance, there is a physical limit to the benefits of acquiring more data, not just a practical one. The residual variance has as components the response model prediction variance and the variance in the truth surrogate. Even if one were to drive the prediction variance arbitrarily close to zero by fitting enough data to do so, the finite variance in the truth surrogate would ensure a limit on how small a model imperfection could be resolved with a given level of confidence. Once the smallest volume of data necessary to fit a polynomial response model has been acquired (one point for each term in the model), the resolution of the experiment can only be further increased by a factor of $\sqrt{2}$ or about 40%, and this only in the limit of very large N. This has practical productivity implications because it suggests that the value of additional data diminishes as the resolution limit is approached.

3.3. Scaling Procedure in an Example Response Surface Modeling Experiment

The following example illustrates typical resource savings when consideration is given in advance to the volume of data required in a response surface modeling experiment. We considered an example earlier in which we used Eq. 10 to estimate the volume of data required to resolve a systematic error in drag coefficient the size of the 95% LSD, with an inference error probability no greater than $\alpha = 0.05$ of indicting an adequate model and an inference error probability no greater than $\beta = 0.01$ of validating an inadequate model. We assumed in that example that the truth surrogate was constant so all variance in the distribution of residuals could be attributed to error in the response model estimates. We found that in order to resolve a systematic error as small as the 95% LSD without incurring more than the specified inference error risks, we would need to fit $N = 2.30p$ points, where p is the number of parameters in the polynomial response model, including the intercept. For the fourth-order polynomial in three factors considered in that example, $p = 35$ and thus $N = 81$, rounded to the next highest integer.

In this prior example, we assumed that the standard deviation in direct estimates of drag coefficient, σ_0, was 0.00005, or a half drag count. Using Eq. 9, the average standard prediction error would have been $\sqrt{\frac{35}{81}}\sigma_0 = 0.657\sigma_0$, so hidden replication in the regression model would have reduced the prediction uncertainty to about 2/3 of the uncertainty of a response estimated by an individual direct measurement.

The criterion for rejecting the null hypothesis, x^*, is $z_\alpha\sigma$, which, for this example, would have been $1.960 \times 0.657 \times 0.5 = 0.64$ drag counts. Therefore, if a given residual had been greater than 0.64 counts, we would have rejected the null hypothesis. If the residual had been less than 0.64 counts we would not have rejected the null hypothesis, and would have inferred with no more than a 1% chance of error that it was due either to random error, or to a systematic modeling error that was within tolerance—a magnitude no greater than the 95% LSD. We would therefore have validated the model at that site.

Consider now a variation of the earlier example, identical in all respects except that instead of assuming the truth surrogate was a constant, we now assume that it displays variance with a standard error of σ_0. In this variation, we scale the experiment using Eq. 22 instead of Eq. 10, recalling from Eq. 13 that $\lambda_0 = (z_\alpha + z_\beta) \times \sigma_0$. Since $\alpha = 0.05$ and $\beta = 0.01$ (so $z_\alpha = 1.960$ and $z_\beta = 2.327$), and $\sigma_0 = 0.00005$, or half a drag count as before, we have the same result of $\lambda_0 = 0.00021$, or 2.1 drag counts. However, we have a problem if we wish to use the same

λ = 95% LSD for our systematic error tolerance because the 95% LSD is 1.414 counts using Eq. 11, which is smaller than λ_0, and with Eq. 22 this would lead to a non-physical result, namely, a negative sample size requirement. Table 2 indicates that if single-point response measurements are used as truth surrogates in a measurement environment with a standard error of σ_0, the smallest bias error that can be resolved with inference error probabilities no greater than α and β is λ_0, no matter what size sample is used to fit the response model.

We can use Eq. 23 to solve this problem in reverse, by calculating the smallest bias error than can be resolved with the specified inference error probabilities if we acquired the same volume of data as in the case with no variance in the truth surrogate; in this case, N = 81 data points. Inserting λ_0 = 2.1 counts, p = 35, and N = 81 into Eq. 23, we have:

$$\lambda = \lambda_0\sqrt{1+\frac{p}{N}} = \lambda_0 \times \sqrt{1+\frac{35}{81}} = 2.1\times 1.2 = 2.5 \text{ counts} \tag{34}$$

So for a measurement environment with the same standard error, σ_0, and the same maximum acceptable inference error probabilities, α and β, the smallest systematic bias error, λ, that can be resolved with the same volume of fitted data is about 20% larger when the truth surrogate is a random variable than when it is a constant. This assumes that the truth surrogate is a single-point measurement as is the usual case. The reduced resolution is attributed to the fact that there is uncertainty in the truth surrogate as well as in the response model estimate. In this example, we can resolve systematic modeling errors smaller than 2.5 counts with the same inference error risks as before by fitting more than N = 81 points, but with Eq. 22, we cannot resolve inference errors smaller than λ_0 = 2.1 counts, no matter how many data points we fit to the same response model.

We revisit Eq. 16, which represents the formula for variance in validation-point residuals when the truth surrogate is not a constant, having a standard error of σ_0. Mild conditions are assumed to be met that permit this to be represented as the sum of the response model prediction variance and the variance in the truth surrogate.

It is not uncommon for the truth surrogate at a given site to consist of a single, unreplicated response estimate. In a computational experiment, the estimate is computed with the underlying code and is not replicated for obvious reasons. In a physical experiment, it is typically an individual validation point measurement that is not usually replicated due to resource constraints. That is to say, the validation point can be considered the mean of an m-point sample, for which m = 1. The variance in the mean of that sample is then just the sample variance of the measurement environment, with a standard error denoted in Eq. 16 as σ_0.

Because the variance in validation residuals is the sum of the prediction variance and the truth-surrogate variance, the standard error of the residuals can never go below a lower limit dictated by the truth surrogate variance, no matter how small the response model prediction variance is driven by fitting a large volume of data. This fact, reflected in Table 1, has the practical effect of limiting the resolution of such an experiment, since there is a lower limit on the modeling error that can be resolved with no more than specified levels inference error risk. Table 2 displays these limits.

4. Site Selection

Having determined how many data points to acquire, one must decide which points to select in order to determine how the data sample is to be distributed within the design space. There are several candidate criteria, not all of which can be satisfied simultaneously. There are some general considerations that we will review briefly, but there is no universal agreement as to which criteria are most appropriate. There can be considerable overlap among the various criteria, and often optimizing for one criterion improves the design with respect to other criteria. The specific criterion to optimize is, to a great extent, a matter of the individual experimenter's particular sensibilities.

One criterion is to optimize coverage of the design space by ensuring that various subspace regions each have a minimum number of points assigned to them. The idea here is to ensure that there are no significant subspaces within which insufficient data are available to adequately model the responses. Such a criterion seeks to minimize lack-of-fit bias errors in the response model. One-factor-at-a-time (OFAT) testing implicitly uses a space-filling strategy, at least over those subsections of the design space where resource constraints, coupled with site prioritization, dictate that data be acquired. Response models are not typically fitted to the data in OFAT testing, but denser coverage of subspaces of interest is generally considered more desirable than coverage that is less dense.

Another general criterion is to optimize various aspects of the distribution of standard errors in the response predictions. Recall that the standard error in the response prediction is the one-sigma value for the distribution of prediction errors that are due to random errors in the fitted data. The prediction variance (square of the standard prediction error) averaged over all fitted points is the same for a given sample size and term-count in the model, no matter the distribution of sites where fitted data are acquired within the design space. However, the specific prediction variance at a given site is site-dependent, and by adjusting the site distribution it is possible to impart certain properties to the distribution of standard prediction errors while maintaining a constant average prediction variance. That is to say, while the total information that can be extracted by fitting a given model to a data sample of a given size is the same for all site distributions, that information can be distributed throughout the design space in different ways depending on which specific points are fitted to the model. Figure 5 illustrates the basic idea.

Figure 5 compares the distribution of prediction errors for two 12-point samples. In both cases, the points have been fitted to a quadratic response model in two factors. The center point of the design on the left is replicated four times. No other sites are replicated in either the design on the left or the design on the right, so they each feature the same 12-point sample size. The optimized site distribution on the left is known as a Box-Wilson or Central Composite Design (CCD). It has many desirable properties that make it the design of choice in process and product improvement experiments, among many other applications. Such experiments are performed in industrial engineering applications where second-order models are typically used to find factor combinations that maximize some process feature, such as throughput, or that minimize something like inspection failures.

Unfortunately, the CCD is of limited utility in applications that require higher-order response models, as is the case in many aerospace applications. For example, even a third-order model in two factors has ten terms in the full model, which exceeds the nine unique sites of the two-factor CCD. It would therefore be impossible to fit such a model. Likewise, a CCD in three factors features only 15 unique design-space sites, while a third-order polynomial in three factors has 20 terms and requires at least that many unique fitting sites.

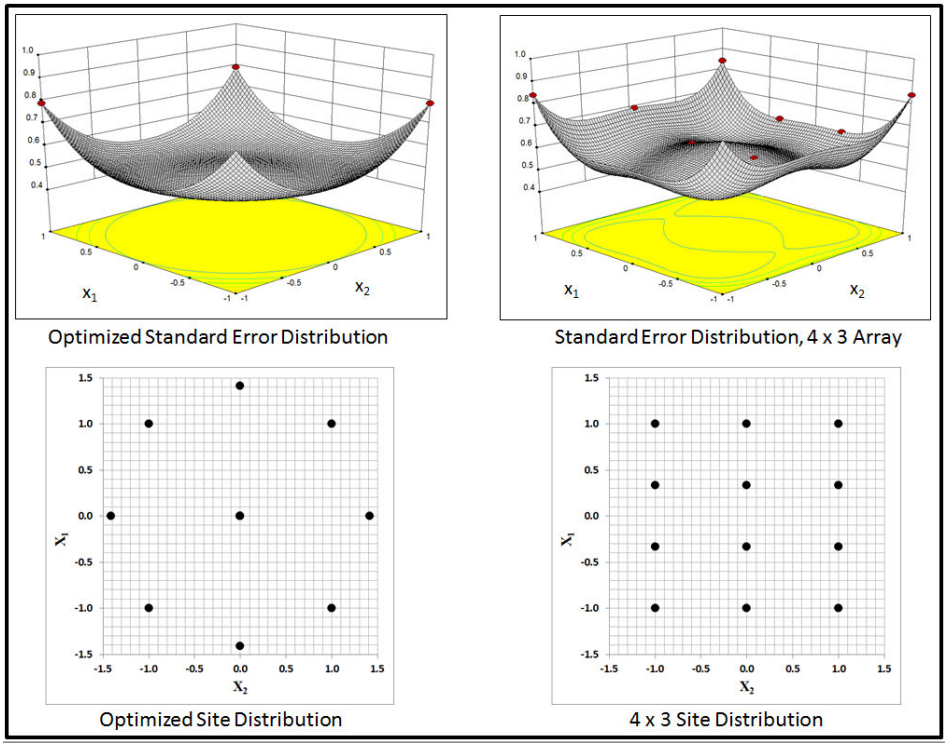


Figure 5. Two 12-point site distributions with corresponding distributions in standard error of prediction for a quadratic model in two factors. Center point of optimized distribution is replicated four times.

The CCD design on the left can be regarded as an optimized variation of the ordinary rectilinear design on the right, which might be specified if prediction error variance is not taken into account. Note that the distribution of prediction errors can be altered fairly substantially by changing the locations within the design space where data are to be acquired. In this case, the design on the left is an improvement on the one on the right in a number of regards. First, while the average prediction error is the same in both cases, the optimized design has lower prediction errors over a relatively broad region about the center of the design space. This is achieved by pushing more of the uncertainty into the corners. The distribution of prediction errors is also symmetric with respect to both direction and distance from the center of the design, which can be desirable in some applications. A design featuring such azimuthal symmetry in the prediction variance is said to be "rotatable."

It is possible to optimize other properties of the prediction error variance. For example, a site distribution that minimizes the uncertainty in response model coefficients is displayed in Fig. 6 for a sample of the same size (12 points), fitted to the same quadratic model in two factors. More precisely, this design—known as a D-optimal design—minimizes the determinant of the covariance matrix, which will be discussed in Section 5. This has the effect of minimizing the volume of the joint confidence ellipsoid for the model coefficients, which is a kind of multi-dimensional confidence interval. This criterion, which is satisfied at the expense of lost azimuthal and radial symmetry in the distribution of prediction standard errors, may be desirable in studies focused on the physics of the underlying process, in which the relative importance of individual model terms is of greater interest than the overall model prediction at a given site.

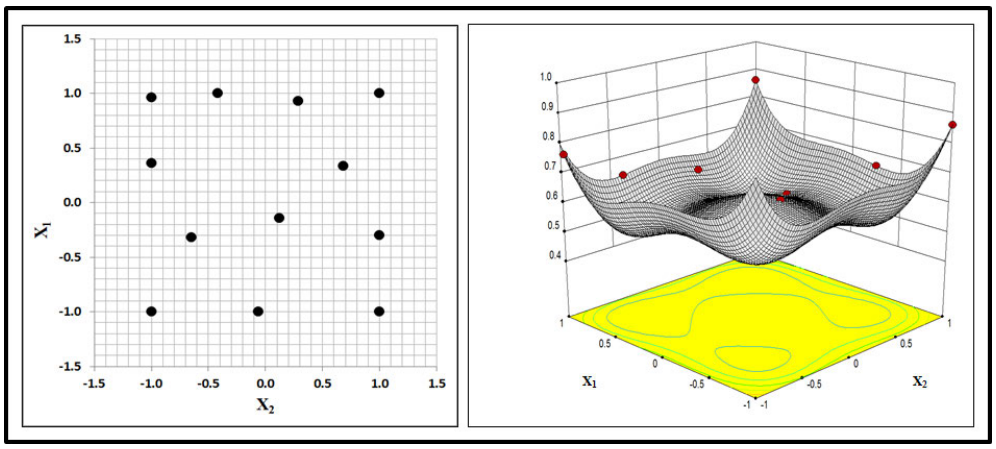


Figure 6. A 12-point D-optimal site distribution with corresponding distribution in standard error of prediction for a quadratic model in two factors. Sites selected to minimize a joint confidence ellipsoid in model coefficients.

Figure 7 displays a distribution of 12 sites that minimizes the integral of the prediction variance over the entire design space. This is known as an I-optimal design, which is desirable in applications for which the goal is to achieve the greatest overall precision in response estimates throughout the design space. Such a design is desirable in many aerospace response surface modeling applications.

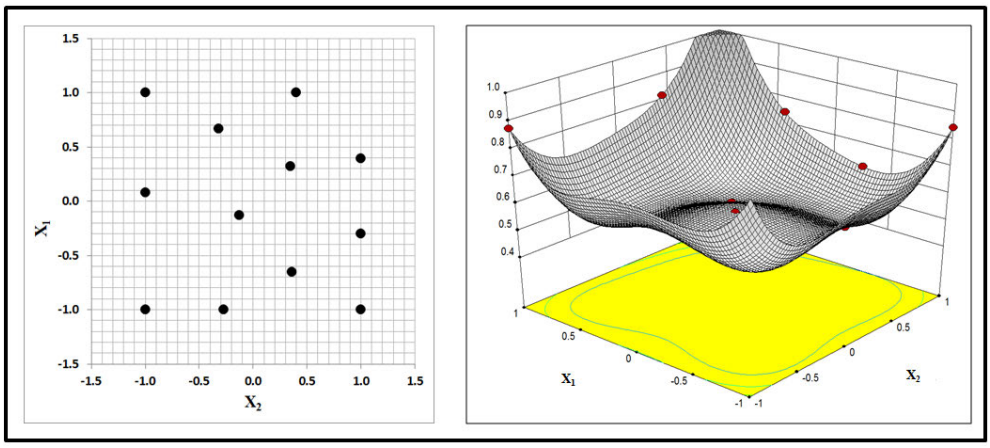


Figure 7. A 12-point I-optimal site distribution with corresponding distribution in standard error of prediction for a quadratic model in two factors. Sites selected to minimize the integral of the prediction variance over the design space.

Figure 8 displays a 12-site distribution that minimizes the trace of the covariance matrix, to be described in Section 5. As in the previous figures, the distribution of standard errors is for a quadratic response model in two factors. A design optimized by minimizing the trace of the covariance matrix is called A-optimal, and has the effect of minimizing the average uncertainty of all the model coefficients.

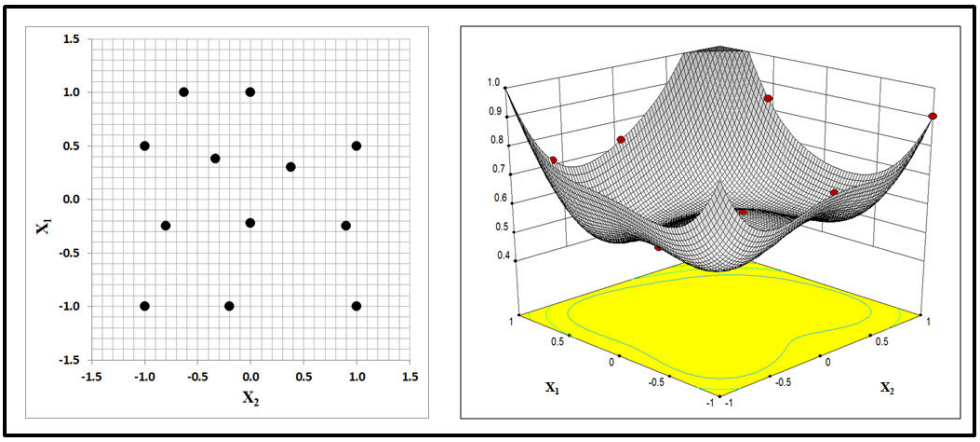


Figure 8. A 12-point A-optimal site distribution with corresponding distribution in standard error of prediction for a quadratic model in two factors. Sites selected to minimize the average variance of the response model coefficients.

There are other optimization criteria among what are known as the alphabetic optimization methods. For example, a G-optimal design distributes sites in the design space in such a way as to minimize the largest prediction variance at any given site. All of these optimization methods are algorithmic, requiring computer software designed for the purpose of generating such designs. The process entails the generation by computer of a large number of candidate site distributions that are rank-ordered according to the relevant covariance matrix metric.

The optimized site distributions in Figs. 6-8 do not differ from each other dramatically. There is considerable correlation among the various criteria; site distributions with low coefficient uncertainties tend to be similar to those that minimize prediction variance, etc. Regardless of the specific optimality criterion, improvements in the distribution of prediction errors are achieved essentially for free, in that they do not require any additional data.

5. Response Model Formation

At this point we have specified precision/accuracy requirements and maximum acceptable probabilities for associated inference error risks. We have scaled the design by determining the minimum volume of data required to satisfy these quality requirements, and have distributed the points from a sample of this size within the design space in order to optimize the distribution of standard prediction errors according to some criterion that we have chosen. We have acquired data at this minimum number of optimally distributed fitting sites, and are now ready to generate response models capable of adequately predicting system responses of interest (force, moments, pressures, etc.) as a function of the factors that have been varied in the experiment. Material in this section draws heavily from work by DeLoach and Ulbrich [5].

Response model building proceeds in stages that will be reviewed in this section. We begin by coding the independent variables to scale and center them, a step that has advantages outlined in the first subsection. While other modeling methods are possible, linear regression for polynomial response models is common in aerospace experiments, and is reviewed in the second subsection. Not all terms in the regression model will be significant; this means the coefficients for some of the terms will not be large compared to the uncertainty in estimating them. Dropping such terms in a process known as model reduction can improve the precision of response model predictions,

as will be described in the third subsection. Model reduction is subject to the need to maintain hierarchy in the response model, as will be explained next. It may be possible to further reduce prediction variance by transforming the response variables and/or the independent variables. Tests to identify such transformations are summarized in the final two subsections.

5.1. Coding of Independent Variables

Coding is a process that invokes a linear transformation to map physical units into a dimensionless scale, typically from –1 to +1. Let ξ represent some independent variable such as angle of attack or Mach number, and let L and H represent the lowest and highest values of this variable in physical units. The following transformation converts a physical variable ranging from L to H into a coded variable ranging from –1 to +1. This transformation both scales and centers the variable:

$$x = \frac{\xi - \tfrac{1}{2}(H + L)}{\tfrac{1}{2}(H - L)} \tag{35}$$

An inverse transformation can be used to convert back to physical units, as follows:

$$\xi = \tfrac{1}{2}\left[H(x+1) - L(x-1)\right] \tag{36}$$

The inverse transformation may be problematical unless hierarchy is maintained. For response model building, coded variables have a number of advantages over variables expressed in engineering units. Some of these will be described below.

5.1.1. Greater Clarity

Coded variables can bring a certain level of clarity to the problem that is more difficult to achieve when physical units are used, because it is easier to see the relative contributions of various terms in a model when the regression has been performed on coded variables rather than physical variables. When the variables are in physical units, the numerical values of the coefficients depend on which units are selected, and the relative importance of some terms becomes less obvious.

By centering the variable, coding ensures that zero is always within the range of every independent variable. This attaches a physical interpretation to the intercept; namely, that it is the average of the response measurements. The regression intercept does not always have a physical meaning if the independent variable range does not include zero, which can be the case if the variables are not coded. Centering also decouples slope and intercept effects. If the independent variable range contains zero, changes in the slopes of a response function (its shape) are independent of changes in the mean level of responses (the intercept). This ensures that the functional form of the response model is decoupled from issues associated with precisely determining the intercept. Bias errors therefore affect only the intercept, and not the details of how responses depend on the independent variables.

5.1.2. Orthogonality

Consider a k-regressor model that is refitted to the data as a function of $k–1$ of the regressors. If the coefficients of all retained regressors remain unchanged, we say that the discarded regressor was orthogonal to the remaining regressors. Orthogonality is a desirable property because it

ensures that the magnitude of a given coefficient is independent of additional terms that may or may not be in a model. This makes the interpretation of regressor effects independent of other terms in the model, which is helpful in understanding the underlying physics.

Consider the following simple illustration of a two-factor, two-level test matrix, where the variables are expressed in physical units (ξ_i) ranging from 1 to 10, and in coded units (x_i) ranging in the usual way from –1 to +1. Equations 63 and 64 can be used in this example to convert from physical to coded units and back by setting $L = 1$ and $H = 10$.

Table 3. Comparison of coded and physical units.

Point	Physical			Coded		
	ξ_1	ξ_2	$\xi_1\xi_2$	x_1	x_2	x_1x_2
1	1	1	1	-1	-1	-1
2	1	10	10	-1	1	1
3	10	1	10	1	-1	-1
4	10	10	100	1	1	1
		Sum:	**121**		**Sum:**	**0**

Recall that two vectors are orthogonal if the sum of their term-by-term cross products (proportional to the cosine of the angle between the two vectors) is zero. Since a test matrix is comprised of a set of vectors representing independent variable levels, the same test of orthogonality can be applied to show that x_1 and x_2, are orthogonal but ξ_1 and ξ_2 are not. This means that if a model is fitted in terms of physical units, the value of the ξ_1 coefficient will depend upon whether ξ_2 is in the model or not (and vice versa), but if the model is fitted in terms of coded units, the value of the x_1 coefficient will be the same whether x_2 is retained or not, and vice versa. This is especially relevant in the context of model reduction by the elimination of terms.

When the regressors are not orthogonal, each estimated regression coefficient is in fact a function of the true coefficients of more than one regressor. The least-squares algorithm commonly used in regression computations is designed to determine the set of coefficients that minimizes the residual sum of squares, and will often produce relatively small test matrix residuals even when the regressors are not entirely orthogonal. However, if the regressors would have assumed different values with a different set of terms in the model, the coefficients that minimize residual error based on fitting data acquired at a given distribution of sites within the design space may not predict responses at other sites with the least possible error.

A class of functions known as orthogonal polynomials has been used in aerospace response surface modeling applications to predict responses using math models for which every term is orthogonal to all other terms [6]. But absent the use of such a specialized class of functions, orthogonality is difficult to eliminate entirely in general regression applications even when the variables are coded. Coding can provides some incremental quality improvement, however.

5.1.3. Computing Resolution

Real numbers of the kind used in regression computations are a mathematical abstraction involving infinite resolution and infinite range. Computers must approximate real number calculations by using floating-point numbers that involve a finite set of values with finite

precision. The ANSI/IEEE Standard 754-1985 for Binary Floating-Point Arithmetic [7] establishes conventions for double-precision floating-point numbers that specify they be stored in 64-bit words, with 52 bits reserved for a mantissa, 11 bits reserved for an exponent, and one bit to represent the sign. The finite bit allocation for the exponent limits the range of floating-point numbers to something on the order of $\pm 10^{308}$, and the finite number of mantissa bits limits the resolution to $2^{-52} = 2.22\text{E-}16$ at best. (Actually, the mantissa and exponent interact in such a way that the spacing between realizable floating-point numbers gets larger for larger numbers). The range limitation does not generally represent a practical constraint in regression analysis but the resolution limit can affect calculations.

To illustrate how limitations in computational resolution can adversely impact the estimation of response model coefficients when factor levels are not coded, we present a simple example that requires us to look ahead briefly to the next section on regression methods. In that section, we represent the response model coefficients as a vector, β, with p elements, where p is the number of terms in the model. Each element in this vector is a coefficient for a particular model term. The vector of coefficients is a function of a matrix, $\mathbf{X}$, known as the design matrix, and a vector of measured responses, $\mathbf{y}$. The i^{th} element of this vector is the response estimated at the i^{th} fitted site in the design space. The design matrix has n rows and p columns, where n is the number of fitted points and p is the number of terms in the model. Each element of this matrix represents the value of a regressor corresponding the column of the matrix, as evaluated at a site in the design space corresponding the row of the matrix. The left-most column, corresponding to the intercept, contains all 1's.

The vector of regression coefficients, $\boldsymbol{\beta}$, is computed as follows:

$$\beta = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{y} \tag{37}$$

To illustrate the point about floating-point resolution, consider a simple example in which the design matrix is as follows:

$$\mathbf{X} = \begin{bmatrix} 1 & 0 & \delta \\ 1 & \delta & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{38}$$

This design matrix corresponds to a simple first-order math model in two variables, to be fitted with three data points. We then have:

$$\mathbf{X}'\mathbf{X} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & \delta & 0 \\ \delta & 0 & 0 \end{bmatrix}\begin{bmatrix} 1 & 0 & \delta \\ 1 & \delta & 0 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 3 & \delta & \delta \\ \delta & \delta^2 & 0 \\ \delta & 0 & \delta^2 \end{bmatrix} \tag{39}$$

If $|\delta| < 10^{-8}$, a floating-point representation of this matrix is singular even with double-precision, because the resolution limit of 10^{-16} dictated by the IEEE standard renders the second and third columns identical. The inverse would then be nonexistent and the regression coefficients could not be computed by Eq. 37. Depending on the details of the test, it is not inconceivable that this limit could be encountered when the variables are expressed in physical units. For example, simply measuring force balance outputs in volts in a calibration experiment instead of microvolts

could produce nearly singular results when the calibration equations are inverted to express loads as a function of electrical output levels. A high correlation between regressors can result in a model that does not predict responses for arbitrary independent variable combinations as well as it does for the test matrix points for which the model coefficients were optimized. Coding the independent variables circumvents these floating-point resolution issues and minimizes the potential for this source of multicollinearity.

5.2. Review of Regression

Consider a polynomial model with p terms comprised of an intercept and $k = p - 1$ additional terms. We assume that a data sample consisting of response estimates made at n factor/level combinations have been acquired, where n must be greater than k (i.e., greater than or equal to p) for all regression coefficients to be estimated including the intercept. The general form of the full polynomial model is then as follows:

$$y_i = \beta_0 + \sum_{j=1}^{k} \beta_j x_{ij} + \varepsilon_i \tag{40}$$

where y_i is the response recorded for the i^{th} factor/level combination, x_{ij} is the i^{th} level of the j^{th} regressor, β_j is the coefficient of the j^{th} regressor, and ε_i is an error term, assumed to be drawn from a normal distribution with a mean of zero and with a constant standard deviation for all responses. The quantity β_0 is an intercept term. For factor levels coded by Eq. 35, the intercept is independent of the β_j regressor coefficients and represents the mean of the sample of data that is fitted to the model.

Equation 40 can be described more succinctly in vector/matrix form as follows:

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon \tag{41}$$

where $\mathbf{y}$ is an $(n \times 1)$ vector of response measurements, $\boldsymbol{\beta}$ is a $(p \times 1)$ vector of coefficients, and $\boldsymbol{\varepsilon}$ is an $(n \times 1)$ vector of error terms.

$\mathbf{X}$ is called the design matrix, consisting of n rows corresponding to the number of data points acquired in the experiment, and p columns, one for each term in the math model, including the intercept term. The design matrix plays a central role in determining both the cost and the quality of any experiment in which a polynomial math model is used to fit system responses as a function of multiple independent variables [8]. Furthermore, the details of the design matrix are largely within the researcher's control. The number of rows depends on the volume of data that the researcher decides to acquire, the number of columns depends on the math model that the researcher selects, and the values of the individual elements of the design matrix depend on the levels of the independent variables that are set for each data point.

The design matrix is an important determinant of both productivity and quality in a response modeling experiment. Productivity is influenced by the volume of data acquired, which is defined by the number of rows in the design matrix. Direct operating costs and cycle time can both be minimized by specifying as few rows in the design matrix as will be adequate to achieve the objectives of the experiment [9]. The impact of the design matrix on quality is revealed through the covariance matrix and how it affects uncertainty both in estimates of the individual regression coefficients, and in response predictions made by applying the model.

The covariance matrix, $\mathbf{C}$, is a ($p \times p$) square matrix computed by pre-multiplying the design matrix by its transpose, inverting the product, and multiplying each element of the resulting matrix by the unexplained variance of the residuals, σ^2:

$$\mathbf{C} = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\sigma^2 \tag{42}$$

It can be shown that the diagonal elements of the covariance matrix represent the variance in estimates of the regression coefficients. That is, the variance in the i^{th} regression coefficient, β_i is simply:

$$Var\left(\beta_i\right) = \mathbf{C}_{ii} \tag{43}$$

The off-diagonal elements of the covariance matrix quantify the degree to which the regressors are correlated. We would generally prefer that the off-diagonal elements of the covariance matrix all be zero. While this ideal state is not easy to achieve in all practical circumstances, much of the quality improvement attributable to formal experiment design is derived from design space site distributions that minimize the off-diagonal elements of the covariance matrix.

Other than the intrinsic variability of the data, Eqns. 42 and 43 clearly indicate that the variance of the estimated regression coefficients depends only on the design matrix. For this reason, uncertainty in estimates of the regression coefficients can be minimized by optimizing the design matrix.

Similarly, when the model is used to predict responses for a specified combination of independent variable settings, the variance in those predictions depends on the design matrix. Consider a vector $\mathbf{x_0} = [1\ x_{01}\ x_{02}\ \ldots\ x_{0K}]'$ representing a data point specified by a given combination of factor levels, where x_{0i} is the level of the i^{th} regressor corresponding to this point. The estimated mean response at this point is

$$\hat{\mathbf{y}}\left(\mathbf{x_0}\right) = \mathbf{x_0'b} \tag{44}$$

where $\mathbf{b}$ is a vector of estimated regression coefficients. That is to say, $\mathbf{b}$ is our best estimate (typically by some least-squares criterion) of the vector of true coefficients, β, in Eq. 41.

The variance in the response prediction at a particular $\mathbf{x_0}$ is computed as follows:

$$Var\left[\hat{\mathbf{y}}\left(\mathbf{x}_0\right)\right] = \sigma^2\mathbf{x_0'}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{x_0} \tag{45}$$

As in the case of the individual regression coefficients, Eq. 44 reveals that except for the intrinsic variability in the data used to fit the response model, the variance in a response prediction for a given point is determined entirely by the design matrix. Equations 42, 43, and 45 suggest that since the design matrix plays such a central role in determining the precision of a response surface modeling experiment, improvements in the quality of such experiments can be achieved by planning the design matrix carefully.

We now present a brief general explanation for why the quality of a response surface modeling experiment can often be improved by eliminating some of the math model terms.

Assume for a moment that Eq. 41 represents the largest math model that can be supported by a given test matrix. That is, we assume that this equation describes the largest math model for which non-singular regression results can be obtained for the prescribed test matrix.

We wish to examine the consequences of reducing the model so that fewer than the *k* regressors of the model described by Eq. 41 are retained. Let *r* represent the number of regressors that we wish to remove and let *p* represent the number of terms that will be retained in the model, including the intercept term. We can then express Eq. 41 as follows:

$$\mathbf{y} = \mathbf{X}_p \beta_p + \mathbf{X}_r \beta_r + \varepsilon \tag{46}$$

Here, $\mathbf{X_p}$ is a ($p \times n$) matrix with columns corresponding to the retained terms in the model, including the intercept, and β_p is a ($1 \times p$) vector of the corresponding regression coefficients for this reduced model. The columns of $\mathbf{X_r}$ represent terms that are deleted from the model, and β_r is a vector of the corresponding regression coefficients.

If $\mathbf{b}$ is a vector of estimated regression coefficients for the unreduced model, and $\mathbf{b_p}$ corresponds to those coefficients that are retained, it can be shown that the matrix $Var(\mathbf{b_p}) - Var(\boldsymbol{\beta}_p)$ is positive semidefinite [10]. Therefore, dropping terms from the full model and refitting the data to a subset of the original regressors results in model coefficient estimates with variance that is less than or equal to the variance in the corresponding coefficients of the full model. In other words, with respect to the precision of the regression coefficient estimates there is nothing to lose, and possibly something to gain, by reducing the number of regressors in the math model.

Consider now the impact of such a model reduction on the variance of response predictions. Note that a vector of predicted responses for each point in the test matrix can be generated from the vector of measured responses, $\mathbf{y}$, by means of the "hat matrix," $\mathbf{H}$, as follows:

$$\hat{\mathbf{y}} = \mathbf{H}\mathbf{y} \tag{47}$$

where

$$\mathbf{H} = \mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}' \tag{48}$$

and $\mathbf{X}$ is the design matrix, as before. The variance in the vector of response estimates is computed as follows:

$$Var\left(\hat{\mathbf{y}}\right) = \mathbf{H}' Var\left(\mathbf{y}\right)\mathbf{H} = \mathbf{H}'\left(\mathbf{I}\sigma^2\right)\mathbf{H} \tag{49}$$

The hat matrix is both symmetric (equal to its transpose) and idempotent, meaning that $\mathbf{HH} = \mathbf{H}$. Equation 49 therefore reduces to

$$Var\left(\hat{\mathbf{y}}\right) = \mathbf{H}\sigma^2 \tag{50}$$

Note that the variance of the i^{th} response prediction is just the i^{th} diagonal element of $\mathbf{H}\sigma^2$. Following Box and Draper [11], we consider the trace of this matrix, which is just the sum of all the diagonal elements:

$$trace\left(\mathbf{H}\sigma^2\right) = \sigma^2 trace\left[\mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\right] = \sum_{i=1}^{n} Var\left(\hat{y}_i\right) \tag{51}$$

We invoke the following matrix identity: trace(**AB**) = trace(**BA**). Let **A** = **X** and **B** = **(X'X)$^{-1}$X'**. Then,

$$\text{trace}\left[\mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\right] = \text{trace}\left[\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{X}\right] = \text{trace}\left(\mathbf{I_p}\right) = p \tag{52}$$

since p is the dimension of the square matrix **(X'X)$^{-1}$**, as noted above in the description of the covariance matrix. Combining Eqns. 51 and 52 we have:

$$\sum_{i=1}^{n} \text{Var}\left(\hat{y}_i\right) = p\sigma^2 \rightarrow \frac{\sum_{i=1}^{n} \text{Var}\left(\hat{y}_i\right)}{n} = \frac{p\sigma^2}{n} \tag{53}$$

The prediction variance averaged over all points in a regression analysis is the same for any order model and depends only on the intrinsic variability of the data, the volume of data fitted, and the number of terms in the math model. Therefore, if a given number of data points are acquired in a measurement environment with a given degree of intrinsic variability, the average prediction variance depends on nothing more than the number of terms in the regression model. Therefore, there is potential to reduce the average prediction variance simply by reducing the number of terms in the model. This is because each term in the model carries with it some contribution to the total uncertainty in the prediction.

To summarize the key points, the design matrix determines the uncertainty in estimates of the regression coefficients and the model predictions. Reducing the number of columns in the design matrix by selecting a reduced math model never increases the variance in estimates of the retained coefficients, and may decrease it. Because the prediction variance averaged over all regression points is directly proportional to the number of terms in the math model, reducing the number of model terms also increases the precision of the response predictions.

5.3. Term Reduction

Equation 53 shows that the average prediction variance across all fitted points is directly proportional to p, the number of terms in the response model. Quite often, not all of terms in a full d^{th}-order polynomial model in k factors are required to achieve an adequate fit to a particular sample of data, so the potential exists to reduce the prediction variance by eliminating some terms.

For example, a full fourth-order polynomial in six factors has 210 terms, but there is a general rule known as the sparsity of effects principle, which states that the lower-order terms account for most of the predicted response. For coded factors, the intercept (zeroth-order term) describes the mean level of the data, the first-order terms describe changes in that level, the second-order terms describe changes in such changes; the third-order terms describe changes to those changes in changes, and so on. These changes generally peter out eventually.

Dropping terms of sufficiently high order from the model can have relatively little downside when the higher-order terms have progressively smaller impact on the model prediction. However, using Eq. 53, the average prediction variance is directly proportional to the term count,

so dropping terms can result in a net increase in the effective signal-to-noise ratio of model predictions, enabling smaller effects to be detected with a given level of confidence.

Note that in the example considered here, in which the model is fourth-order in six factors, over half of the terms—126 of 210—are fourth-order. While some of these fourth-order terms may be needed to achieve a good fit over a moderate range of independent variables, and some terms of even higher order may improve the fit, it is likely that many of the fourth-order terms (and some of the lower-order terms) do not even exist in nature, and are only non-zero because of random errors in the fitted data.

It requires some experience and subject matter expertise to make an initial estimate as to how complex a response model to fit to a given sample of data. One might try to estimate how high an order of polynomial is necessary to span a design space of given dimensions, but a better approach might be to first declare the order of model—fourth-order in this example—and then rely on experience and subject matter expertise to define limits on the design space within which such a polynomial is likely to be satisfactory. If these limits are not great enough to span the entire design space, it can be partitioned into subspaces. Response models developed in these subspaces can be of lower order. They also permit different models to describe regions of the design space where the underlying physics might be sufficiently different as to make a single high-order response model spanning all of them more difficult to interpret.

If a candidate model is of too low an order to fit the data adequately, it will generate biased predictions that result in systematic (not random) lack-of-fit errors. These errors can be detected during the model assessment phase of the experiment. In this phase, model predictions are compared with direct response estimates at selected validation sites throughout the design space, where data are acquired to test the model and are not used in the original fitting of the model. Model assessment will be discussed in Section 6, but if lack-of-fit is detected, it can be addressed by fitting a different model (first acquiring additional data if necessary), or by fitting the current model over a reduced factor range. Note that the experiment will have been scaled, by acquiring a sufficient volume of data per Section 3, to provide the resolution necessary to detect lack-of-fit errors of a specified size.

Response models are generally constructed with the aid of software designed for this purpose. Several commercial software packages are available [12-16], as well as packages developed by NASA [17], Sandia Labs [18], and other organizations. In addition to automating the vector-matrix calculations summarized above, by which model coefficients are estimated as well as the variance for each coefficient and the covariance between pairs of coefficients, such software can also systematically evaluate the impact of eliminating selected terms from the model. This process, known as term reduction, is outlined here.

Computerized regression procedures generally produce a list of model terms with corresponding *p*-statistics. Here, *p* does not refer to the term count, but to an inference-error probability associated with a null hypothesis claiming that the true value of the coefficient for this term is zero. The *p*-statistic represents the probability that the coefficient for this term is as large as it is simply due to random error. This means it represents the probability of an inference error if one were to infer that the true coefficient is non-zero. A very low *p*-stat suggests that there is a low probability that the true coefficient is zero, which suggests that this term should be retained in the model.

The simplest term reduction strategy, then, would be to manually discard all terms from the model with *p*-stats above a specified level, such as 0.05. This would ensure that one could reject

the null hypothesis that the true coefficient of a retained model term is zero with no more than a 5% chance of an inference error. This would seem to imply that the decision to retain a term in the model is made with 95% confidence.

Manual term reduction assumes that all of the model coefficients are independent. Unfortunately, there is almost always some degree of correlation among the model coefficients, a condition arising out of what is known as multicollinearity. This means that the *p*-stat of one term may change if another term is removed from the model. The degree of correlation can be minimized by judicious site selection for the fitted data points, and while the site selection optimization schemes described earlier do seek to minimize multicollinearity, there is always a risk that dropping one term will affect the decision to drop or retain other terms.

We describe three methods that are designed to cope with multicollinearity during model reduction, which are variations within a general class of methods known as stepwise-type procedures. The three methods are known as Forward Selection, Backwards Elimination, and Stepwise Regression, the last of which is actually a combination of the first two methods. This list is intended to be representative rather than exhaustive; the reader can consult the literature for other term reduction methods.

Forward Selection begins by assuming that only the intercept term is in the model. It then provisionally adds the one term that has the highest correlation with the response. Every model, including this two-term model, explains a certain portion of the total variance in sample of fitted data. Absent random error and assuming a perfect model, all of the variance in a sample of fitted data would be explained by the model itself. The popular R^2 statistic of linear regression represents the fraction of the total variance that the model explains.

The explained variance increases each time a term is added provisionally to the model. The degree of this increase can be quantified by dividing it by the explained variance before the term was added. The variances are chi-square distributed and the ratio of two chi-square distributed quantities follows an F-distribution, which has associated with it a p-statistic that represents the probability that this term makes an insignificant change to the explained variance of the model. A low p-stat indicates that the magnitude of the ratio is systematic and not random, which in this case indicates that adding the term to the model has resulted in a true increase in explained variance. This suggests that the added term should be retained in the model, in which case, the forward selection process continues.

The next term to be provisionally entered into the model is the one that has the highest correlation with the response after correcting for the first term entered. This is called a partial correlation, and the associated p-value indicates as before the probability that this term makes an insignificant change to the explained variance of the model. If it is below a specified entry criterion, this term is retained. The process continues until either all terms from the initial candidate model have been included, or until the most significant remaining term does not cause enough of a change in the explained variance to satisfy the p-stat entry criterion.

Backward Elimination is similar to forward selection except that the process attempts to identify terms for the final model by working in the opposite direction. The backward elimination method begins with all terms from the full d^{th}-order model in k factors included. The term with the weakest correlation with the response is provisionally rejected, and the impact on the explained variance of the model is again assessed by an F-test. If the rejection of this term produces a significant reduction in the variance explained by the model, it is retained and the process stops. Otherwise, the process continues until no terms in the model can be rejected without causing a

significant reduction in the variance explained by the model, or until the only remaining term is the intercept.

Stepwise Regression is a combination of forward selection and backward elimination. We begin as a forward selection process and continue until the model contains the intercept and two regressors. Now apply backward elimination to the three-term model, eliminating each regressor in turn to assess the reduction in the model's explained variance via an F-test. When the backward elimination step is finished, resume the forward selection process with whichever remaining term causes the most significant increase in the explained variance. If such a term increases the explained variance by a user-specified threshold amount, retain it and initiate the backward elimination on the new model. Continue until there are no candidate regressors significant enough to enter the model and none in the model that are so weak that they can be eliminated with no significant effect.

Proponents of stepwise regression note that the backward elimination step protects against multicollinearity. If two regressors are highly correlated, adding one of them to the model may render the first one superfluous. In such a case, it is better to eliminate that term.

It is common to apply more than one stepwise procedure to the same data set. For example, one might use backward elimination first, to give all model terms a chance to be included in the reduced model, then apply stepwise regression to the surviving model terms, and finish with one more application of backward elimination, to reject the occasional high-order term that survives the first two applications without contributing significantly to the model. Textbooks treating regression analysis [10, 20] provide a more detailed description of the stepwise procedures outlined here.

This section has described the implementation of stepwise regression procedures. To summarize this process, a full candidate d^{th}-order polynomial model is proposed as a starting point. Terms from this model are discarded or retained by the stepwise procedures, which seek to minimize unexplained variance and multicollinearity. Individual terms can then be added or deleted to fine tune the final model by attempting to incrementally improve the fit, to resolve multicollinearity issues, or to reflect the user's judgment and subject matter expertise. Individual terms are also commonly added to the reduced model in order to impose hierarchy, which is described in the next section.

5.4. Hierarchy

A model is said to be hierarchical if it contains all the components of terms that are second-order and higher. For example, if an AB interaction term is included in the model, both the A and the B first-order terms must also be included to maintain hierarchy. Likewise, if terms of the form A^2B are in the model, then so must A, B, and AB for the model to be hierarchical.

Hierarchy becomes especially relevant when the independent variables have been subjected to coding transformations that include both scaling and translation, as the transformations of Eqns. 35 and 36 do. Peixoto [21] describes a general coding transformation consisting of a combination of scaling and translation, and demonstrates that while scaling transformations do not change the estimation space of a polynomial function of two or more variables, the estimation space of such a polynomial is invariant under translation if and only if the polynomial is hierarchical.

This means that hierarchically inferior terms must be retained in order to maintain hierarchy and preserve the invariance of the coding transformation, even if they are statistically insignificant.

This is especially relevant when some multicollinearity is present, so that the coefficients are not all independent. In such a case, the coefficient of a term of second order or higher could be a function of the coefficients of its hierarchically inferior components, so that removing them from the model skews the coefficient of some of the retained terms.

Also, the residual sum of squares of a nonhierarchical model includes components due to the missing hierarchically inferior terms. This makes the residual variance less representative of the true experimental error. For this reason, some commercial packages (e.g., Minitab [13]) will not perform an analysis of variance on nonhierarchical models. Design-Expert [12] will permit such an analysis, but only under duress. Whenever a model is generated with missing hierarchically inferior terms, Design-Expert generates a prompt recommending that hierarchy be imposed. If the user declines, Design-Expert provides a warning and a second prompt. If the user still declines the option to make the model hierarchical, Design-Expert generates the model, but with this disclaimer:

> *"Using this non-hierarchical polynomial regression model (it excludes hierarchically inferior terms) is not recommended. Measures of goodness of fit and the predicted response values may be not be the same as those from the coded equation. All analysis within Design-Expert software is based on the coded equation."*

Draper and Smith [20] also argue against dropping hierarchically inferior terms under a translation of origin. They propose the following rule:

> *"If a model is to be consistent under a shift in origin, only the highest-order terms can be deleted at first and any chosen deletions must keep the model well-formulated. Moreover, if any of the highest-order terms are retained, all terms of lower order affected by them in a shift of origin must also be retained, whether or not their estimates are significant in the regression fit."*

Draper and Smith also provide guidelines for removing terms when a rotational transformation is applied, as is commonly the case when certain canonical forms are invoked in the analysis of response surface models in order to remove cross-terms from the model. This is useful when it is of interest to determine variable combinations that correspond to a response maxima or minima, for example.

It is important to note that even seemingly innocuous variable transformations can have unintended consequences that must be thoroughly understood. For example, Peixoto's original work on this subject was motivated by a study of average daily temperatures in 56 U.S. cities as a function of longitude and latitude in which he simply translated the origin of the longitudinal variables to center them in the United States. This ostensibly benign change was the equivalent of redefining the origin of longitude measurements to pass somewhere near St. Louis rather than through Greenwich, a change that would hardly be expected to influence temperature predictions. Nonetheless, the functional form of non-hierarchical third-order polynomial functions of longitude did change under this origin translation, while the functional form of hierarchical models remained the same.

Peixoto's temperature modeling example illustrates that nonhierarchical models are at a disadvantage under any translation of the origin, not just transformations used to code the variables. Draper and Smith suggest that models generated by automated selection procedures

such as those described in the previous subsection should be augmented as necessary to impose hierarchy.

5.5. Transformation of Response Variables

A hierarchical model with the fewest possible terms might be further improved by transforming the response variable. For example, there are data sets for which a better fit can be achieved by modeling the log of the response, say, or the square root. As an example, consider results obtained in a computational experiment conducted as part of the Space Shuttle Return to Flight program after the Columbia accident.

On February 1, 2003, the Space Shuttle Columbia was lost on reentry. A piece of foam insulation from the shuttle's propellant tank had broken off on ascent and breached the leading edge of the shuttle's left wing. This allowed gasses that were heated by reentry friction to bypass the thermal protection system and destroy the wing.

Computer codes were subsequently developed to model the thermal and structural reentry loads associated with debris impact on ascent but the uncertainty in reentry load predictions was unknown. These loads depend in a complex way on many independent variables—over 500 potentially significant main effects and interactions, each of which may or may not have been significant, and each of which contributed in an unknown way to the total prediction uncertainty.

The computer codes were approximated with simpler response models that could be used to propagate various uncertainties into estimates of the total thermal and structural load uncertainty [22]. These models were developed with computational experiments in which better fits to the data were achieved by transforming the response variables.

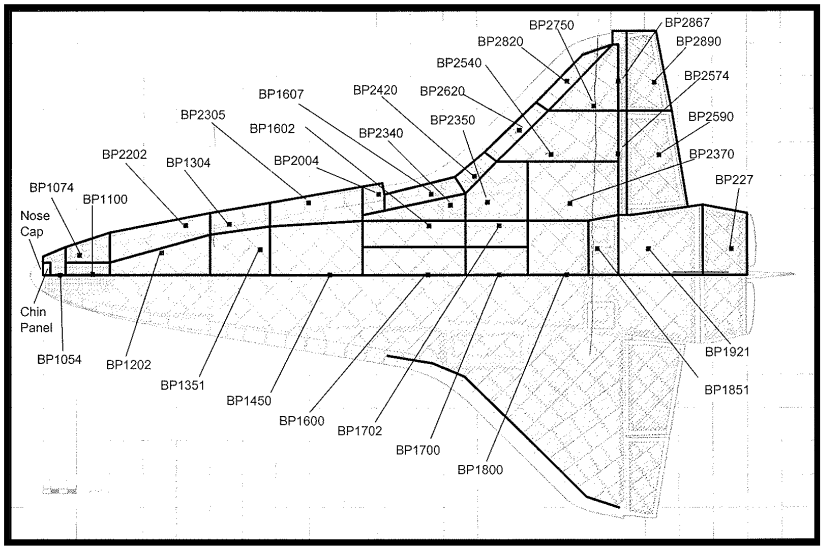


Figure 9. Body Point Zones for the Shuttle Thermal Protection System.

Figure 9 displays Body Point Zones that were used to characterize thermal and structural reentry loads. Models were developed in each of the zones for a number of response variables. Each such model was tested to see if the fit to the data could be improved with a power transformation of the response variable, using the method of Box and Cox [23].

The Box-Cox transformation test examines power transformations, y^{λ}, of the response variables, y. Models are fitted for each λ in a range of candidate λ values. For each λ, residuals are generated, squared, and summed. A plot is generated of the natural log of this sum of squared residuals vs. λ. The point at which this function is a minimum is a maximum likelihood estimator of the exponent that produces the best fit in a least-squares sense.

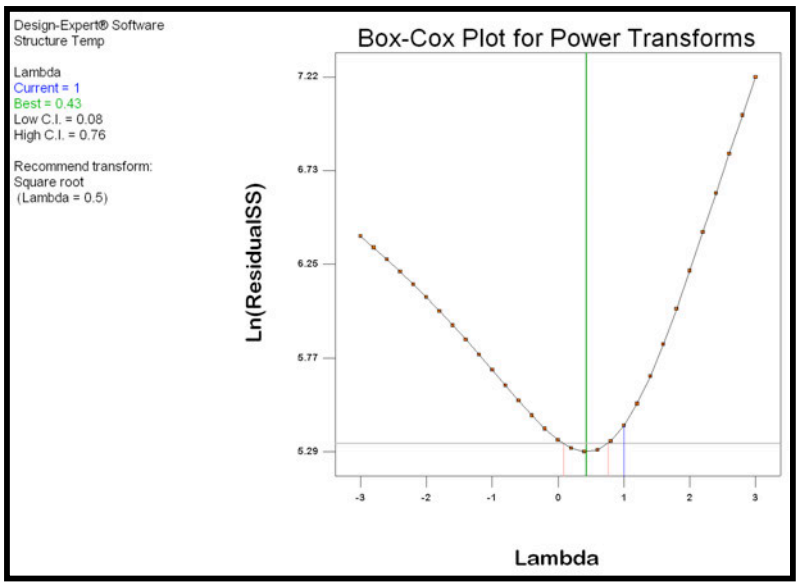


Figure 10. Box-Cox power transformation plot. Shuttle structural temperature in body point zone 1800.

Figure 10 is a Box-Cox transformation plot for the structural temperature model developed for body point zone 1800. It was developed using the Design-Expert software package cited above. It features a minimum at $\lambda = 0.43$, with 95% confidence interval limits for λ of [0.08: 0.76]. This interval contains the convenient value of 0.5, suggesting that, within experimental error, a square root transformation of the structural temperature response predictions would yield a better fit to the data. Also note that the upper limit of this confidence interval is less than 1, implying that the fit can be improved over the untransformed case for which $\lambda = 1$.

There was an interesting result in the power transformation investigations, illustrated in Fig. 12. This is a plot of the temperature at each of the body point zones for the room temperature vulcanized (RTV) adhesive bond used to bond the tiles of the Shuttle Thermal Protection System to the shuttle's surface. If the temperature of this bond exceeds a certain threshold, the bond will fail and the tile will separate.

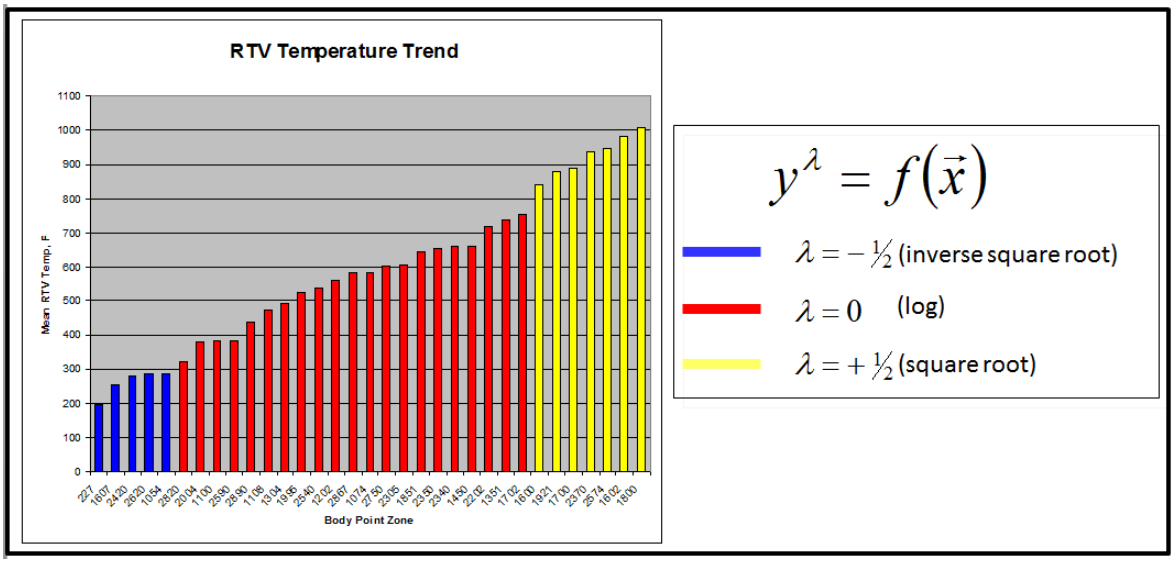


Figure 11. Trend in power transformations of the average RTV adhesive temperature variation, with body point zones.

Figure 11 reveals that the coolest body point zones (with mean temperatures of less than 300° F) were described by RTV temperature models that were improved using inverse square root transformations of the response variables, $\lambda = -1/2$. Response models were improved in body point zones with mid-range RTV adhesive temperatures (nominally between 300° F and 800° F) when a logarithmic transformation was used ($\lambda = 0$). For the body point zones with the highest temperatures (above 800° F), a square root transformation, $\lambda = +1/2$, improved the fit.

The fit is improved by a transformation that makes the residuals more normally distributed with a constant standard deviation. This is because linear regression assumes normally distributed errors with a constant standard error. A class of transformations known as "variance stabilization transformations" serves to make the unexplained variance associated with transformed responses more constant, which adheres more closely to basic assumptions in least-squares regression analysis and therefore often generates better models.

Consider, for example, a response for which the standard deviation of random errors is not constant, but is directly proportional to the response variable. Assume we have the following:

$$\sigma_y = ky \tag{54}$$

Let us apply a logarithmic transformation to the responses to produce a transformed response, $w = ln(y)$. Compute σ_w as follows:

$$\sigma_w^2 = \left(\frac{\partial w}{\partial y}\right)^2 \sigma_y^2 = \left(\frac{1}{y}\right)^2 (ky)^2 = k^2 \tag{55}$$

so $\sigma_w = k$, a constant, and the fit to $w = ln(y)$ should be improved. This is precisely what occurred with RTV adhesive temperature models developed in body zones exposed to mid-range temperatures between 300° F and 800° F, where the Box-Cox transformation test produced a plot with the smallest residuals for a power transformation exponent of $\lambda = 0$.

It is not immediately obvious that a power transformation with an exponent of zero corresponds to a logarithmic transformation, but this result derives from the fact that within the limit as λ approaches zero, changes in y^{λ} and *log(y)* are progressively less distinguishable. This means for small lambda, y^{λ} plots against log(y) very nearly as a straight line [11].

The general power transformation can be written as $w = y^{\lambda}$, so the variance in w is

$$\sigma_w^2 = \left(\frac{\partial w}{\partial y}\right)^2 \sigma_y^2 = \left(\lambda y^{\lambda-1}\right)^2 \sigma_y^2 \tag{56}$$

The condition for the fit of transformed responses, w, to produce smaller residuals is that σ_w is constant. Let $\sigma_w = C$, a constant. Then,

$$C^2 = \left(\lambda y^{\lambda-1}\right)^2 \sigma_y^2 \tag{57}$$

The Box-Cox transformation test reveals that response model fits are improved in the relatively cool body point zones when the response estimates are transformed with a y^{λ} power transformation for which $\lambda = -1/2$, suggesting that the standard error of the residuals for the $w = y^{-1/2}$ transformed response is a constant. For this case Eq. 57 becomes:

$$\lambda = -\frac{1}{2} \rightarrow C^2 = \left(-\frac{1}{2} y^{-3/2}\right)^2 \sigma_y^2 \rightarrow C = \frac{\sigma_y}{2y^{3/2}} \rightarrow \sigma_y \propto y^{3/2} \tag{58}$$

We infer from this that the error in RTV adhesive temperature estimates increases fairly rapidly with temperature for relatively low RTV adhesive temperatures.

At relatively high temperatures, the Box-Cox test suggests that the variance stabilization power transformation requires a lambda value of +1/2. For that case we have

$$\lambda = +\frac{1}{2} \rightarrow C^2 = \left(\frac{1}{2} y^{-1/2}\right)^2 \sigma_y^2 \rightarrow C = \frac{\sigma_y}{2y^{1/2}} \rightarrow \sigma_y \propto y^{1/2} \tag{59}$$

This indicates that the error in RTV adhesive temperature estimates increases with temperature for relatively high temperatures, just as it did for relatively low temperatures, but at a lower rate. In between, in the mid-range temperatures, the prediction uncertainty is independent of temperature, by Eq. 55.

The Shuttle Return to Flight example illustrates that transformations of the response variables can not only improve the fit of candidate response models, but can also yield insights into the underlying physical processes under study. In the case of the RTV adhesive temperature models, it is clear that prediction uncertainty depends on temperature for relatively low and relatively high temperatures, and is independent of temperature in the mid-ranges. It is also clear that the dependence of prediction uncertainty on temperature is greater for lower temperatures than higher temperatures.

5.6. Transformation of Independent Variables

The previous section examined how the fit could be improved by invoking variance stabilization transformations of the response variables. These transformations are designed to make the model residuals conform better to the theoretical assumptions of linear regression, specifically, assumptions that the fitted response data are drawn from a sample with a constant standard deviation. We will consider whether some transformation of the independent variables might provide similar improvements.

We use, as an example, a model of vehicle stopping distance as a function of velocity and coefficient of friction. This set of computer-generated data was used by the author of this chapter to train colleagues how to use formally designed experiments in a program to evaluate a passive runway arresting system under consideration at the JFK International Airport [24]. The system consisted of runway treatments with various materials capable of preventing an aircraft from overshooting the runway on approach.

Table 4. Vehicle Stopping Distance Data.

Run	Velocity (ft/sec)	Coefficient of Friction	Z_1=0.064Vln(V)	Z_2=-1.758μln(μ)	Stopping Distance, ft
1	44	0.4	10.664	0.644	75.3
2	22	0.4	4.355	0.644	18.5
3	44	0.8	10.664	0.314	37.7
4	33	0.6	7.390	0.539	28.7
5	33	0.6	7.390	0.539	28.3
6	33	0.6	7.390	0.539	29.1
7	44	0.8	10.664	0.314	37.5
8	44	0.4	10.664	0.644	75.5
9	22	0.4	4.355	0.644	19.1
10	22	0.8	4.355	0.314	9.5
11	22	0.8	4.355	0.314	9.2
12	33	0.6	7.390	0.539	28.2
13	17.4	0.6	3.194	0.539	7.9
14	33	0.6	7.390	0.539	28.2
15	48.6	0.6	12.074	0.539	61.7
16	33	0.9	7.390	0.193	19.1
17	33	0.3	7.390	0.640	53.0
18	33	0.6	7.390	0.539	28.0

In this small data set, eighteen combinations of velocity and friction coefficient are tabulated, with associated stopping distances. Two columns labeled Z_1 and Z_2 contain numbers that are generated in the analysis that follows.

These data were originally fitted to a second-order polynomial in the two factors, V (velocity) and μ (friction coefficient). A Box-Cox transformation test, as described in the previous section, suggests that a logarithmic transformation of the predictor variable (stopping distance) would result in a better fit than a model in untransformed stopping distances. The transformed model was of this form, where the b_i variables are regression coefficients:

$$Y = \ln(y) = b_0 + b_1 V + b_2 \mu + b_{12} V \mu + b_{11} V^2 + b_{22} \mu^2 \tag{60}$$

In spite of the improvement attributable to the logarithmic predictor variable transformation, there was still substantial lack-of-fit. We therefore ask if a transformation of the predictor variables might improve the fit. Specifically, we seek to reduce the average prediction variance by transforming the independent variables into factors that can be used to construct an adequate second-order model, reducing the average prediction variance by decreasing the number of terms in the model. Note that, using Eq. 53, if we can fit the stopping distance data to a first order model (three terms) instead of a quadratic model (six terms), we can cut the average prediction variance in half. We seek to do this by fitting transformed versions of the independent variables by asking what values of α_1 and α_2 will yield a better fit to the following model:

$$Y = \beta_0 + \beta_1 V^{\alpha_1} + \beta_2 \mu^{\alpha_2} \tag{61}$$

where Y is the natural log of stopping distance.

Following Box and Tidwell [25], we expand Y as a first-order Taylor's series about $\alpha_1=\alpha_2=1$ to produce this approximation:

$$Y = \left[\beta_0 + \beta_1 V^{\alpha_1} + \beta_2 \mu^{\alpha_2}\right]_{\alpha_1=\alpha_2=1} + (\alpha_1 - 1)\left[\frac{\partial Y}{\partial \alpha_1}\right]_{\alpha_1=1} + (\alpha_2 - 1)\left[\frac{\partial Y}{\partial \alpha_2}\right]_{\alpha_2=1} \tag{62}$$

The derivatives on the right of Eq. 62 are evaluated as follows:

$$Y = \beta_0 + \beta_1 V^{\alpha_1} + \beta_2 \mu^{\alpha_2} \rightarrow \left[\frac{\partial Y}{\partial \alpha_1}\right]_{\alpha_1=1} = \left[\frac{\partial V(\alpha_1)}{\partial \alpha_1}\right]\left[\frac{\partial Y}{\partial V}\right]_{\alpha_1=1} \tag{63}$$

$$V(\alpha_1) = V^{\alpha_1} = e^{\alpha_1 \ln(V)} \rightarrow \left[\frac{\partial V(\alpha_1)}{\partial \alpha_1}\right] = V^{\alpha_1}\ln(V) \tag{64}$$

$$\left[\frac{\partial Y}{\partial \alpha_1}\right]_{\alpha_1=1} = \left[\frac{\partial Y}{\partial V}\right]_{\alpha_1=1} V\ln(V) \tag{65}$$

And likewise:

$$\left[\frac{\partial Y}{\partial \alpha_2}\right]_{\alpha_2=1} = \left[\frac{\partial Y}{\partial \mu}\right]_{\alpha_2=1} \mu\ln(\mu) \tag{66}$$

We therefore have to first order:

$$Y = \beta_0 + \beta_1 V + \beta_2 \mu + (\alpha_1 - 1)\left[\frac{\partial Y}{\partial V}\right]_{\alpha_1=1} V\ln(V) + (\alpha_2 - 1)\left[\frac{\partial Y}{\partial \mu}\right]_{\alpha_2=1} \mu\ln(\mu) \tag{67}$$

which can be simplified as:

$$Y = \beta_0 + \beta_1 V + \beta_2 \mu + (\alpha_1 - 1)Z_1 + (\alpha_2 - 1)Z_2 \tag{68}$$

where:

$$Z_1 = \left[\frac{\partial Y}{\partial V}\right]_{\alpha_1=1} V\ln(V) \tag{69a}$$

$$Z_2 = \left[\frac{\partial Y}{\partial \mu}\right]_{\alpha_2=1} \mu\ln(\mu) \tag{69b}$$

From Eq. 61, we have

$$Y = \beta_0 + \beta_1 V^{\alpha_1} + \beta_2 \mu^{\alpha_2} \rightarrow \left[\frac{\partial Y}{\partial V}\right]_{\alpha_1=1} = \beta_1 \tag{70a}$$

and

$$Y = \beta_0 + \beta_1 V^{\alpha_1} + \beta_2 \mu^{\alpha_2} \rightarrow \left[\frac{\partial Y}{\partial \mu}\right]_{\alpha_2=1} = \beta_2 \tag{70b}$$

Insert Eqns. 70 into Eqns. 69:

$$Z_1 = \beta_1 V\ln(V) \tag{71a}$$

$$Z_2 = \beta_2 \mu\ln(\mu) \tag{71b}$$

The β coefficients can be estimated by fitting Eq. 61 to the data in Table 4:

$$Y\big|_{\alpha_i=\alpha_2=1} = \beta_0 + \beta_1 V + \beta_2 \mu = 2.242 + 0.064V - 1.758\mu \tag{72}$$

and so Eqns. 71 become:

$$Z_1 = +0.064V\ln(V) \tag{73a}$$

$$Z_2 = -1.758\mu\ln(\mu) \tag{73b}$$

Eqns. 73 are used to compute the Z_1 and Z_2 numbers listed in Table 4. We have, to the first-order, the following:

$$Y = \ln(y) = \beta_0 + \beta_1 V + \beta_2 \mu + \left(\alpha_1 - 1\right)Z_1 + \left(\alpha_2 - 1\right)Z_2 \tag{74}$$

which, when fitted to the data in Table 4, becomes:

$$Y = 1.163 + 0.356V - 2.531\mu - 1.020Z_1 - 0.941Z_2 \tag{75}$$

The standard errors in estimating the Z_1 and Z_2 coefficients are:

$$\sigma_1 = 0.042 \tag{76a}$$

$$\sigma_2 = 0.084 \tag{76b}$$

So within $\pm 1\sigma$, the optimum exponents for power transformations of the predictor variables are:

$$\alpha_1 = \left(-1.020 + 1\right) \pm 0.042 = -0.020 \pm 0.042 \tag{77a}$$

$$\alpha_2 = \left(-0.941 + 1\right) \pm 0.084 = +0.051 \pm 0.084 \tag{77b}$$

Since zero lies within the confidence interval for both power transformation exponents, we infer that $\alpha_1 = \alpha_2 = 0$, within experimental error. Recall from the previous section describing the Box-Cox power transformation test, y^λ approaches $ln(y)$ for small λ, so for small α_1 and α_2, Eq. 61 becomes

$$Y = \ln\left(y\right) = \beta_0 + \beta_1 \ln(V) + \beta_2 \ln(\mu) \tag{78}$$

where y is the stopping distance. This can be expressed as follows:

$$\ln\left(y\right) = \ln\left(e^{\beta_0}\right) + \ln(V^{\beta_1}) + \ln(\mu^{\beta_2}) = \ln\left(e^{\beta_0} V^{\beta_1} \mu^{\beta_2}\right) \tag{79}$$

or

$$y = e^{\beta_0} V^{\beta_1} \mu^{\beta_2} \tag{80}$$

Take the logarithm of both sides of Eq. 80:

$$\ln\left(y\right) = \beta_0 + \beta_1 \ln\left(V\right) + \beta_2 \ln\left(\mu\right) \tag{81}$$

Through regression on the measured data, we find these values for the coefficients, as shown in Table 5.

Table 5. Regression coefficients in factor transformation calculations.

Coefficient	Estimate	Std Err	Convenient Value
β_0	-4.170	0.036	--
β_1	2.003	0.010	2
β_2	-0.998	0.010	-1

Insert the coefficients from Table 5 into Eq. 80:

$$y = k' \frac{V^2}{\mu} \tag{82}$$

where $k' = e^{-4.170} = 0.01545$ is a constant.

Eq. 82 features less prediction uncertainty than that Eq. 60. It also reveals insights that are not as clear in a generic polynomial response model. Note that μ is dimensionless, V has units of ft/sec, and y has units of ft. This means that k' must have units of sec^2/ft, the reciprocal of acceleration. Let $k=1/k'$ be a constant with units of acceleration. Then:

$$y = \frac{V^2}{k\mu} \tag{83}$$

Where $k = e^{-\beta 0} = e^{4.170} = 64.72$ ft/sec^2. Because of experimental uncertainty in b_0, we can write k as $[1.953:2.096]g$, where $g = 32$ ft/sec^2 is the acceleration of gravity. So within experimental error, $k = 2g$, and:

$$y = \frac{V^2}{2g\mu} \tag{84}$$

This is a well-known result, so it was not necessary to use response surface modeling methods to analyze a sample of data in order to derive Eq. 84. However, it can be helpful as a training exercise, which this example was designed to be, to arrive at a familiar result when a new method is used to get there.

6. Quality Assessment

Data are fitted to a mathematical model by methods such as linear regression, with the expectation that such a model will adequately predict responses of interest for specified sites in the design space of the experiment. We define an adequate prediction as one that can be said to be within a prescribed tolerance with a specified level of confidence. We test to see if these expectations can be confirmed in the quality assessment phase of an experiment, described in this section.

6.1. Residual Patterns

Except for information that may be available outside of the experiment, everything that can be learned about the quality of a response surface modeling experiment comes from the residuals. Plotting the residuals in certain ways can be especially instructive, as will be illustrated in this section.

It is a standard practice to plot residuals against predicted response levels. Such plots are expected to exhibit no functional dependence on the level of the predicted response. If this is not the case—if the residuals are proportional to the predicted response, for example—it is important to know this is because the least squares calculations assume a uniform variance for all data points. If this assumption does not hold, the coefficient estimates will not be unbiased estimators of the true

model coefficients. In such a case, a variance stabilization transformation of the response variables often addresses the problem, as discussed in the earlier section on stepwise procedures. Recall that the natural log of the response was shown to have a more constant variance when the untransformed variance depends on the magnitude of the response.

If the model fits the data poorly, a quadratic or higher-order dependency of the residuals will be apparent when they are plotted against predicted levels. To see why that is so, consider a math model with a poor fit. Let

$$y = f(\mathbf{x}, \mathbf{b}) \tag{85}$$

be the current (poor) model of a response, y, as a function of a vector of independent variables, $\mathbf{x}$, using a set of regression coefficients, $\mathbf{b}$.

Following the work of Box and Draper [11], assume that this model fits some other set of response data well, instead of the current data. Assume there is some response (w, an unknown function of y), which this model fits better:

$$w = f(\mathbf{x}, \mathbf{b}) \tag{86}$$

Even though w is an unknown function of y, we can expand it over a suitably limited range as a Taylor series. If the data set that would correspond better to the current model is sufficiently close to the current data set (if the current model is only marginally suboptimal), we can neglect third- and higher-order terms in the Taylor series, as follows:

$$w = a_0 + a_1 y + a_2 y^2 \tag{87}$$

Equate 86 and 87, divide by a_1, and absorb a_0 into the intercept term of the coefficients vector, $\mathbf{b}$. Renaming the coefficients leads to a function of this form:

$$y = \alpha y^2 + f(\mathbf{x}, \mathbf{b}') \tag{88}$$

This relationship holds for all n data points, so we have

$$y_i - f(\mathbf{x}_i, \mathbf{b}') = \alpha y_i^2, \qquad i = 1, 2, \ldots, n. \tag{89}$$

The term on the left can be estimated by residuals of the fitted model, and the quantity y_i on the right can be estimated from model predictions. So if the current model is not a good fit and would therefore fit another data set better than the current one, we would expect to see a characteristic quadratic dependence in the plot of residuals against model predictions, per Eq. 89.

Figure 12 illustrates the quadratic nature of plotted residuals when the model is inadequate. These data are from a six-component force balance calibration experiment for which a full second-order response model would have 28 terms. Eleven terms were rejected from the normal-force model using stepwise term reduction procedures outlined earlier in this chapter. The residuals from the resulting 17-term model are plotted against response predictions in Fig. 12a and show no particular dependence on predicted response, suggesting that this model fits the data. The residuals in Fig. 12b are from the full 28-term model without the six pure quadratic terms. That is, terms of the form A^2 were excluded from the model, with only first- and second-order terms of

the form AB (interaction terms) retained. Note the characteristic quadratic shape of the residuals in Fig. 12b, confirming Eq. 89. Third- and higher-order dependencies can be seen in the plot of residuals against predicted responses if the current fit is sufficiently poor. Such cases occur when the data set that would correspond better to the current model is sufficiently different from the current data set that third- and higher-order terms in the Taylor series represented in Eq. 87 are not negligible. In any case, curvature of any kind in the plot of residuals against predicted response is an indicator of lack-of-fit.

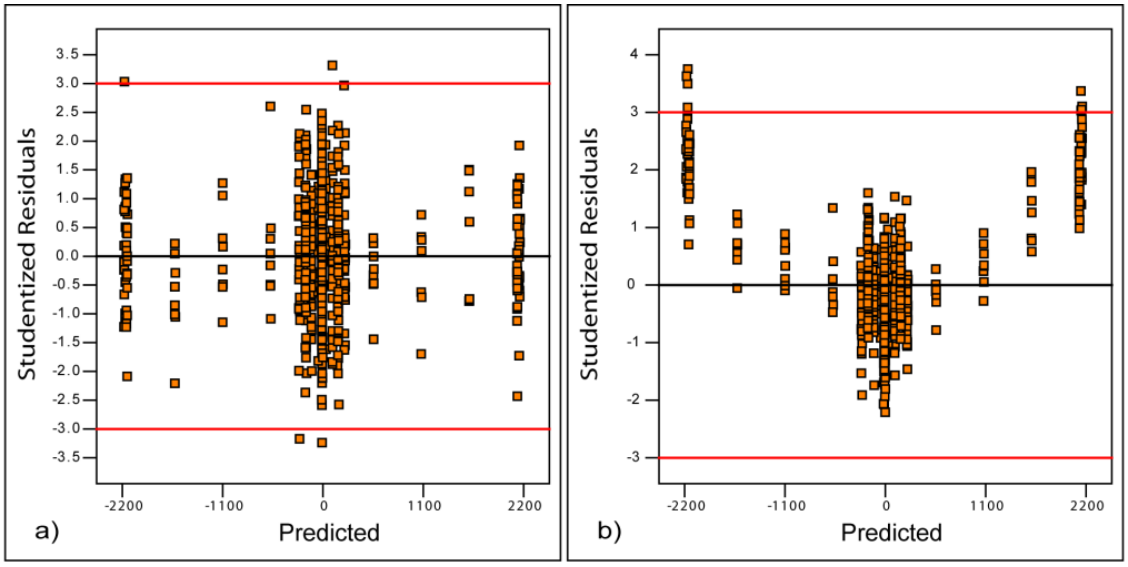


Figure 12. Residuals vs. predicted responses in (a) a quadratic model, and (b) a factor interaction model.

Residuals are also commonly plotted as a function of data acquisition time. Figure 13 presents two examples, where the run number serves as a surrogate for time. There is no trend evident in Fig. 13a, while there is a trend in the data in Fig. 13b. If the residuals are not independent of time as in Fig. 13b, it suggests that changes were occurring during the experiment that affected the response measurements. These changes can be caused by temperature effects, instrument drift, operator fatigue, or any number of other influences.

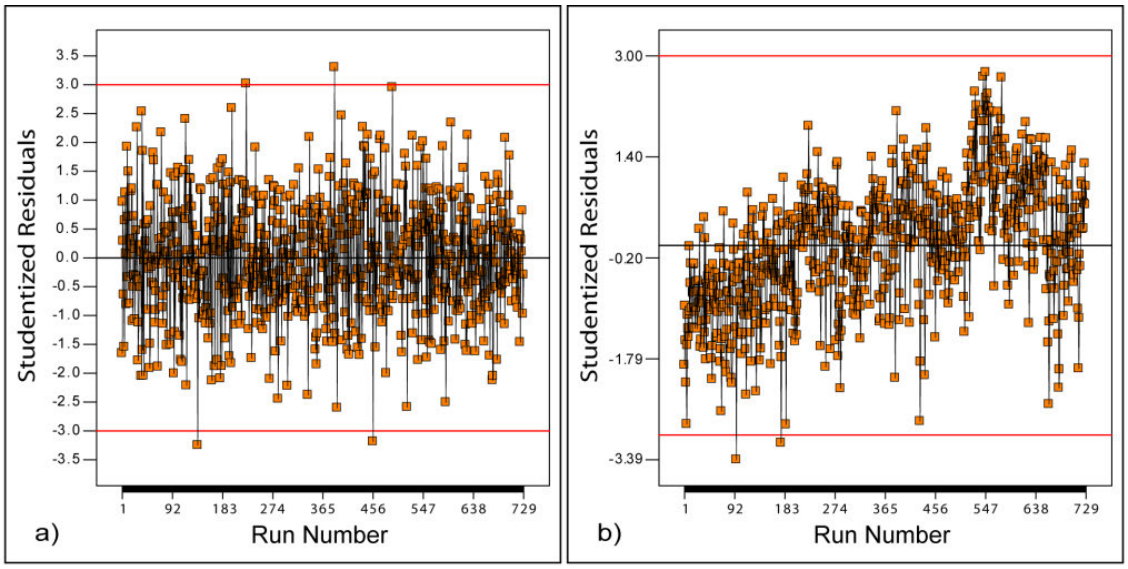


Figure 13. Residual time histories of (a) no time-varying bias, and (b) bias errors increasing with time.

Time-varying bias errors of the kind that Fig. 13b reveals are especially troublesome in experiments in which independent variables are changed systematically over time. In a wind tunnel test, for example, it is common to change angle of attack monotonically—from some lower value to some higher value—in prescribed increments. If a time-varying bias error is in play, it is not possible to isolate genuine angle of attack effects because they are confounded by the time-varying bias error. This is because forces and moments cannot be plotted against angle of attack alone; rather, such plots show how the forces and moments change as a function of changes in angle of attack, plus whatever is responsible for the time-varying bias error.

Unfortunately, if systematic errors are present, the errors associated with successive measurements in a time series are not independent. In other words, if the i^{th} measurement is too high, the $(i + 1)^{st}$ measurement is more likely to be too high than too low. This loss of independence violates one of the fundamental assumptions upon which regression analysis is based, bringing into question the validity of any response model fitted to data acquired under such circumstances. The solution is to ensure that the independent variable levels are not changed systematically with time, but are set in random order. This is a common quality assurance tactic of the modern design of experiments (MDOE). Sometimes the independent variables are hard to randomize, for which case there is a whole class of specialty experiment designs called split-plot designs. The reader is encouraged to consult the literature for more information on this class of experiments.

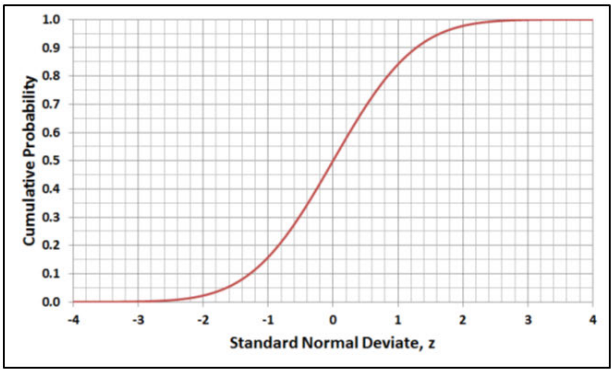
Residuals are commonly plotted against factor levels as well as response prediction levels. Any systematic variation in such plots indicates that the factor under consideration is misrepresented in some way in the response model. For example, if the residuals increase in magnitude with one of the independent variables, it is an indicator that the response has a different dependence on that factor than is represented in the model.

Patterns considered thus far are revealed when residuals are plotted against some physical quantity: a dependent variable, an independent variable, or time. In general, it is useful to plot residuals against anything if subject matter expertise suggests it might shed light on the efficacy of the response model. We now consider a somewhat different variety of residual plot, which is designed to reveal information about the distributional properties of the residuals. This is important for two reasons. First, while regression methods are robust with respect to the assumptions upon which they are based, the best fits generally occur when experimental errors are normally distributed, at least approximately. Secondly, statements about the uncertainty of experimental results can be most easily made when residuals are normally distributed. It is only then, for example, that $\pm 2\sigma$ represents a 95% confidence interval. A test for normality using normal probability plots is described in the next section.

6.2. Probability Plots

Points distributed normally about some reference will fall along what is nominally a straight line in a normal probability plot. Systematic departures of the points from a straight line indicate that they are not distributed normally. If these points are residuals, this would suggest lack-of-fit errors due to an inadequate response model that fails to fare through the heart of the data. On the other hand, if the residuals are distributed normally, we can estimate the width of certain useful precision intervals used to specify uncertainty from a quantitative estimate of the variance. For example, the 95% confidence interval half-width is roughly twice the square root of the variance (twice the standard error, or “two sigma”), if the distribution is normal. If it is not normal, even if the variance estimate is valid, the physical interpretation of the corresponding uncertainty can be less clear.

There are different probability plots for different types of distributions, but they all entail a graphical comparison between theoretical and observed distributions to assess qualitatively how well they match each other. To illustrate how such a plot is constructed, consider a sample of data drawn from a distribution known to be normal.



Point (i)	f_i	Expected Z	Sorted Normal Residuals
1	0.0200	-2.054	-2.212
2	0.0600	-1.555	-2.031
3	0.1000	-1.282	-1.835
4	0.1400	-1.080	-1.621
5	0.1800	-0.915	-1.508
6	0.2200	-0.772	-1.277
7	0.2600	-0.643	-0.786
8	0.3000	-0.524	-0.636
9	0.3400	-0.412	-0.616
10	0.3800	-0.305	-0.574
11	0.4200	-0.202	-0.403
12	0.4600	-0.100	-0.334
13	0.5000	0.000	-0.215
14	0.5400	0.100	-0.189
15	0.5800	0.202	0.032
16	0.6200	0.305	0.154
17	0.6600	0.412	0.292
18	0.7000	0.524	0.325
19	0.7400	0.643	0.484
20	0.7800	0.772	0.499
21	0.8200	0.915	0.793
22	0.8600	1.080	0.817
23	0.9000	1.282	1.547
24	0.9400	1.555	1.759
25	0.9800	2.054	2.123

Figure 14. The random, normally distributed data in the table on the right were generated using the normal cumulative probability density function on the left.

The column on the right in the table of Fig. 14 displays 25 numbers drawn at random from a normal distribution with a mean of zero and a standard deviation of one. These numbers were generated using a random number generator in conjunction with the cumulative normal distribution function on the left of Fig. 14. Ordinates of the function in Fig. 14 *(y)* were selected with a random number generator. The corresponding abscissa *(x)* represents a number in standard deviations from the mean, drawn at random from a normal distribution. These numbers could be expressed in physical units by multiplying them by the standard deviation of the sample in physical units, but for this illustration, the numbers are represented simply as multiples of such a standard deviation. They can be said to represent the residuals of a regression analysis in which the response model displays no lack-of-fit, so that the residuals have no systematic component and are entirely random.

The first step in constructing a normal probability plot is to sort the points in the sample from smallest to largest, as in the rightmost column of the table in Fig. 14. An index number, i, is assigned to each data point in ascending order as in the first column of the table in Fig. 14.

The column labeled f_i is computed in this example using the formula $f_i = (i - ½)/n$, where i is the index, and n=25 is the sample size. This function maps the n points into uniformly spaced locations on the vertical axis of the normal cumulative probability distribution function. The column labeled "Expected Z" is computed from f_i as the abscissas of each of these uniformly spaced ordinates. The residuals of a regression analysis, expressed as multiples of the standard deviation of the residuals (rightmost column in Fig. 14), will be proportional to the expected Z-values if the distribution of the residuals around the regression model is perfectly normal. Plotting column 3 against column 4 in Fig. 14 will yield a nominally straight line in that case.

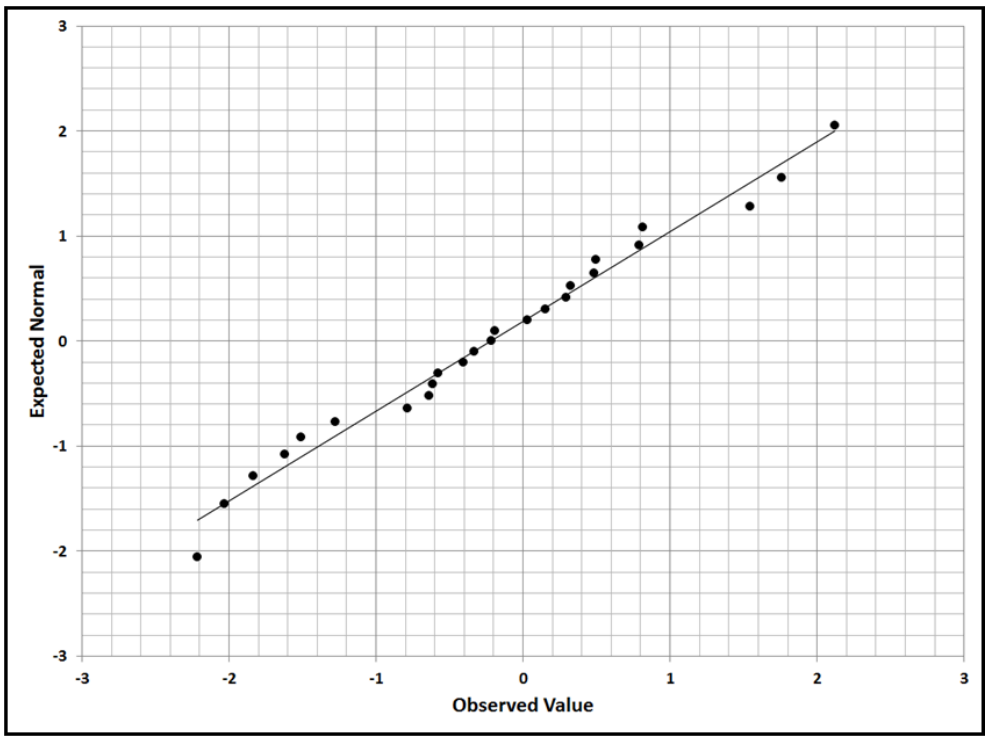


Figure 15. Normal probability plot for 25 points drawn at random from a probability distribution known to be normal.

Figure 15 displays the data in column 3 of the table in Fig. 14, plotted against the data in column 4. The points fall nominally along a straight line, suggesting that the residuals of this example are in fact normally distributed (as they are known to be). The fact that the points do not all fall precisely on a perfectly straight line is due simply to the finite size of the data sample. All 25 points were drawn from a computer-generated population known to be normally distributed with a mean of zero and a standard deviation of one, but a finite number of individual points will each have been drawn from a population with a slightly different mean and standard deviation except in instances of sheer coincidence. A colloquial criterion for normality, attributed to Douglas Montgomery of Arizona State University, is that all the points should pass "the fat pencil test." That is to say, if all the points in a normal probability plot are indeed drawn from a normal distribution, it should be possible to cover them with a fat pencil. The points in Fig. 15 satisfy this criterion.

Figure 15 is an artificial example of a normal probability plot, manufactured simply to illustrate the method of construction. Figure 16 is an example of a probability plot developed from actual experimental data. These data were acquired during computational experiments conducted as part of the Space Shuttle Return to Flight program, which was described earlier in this chapter.

For each of 33 different body zones, polynomial models were developed to predict such responses as tile bond-line temperature, structural temperature, tile factor of safety, and structural margin of safety. Models were also developed to predict tile out-of-plane deflections for two reentry trajectory regimes: Mach 25 to Mach 4, and the Terminal Area Energy Management (TAEM) regime from Mach 4 to touchdown. Each response was modeled as a function of the dimensions of dents in the Thermal Protection System caused by the impact of ice and foam debris falling off of the external tank on ascent. The dimensions that were modeled were length,

width, depth, and the angles of the entrance, side, and exit faces. Various shuttle state variables were also identified as independent variables for these response surface models.

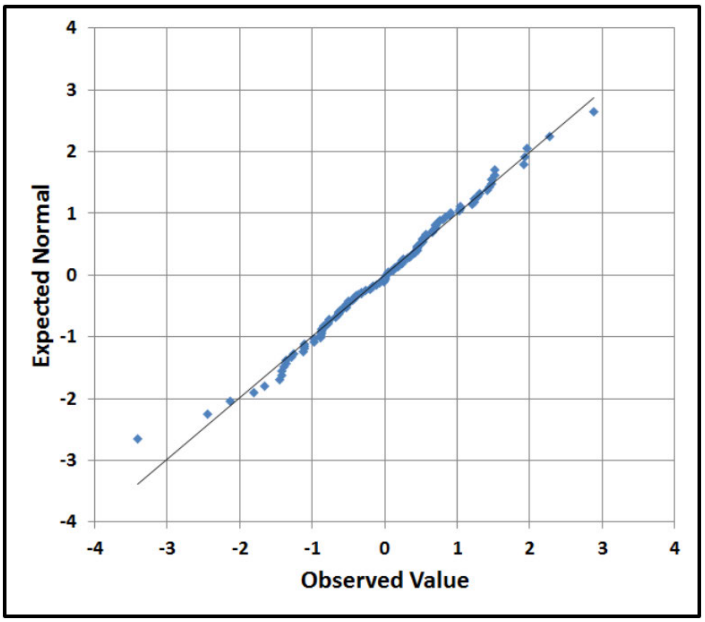


Figure 16. Representative normal probability plot for shuttle structural temperature response surface model.

Figure 16 displays the 124 residuals of a response surface model representing structural temperature as a function of divot geometry and shuttle reentry state variables at a particular location on the windward side of the vehicle. As with the example that produced Fig. 15, the residuals lie along a nominally straight line, indicating that they are normally distributed. We conclude, therefore, that the unexplained variance in this sample of data behaves as if it is due primarily to ordinary random error rather than systematic bias error, indicating that there is no significant lack-of-fit in the polynomial response model for structural temperature. This means structural temperature predictions made with the polynomial response model approximated those made with the full underlying code, as intended.

Figure 17 displays normal probability plots for two additional regression models developed during the Shuttle Return to Flight program. A total of 198 such models were assessed, six in each of 33 body point zones. The residuals for all of these models fell on straight lines in normal probability plots, suggesting that they were normally distributed about the regression models developed as surrogates for the more complex underlying codes.

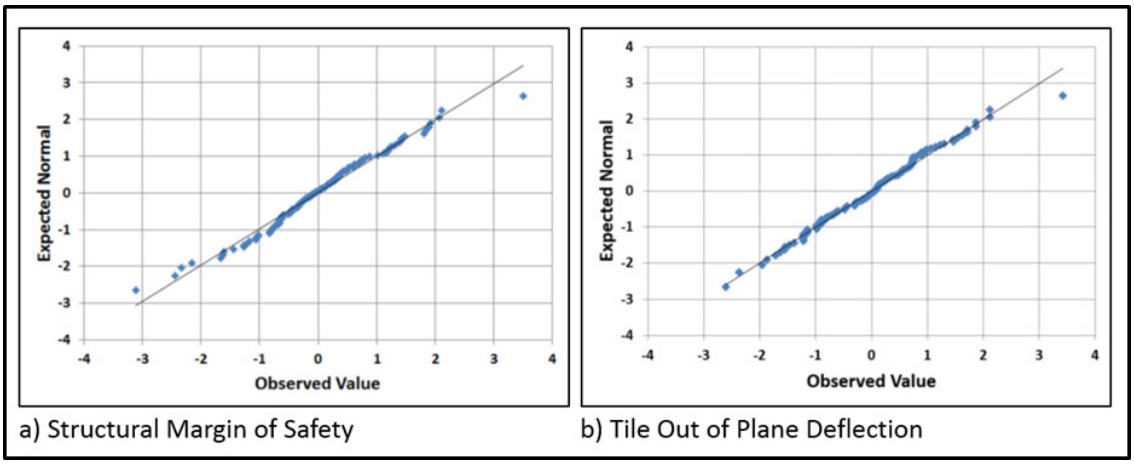


Figure 17. Normal probability plots used to assess the fits of two representative Shuttle Return-to-Flight regression models: (a) structural margin of safety, (b) tile out-of-plane deflection.

Figure 18 displays normal probability plots of residuals for a practical application of response surface modeling touched upon earlier, in the discussion of residual patterns. In this application, four different candidate models were fitted to data acquired in a six-component force balance calibration experiment. A force balance is designed to be a linear transducer, so in principle, a first-order response model would adequately predict electrical outputs from the applied aerodynamic loads in a wind tunnel test. In practice, there are interactions among the load channels; for example, the electrical output caused by a given normal force might depend on the level of axial force that is simultaneously applied. Likewise, the individual load channels are not perfectly linear; there can be some slight curvature in the plot of a given electrical output against its corresponding load. For these reasons, force balance calibration data are commonly fitted to a second-order polynomial calibration equation.

A full second-order response model in all six-component loads has 28 terms. For the specific example considered here, the term reduction procedures described earlier identified 11 of these terms as insignificant, resulting in a 17-term reduced model.

Figure 18a is the normal probability plot of residuals from the 17-term reduced model. The straight line suggests that the residuals are normally distributed, which further suggests that the fit is adequate, as the model holds up through the data in such a way that any differences between the model predictions and the measured data are due only to random experimental error that we expect to be normally distributed by the Central Limit Theorem, and not due to any systematic bias error caused by the lack-of-fit of an unsatisfactory model.

Compare Fig. 18a with Fig. 18b, the normal probability plot of the full 28-term second-order model. This comparison indicates that no lack-of-fit was introduced by significantly reducing the number of model terms, so model accuracy was not compromised in the term reduction process. At the same time, reducing the number of terms in the model from 28 to 17 means the average prediction variance was reduced by over 39%, per Eq. 53. This corresponds to a reduction in the standard error of prediction (square root of the prediction variance) of over 60%, a substantial increase in precision.

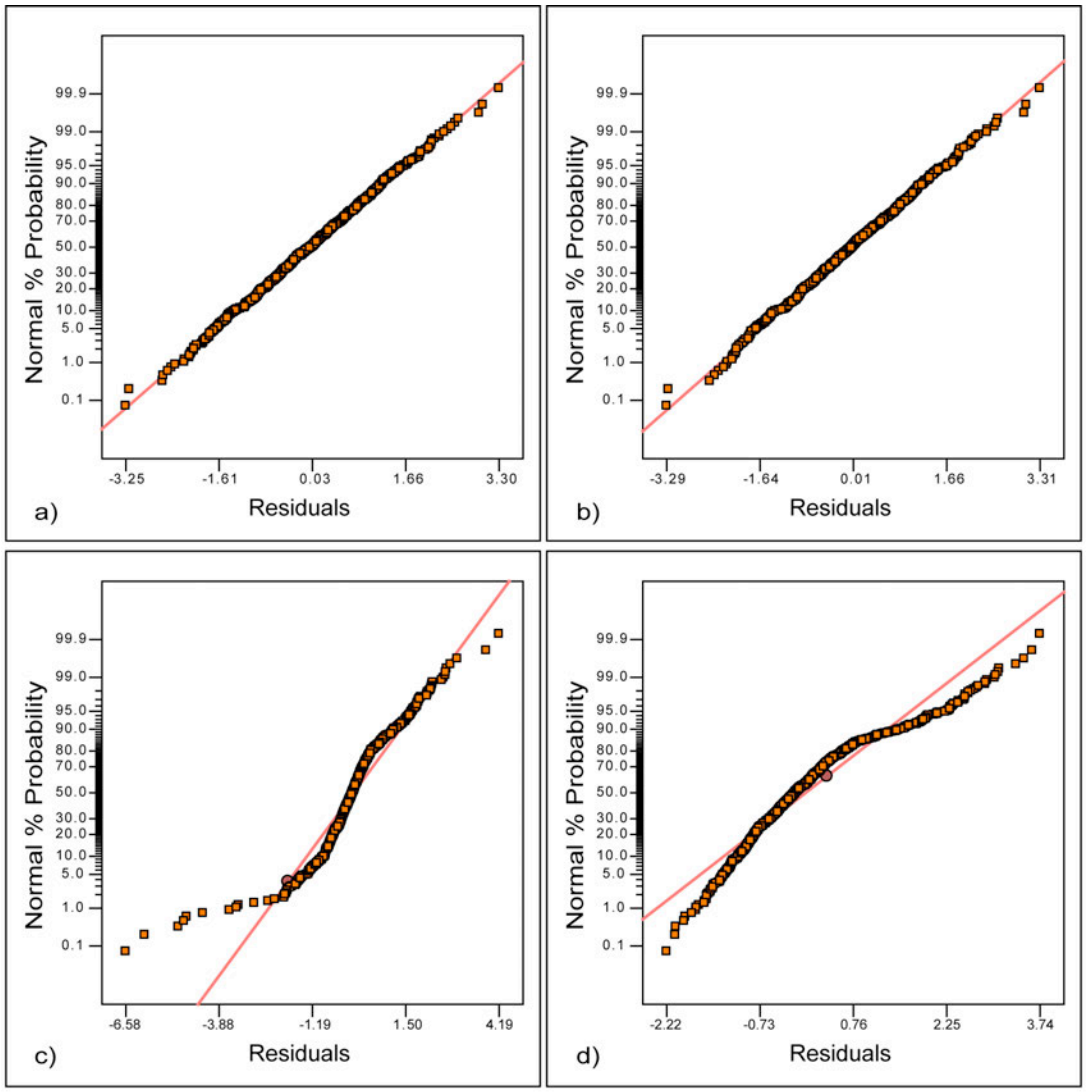


Figure 18. Normal probability plot of normal force residuals: (a) a17-term reduced model, (b) a full 28-term second-order model, (c) a 6-term first-order model, and (d) a 22-term factor interaction model.

The residuals from the 17-term reduced model used to generate the normal probability plot in Fig. 18a are the same residuals that are plotted against response predictions in Fig. 12a. They show no particular dependence on predicted response, suggesting that this model fits the data.

Figures 18c and 18d show how the normal plot of residuals would look if an inadequate model were fitted. Figure 18c displays residuals from a first-order model featuring only the intercept and the six first-order load terms. As noted, one might have thought that such a model would fit the data reasonably well, given that the force balance is explicitly designed to be a linear transducer. Figure 18c illustrates the importance of accounting for the small interaction and curvature terms of a second-order calibration model.

Figure 18d further illustrates the importance of the second-order terms. Here, residuals of a two-factor interaction model are plotted. A two-factor interaction model consists of the full second-order model, absent only the six pure quadratic terms. That is to say, terms of the form A^2 are

excluded from the model, with only first-order terms and second-order terms of the form AB (interaction terms) retained. In spite of how small the curvature terms are for a transducer designed not to have them, there is a substantial systematic departure from a straight-line plot of the residuals, indicating lack-of-fit induced by the absence of the quadratic terms. Residuals used to generate the normal probability plot in Fig. 18d were also used in the plot of residuals against predicted response displayed in Fig. 18d.

6.3. The Critical Binomial Number Test

This section will describe a test of model adequacy that is somewhat more quantitative than the residual plotting approach of the previous subsection. It assumes that model predictions have been made for a number of validation sites randomly distributed throughout the design space, where data have been acquired but not used to fit the response model that is being evaluated. The residuals from these validation sites are regarded as Bernoulli trials with the customary pass/fail binary outcome possibilities. If the model predicts adequately at a given validation site it is said to have passed that Bernoulli test, or else it is said to have failed it. Material in this section draws heavily from work by DeLoach [26].

We begin by defining an "adequate prediction" as one that results in a residual that is within some prescribed tolerance. As we will see, this seemingly innocuous definition of prediction adequacy features an assumption that is not true in the presence of inference error risk. Ironically, the better the model fits the data, the less likely this assumption is to be true. We will address this assumption, but will begin by neglecting inference error risk.

Recall the following definition of inference error. At each validation site we infer that the model predicts adequately or that it does not. In either case, we may be right or we may be wrong. If we are right, we will have made a proper inference whether the model predicts adequately or not. If we are wrong, we will have made an inference error.

Occasionally, random experimental error can be large enough to cause an inference that the model predicts inadequately at a given design space site even when it actually predicts adequately there. This happens when random error artificially inflates a residual enough that it is out of tolerance. Recall that we refer to such an inference error as a Type I (alpha) inference error. The consequences of such an error generally take the form of wasted resources consumed in an unnecessary attempt to improve the model.

Random experimental error can also interact with a genuine systematic lack-of-fit bias error to cause us to infer that the model predicts adequately at a given design space site where in fact it does not. This happens when a systematic bias error and an ordinary random experimental error combine to produce a net residual that is small enough to be within tolerance. We refer to such an inference error as a Type II (beta) inference error. This type of inference error can result in the validation of an inadequate response model, and is therefore especially worrisome.

There is a discussion earlier in this chapter of how to scale the experiment in order to drive the probabilities of both types of inference error below prescribed maximum acceptable levels. While inference error probabilities can be driven arbitrarily low by acquiring enough data, they can never be driven to zero with a finite volume of data. Non-zero inference error probabilities mean the one can never entirely rely upon the assumption that an out-of-tolerance residual is a reliable indicator of an inadequate response prediction, and in fact this assumption is wrong a surprising large percentage of the time, as will be explained further below.

Let us assume that data have been acquired at S validation sites, and that there is an expected per-trial success probability of p_s at each of these sites. The probability of failure is p_f, which is of course, $1 - p_s$. We expect the success probability to be no less than $1 - \alpha$, where α is the maximum Type I inference error risk for which the experiment has been scaled; for example, 0.05. That is to say, we expect $(1 - \alpha) \times 100\%$ of the validation-site residuals to be within tolerance in the absence of significant lack-of-fit, with no more than $\alpha \times 100\%$ of them out of tolerance due to ordinary random experimental error. A significantly higher failure rate would be interpreted as evidence of the kind of systematic lack-of-fit bias errors that characterize an inadequate model.

Figure 19 displays the binomial distribution for 100 trials with a per-trial success probability of 0.95. The Critical Binomial Number (CBN) for a significance of 0.01 is 89. This means that if the per-trial success probability is in fact 95%, there is less than a 1% chance that there will be fewer than 89 successes in 100 trials. If there are 89 or fewer successes, we can infer that the per-trial success probability is less than 95%, with no more than a 1% chance of an inference error.

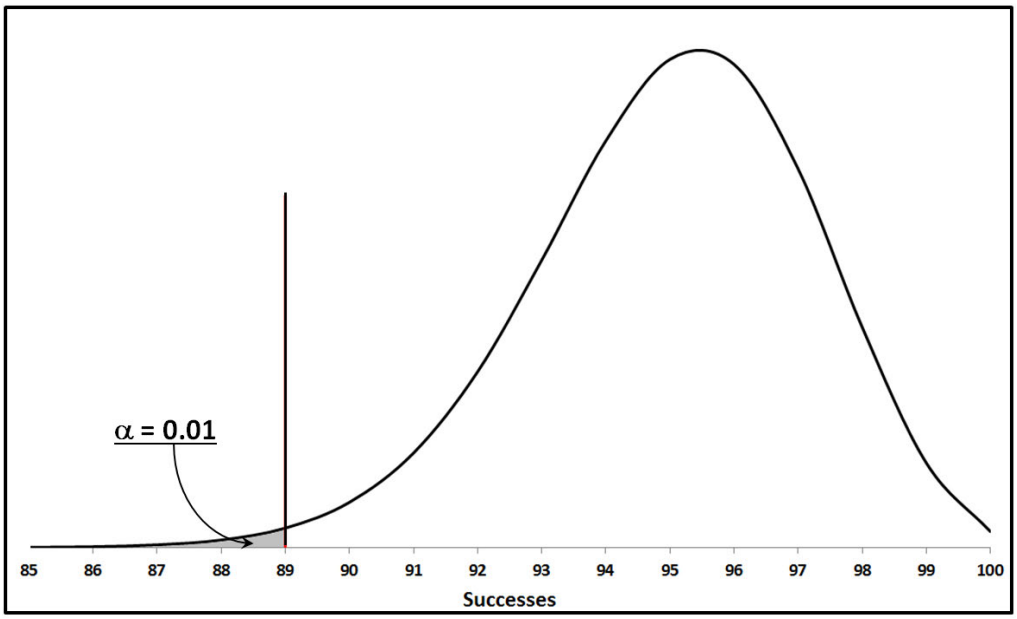


Figure 19. Binomial distribution for 100 trials with $p_s = 0.95$. The CBN for $\alpha = 0.01$ is 89.

The test of model adequacy simply requires that the number of successful Bernoulli trials exceed the CBN. This appears to have a 99% chance of occurring in this example if the per-trial success probability is in fact 95% as we expect; that is to say, if there actually is a 95% probability that a measurement made at a given site will lie within the model's 95% prediction interval for that site.

The apparent ease with which an adequate model ought to be able to pass this test notwithstanding, experience has shown that this happens much less often than expected, even for models that pass all other tests of model adequacy and give every indication of being perfectly valid response models. This apparent mystery can be explained by what is known in forensic science as the Prosecutor's Fallacy. The prosecutor commits this fallacy when he demonstrates that there is a high probability that the defendant's fingerprints will be on the murder weapon if he is guilty, and then introduces evidence to show that his fingerprints are in fact on the weapon. The fallacy is in assuming there is relevance in the high probability that the defendant's fingerprints would be on the weapon given that he is guilty. To obtain a legitimate guilty verdict,

the prosecutor must show a high probability of guilt given that the defendant's fingerprints are on the weapon, not the reverse of this. There may be some other explanation for the fingerprints—perhaps the defendant handled the weapon before or after the crime—or reliable witnesses may place him somewhere else at the time of the crime. The conditional probability of A given B, $P(A|B)$, is not generally the same as the conditional probability of B given A, $P(B|A)$, and to assume so is to commit the Prosecutor's Fallacy.

We commit the Prosecutor's Fallacy during the CBN test (or any time we assess model prediction adequacy by counting out-of-tolerance residuals at validation sites) when we note the high probability that a residual will be out of tolerance given that the model prediction is biased, then note that a residual is out of tolerance at a given validation site, and then conclude that the model fits poorly there. We are not interested in the high probability that a residual will be out of tolerance at some site in the design space given that the model is biased there. We want to know the probability that the model prediction is biased at some site given that the residual is out of tolerance there.

Consider a good model for which there is only a small probability, ε, that predictions are biased at any given site in the design space. The model predicts adequately *at* $(1 - \varepsilon) \times 100\%$ of the sites. Random error is expected to cause $\alpha \times 100\%$ of the residuals to be out of tolerance even where the model is unbiased, so $\alpha \times (1 - \varepsilon) \times 100\%$ of the validation sites will feature residuals that are out of tolerance only because of random experimental error, and not because of any response model bias. Let us further assume that the experiment has been scaled to ensure that there is only a small probability, β, of erroneously validating a biased model, so there is a probability of $(1 - \beta) \times 100\%$ that a residual will be out of tolerance where the model truly is biased. This means that $\varepsilon \times (1 - \beta) \times 100\%$ of the validation sites will feature residuals that are out of tolerance because the model is in fact biased.

The residual at a randomly selected validation site can thus be out of tolerance whether the model is biased or not. The probability of an out of tolerance residual is $\varepsilon(1 - \beta)$ when the model is biased and $\alpha(1 - \varepsilon)$ when the model is unbiased, so there is a total probability of $\varepsilon(1 - \beta) + \alpha(1 - \varepsilon)$ that a residual will be out of tolerance either due to a bias error or due simply to random error. It is only in the former case that the model can be said to predict inadequately. A residual that is out of tolerance simply due to random experimental error does not reflect negatively on the model.

Let A represent the case in which the model is biased, and let B represent the case in which the residual is out of tolerance. Then $P(A|B)$ is the conditional probability that the model is biased, given that the residual is out of tolerance:

$$P(A \mid B) = \frac{\varepsilon(1-\beta)}{\alpha(1-\varepsilon) + \varepsilon(1-\beta)} \tag{90}$$

Expanding and rearranging terms:

$$P(A \mid B) = \frac{\varepsilon - \varepsilon\beta}{\alpha - \alpha\varepsilon + \varepsilon - \varepsilon\beta} = \frac{\varepsilon - \varepsilon\beta}{\alpha + \varepsilon - \varepsilon(\alpha + \beta)} \tag{91}$$

Assuming that the model has been fitted to a data sample scaled to ensure small inference errors, and further assuming that the residuals provide evidence of a generally adequate model (so small

α, β, and ε), then we can neglect second-order terms so that Eq. 91 becomes, to a good approximation:

$$P(A \mid B) \approx \frac{\varepsilon}{\alpha + \varepsilon} \tag{92}$$

The effect of a non-zero Type I inference error probability, α, is to ensure that $P(A|B)$ is not 100% as one might expect it to be. Alpha is not zero because even an adequate model is subject to random prediction error, with some non-zero probability that this error will exceed a specified tolerance at a given site in the design space. It is for this reason that an out-of-tolerance residual is not a sure indicator that the model predicts inadequately at a given site; the residual could be out of tolerance simply because of random prediction error instead. There is only one way for the model to be biased (an inadequate model formulation, with probability ε that this occurs at a randomly selected site in the design space), but there are two ways for the residual to be out of tolerance (random error with probability α, and a systematic lack-of-fit bias error due to inadequate model formulation with probability ε).

Even if a residual is out of tolerance at some site in the design space, there is some probability that the model still predicts adequately there, and that the residual is only out of tolerance because of random prediction error and not because of any defect in the model. Figure 20 plots that probability against the fraction of the design space where predictions are unbiased. The quality of the model therefore increases from left to right in Fig. 20.

On the right side, where the model is unbiased at every site in the design space, there is a 100% probability that the model is actually unbiased even though the residual is out of tolerance. In that case, the out-of-tolerance residuals can be entirely attributed to experimental error at all sites, with none that can be attributed to imperfections in the model.

On the left side of Fig. 20 where the model is significantly biased at every site in the design space, there is very little probability that an out of tolerance residual is due to anything but lack-of-fit error. The probability of a false indictment of the model by an out-of-tolerance residual transitions from zero to 100% according to the quality of the model, per Fig. 20.

The curves in Fig. 20 correspond to three different Type I inference error probabilities. A Type II inference error probability of $\beta = 0.01$ is assumed in all three cases, but the probability of a false indictment of the response model at a randomly selected design-space site is largely independent of β for small β. See Eqns. 91 and 92.

The vertical line at $1 - \varepsilon = 0.95$ in Fig. 20 cuts through the three curves at a point representative of a model that is likely to be regarded as adequate, given a 95% probability that it produces unbiased response estimates for any randomly selected site in the design space. This indicates that the probability of a model prediction being unbiased at some site where the residual is out of tolerance increases with α. It illustrates that the greater the probability of significant random error, the greater the probability that an out-of-tolerance residual is an unreliable indicator of model inadequacy. This is especially so when model predictions are unbiased at a high percentage of the design space sites.

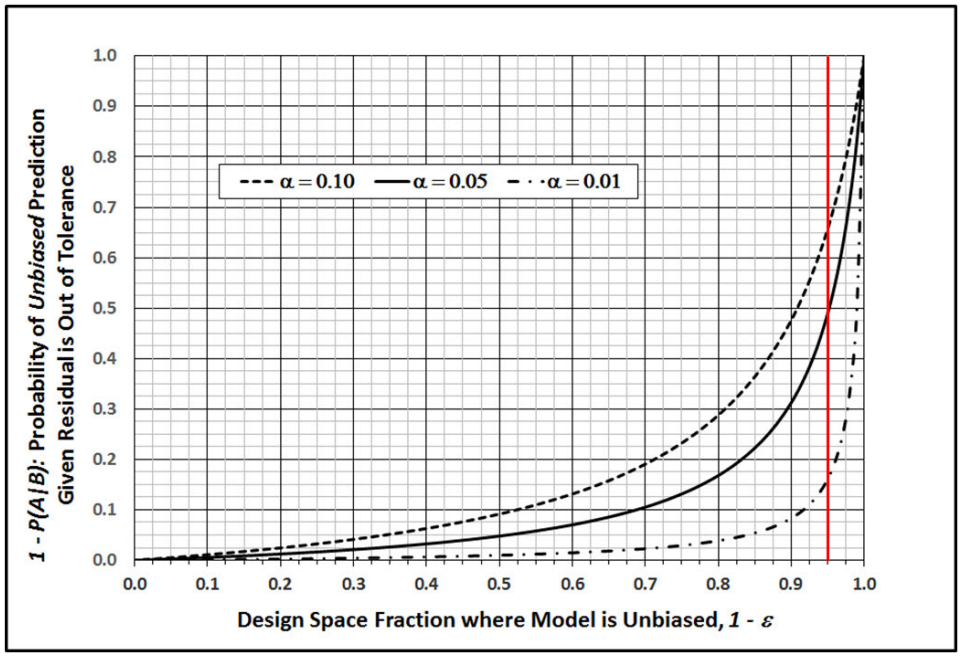


Figure 20. Probability that a response model prediction is actually unbiased at a given design space site, even though the residual is out of tolerance at that site.

6.4. A Tolerance Metric for Response Model Adequacy Assessment

It has been demonstrated above that the probability of a residual being out of tolerance, either due to random experimental error or due to systematic lack-of-fit bias error, is $\varepsilon(1-\beta)+\alpha(1-\varepsilon)$. We therefore expect the same fraction of randomly selected validation sites to generate residuals that are out of tolerance when the model is assessed by comparing model predictions to independent estimates of the true response. Assume that such comparisons have been made at a number of independent validation sites, and let P_f represent the percentage of failed validation trials. That is to say, let P_f be the fraction of validation sites where residuals are out of tolerance. Then,

$$P_f = \varepsilon(1-\beta)+\alpha(1-\varepsilon) \tag{93}$$

Solving for ε:

$$\varepsilon = \frac{P_f - \alpha}{1-\alpha-\beta} \tag{94}$$

Equation 94 indicates that the fraction of the design space in which the response model actually is biased, ε, is the fraction of the design space, P_f, in which the response model only appears to be biased (due to out-of-tolerance residuals), corrected for Type I and Type II inference error probabilities α and β respectively. It is only when both α and β are zero that $\varepsilon = P_f$.

For the purpose of quantifying the suitability of a response model, it may make more sense to quantify the compliment of ε; namely, $1-\varepsilon$, as follows:

$$\varepsilon = \frac{P_f - \alpha}{1 - \alpha - \beta} \rightarrow 1 - \varepsilon = 1 - \frac{P_f - \alpha}{1 - \alpha - \beta} = \frac{1 - \alpha - \beta - \left(P_f - \alpha\right)}{1 - \alpha - \beta} = \frac{\left(1 - P_f\right) - \beta}{1 - \alpha - \beta} \tag{95}$$

Let $P_s = 1 - P_f$: represent to percentage of successes in a series of Bernoulli validation trials. Then Eq. 95 becomes:

$$1 - \varepsilon = \frac{P_s - \beta}{1 - \alpha - \beta} \tag{96}$$

The quantity $1 - \varepsilon$ in Eq. 96 is an estimate of the fraction of the design space in which the response model exhibits no significant lack-of-fit bias. It is P_s, corrected for Type I and Type II inference error probabilities α and β respectively, which are known because they were specified as a prerequisite for scaling the experiment. We compare $1 - \varepsilon$ with some specified minimum standard to assess whether the response model meets quality specifications. For example, assume a model that predicts without significant bias error in 95% of the design space is to be regarded as adequate. Assume further that a sufficient volume of data has been acquired to ensure maximum Type I and Type II inference error probabilities of $\alpha = 0.05$ and $\beta = 0.01$, respectively, and that only P_s= 92% of the validation-site residuals are within tolerance. Absent any correction for inference error, it would appear as if the model is inadequate, since only 92% of the residuals are within tolerance, which is less than the 95% criterion. However, use the above equation to correct P_s for α and β:

$$1 - \varepsilon = \frac{P_s - \beta}{1 - \alpha - \beta} = \frac{0.92 - 0.01}{1 - 0.05 - 0.01} = 96.8\% \tag{97}$$

We see that an original estimate suggesting that the response model is unbiased over only 92% of the design space, which is unacceptable, has been revised to an acceptable value of 96.8% by taking inference error risk into account.

7. Concluding Remarks

Experimentation has been described as a "controlled experience" in which factors that influence the behavior of a system of interest are varied in order to observe how the system responds. Conventional aerospace experimentation, especially experimental aeronautics, relies on high-volume data collection to quantify how various system responses change for as many combinations of factor changes as limited resources permit. The chief product of a conventional experiment is generally data, and productivity is construed to mean generating as much of that product as possible within the test budget.

Quality in a conventional experiment refers to quality of the principal product: data. A high-quality conventional experiment features high quality data, by which is meant data with relatively little unexplained variance. Most of the variance in a high-quality, conventional experiment is intentionally induced by factor changes executed during the normal course of the experiment. Unexplained variations such as random error are small by comparison. Quality assurance in conventional experimentation is focused on perfecting the measurement environment by exorcizing as many sources of unexplained variance as possible and striving for a state of statistical control, in which all unexplained variance occurs randomly about mean values that are time-independent. Considerable attention is paid to executing all experimental procedures

according to fixed protocols and to frequent within-test calibration of instrument systems to stabilize bias errors.

Productivity in a conventional experiment generally requires that data acquisition speed be maximized in order to ensure the largest volume of acquired data in the fixed interval of the test. This necessary focus on speed forecloses certain quality assurance and quality assessment tactics that might otherwise reduce the uncertainty in system response estimates, especially in physical experiments. In any case, even an experiment that is only moderately complex will feature more potential factor combinations than typical resource constraints will permit. This results in the need for prioritization and the inevitable loss of information that there is no time to gather, with the attendant dubious rationalization that such missed data is likely to have been relatively uninteresting anyway.

Formal experiment design differs from conventional experimentation principally in what is perceived to be the product of an empirical investigation. The product is "data" in a conventional test ("good quality" data and lots of it), but in a designed experiment the product is "knowledge." The objective of a designed experiment is to learn enough about the system under study that its future behavior can be adequately predicted for any combination of factor levels within the range they were varied in the experiment. Here, "adequately" means that such predictions will occur within a prescribed tolerance with a specified minimum frequency.

This change from data to knowledge as the product of an experiment has profound implications for both quality and productivity. A high-quality designed experiment is one that "gets the right answer," independent of the nature of the measurement environment. Poor quality measurement environments can be made to yield high quality experimental results by implementing various quality assurance tactics and by acquiring enough data.

A productive designed experiment minimizes data volume to the extent possible rather than maximizing it. The essence of productivity in a designed experiment is to get in and out of the test facility as quickly as possible, with the smallest expenditure of test resources.

A data sample that is of high quality by the standards of conventional experimentation cannot rescue a result that produces an improper inference about the behavior of the system under study, nor will it matter if record volumes of data were acquired in the test. High quality data acquisition is a legitimate element of the approach to achieving the objective of a well-designed experiment, but it cannot be the objective itself.

The formal design of an experiment progresses through the stages that have been described in this chapter. The objective is quantitatively formulated in an initial design phase by defining a design space over which results are to apply, and by specifying tolerance levels and acceptable probabilities of committing the two types of possible inference error associated with a designed experiment: the validation of an inadequate response model or the indictment of an adequate one. A candidate response model is proposed that typically consists of a low-order Taylor-series approximation of the unknown underlying response function, with the order of the series informed by experience and subject matter expertise. The experiment is then scaled using this information, meaning that the minimum volume of data to be acquired is estimated. Sufficient data are specified to fit the model, with ample residual degrees of freedom to ensure specified prediction uncertainty limits and to test for lack-of-fit.

Data are acquired in the execution phase of the experiment, and in the analysis phase those data are fitted to the candidate model. The model is then assessed by various tests applied to the

residuals formed by subtracting independent estimates of the true response from predictions made using the candidate response model.

Models that are judged to be biased by systematic lack-of-fit errors are rejected or modified, and additional data needed to fit the revised model are acquired if necessary. The end result is expected to be a model that can be used to adequately predict responses for all sites of interest in the design space, including all sites that are precluded from being estimated by the resource constraints of a conventional experiment. The uncertainty of each response can be estimated, and the cost of acquiring enough data to produce such a model is considerably less than the cost of directly generating the large databases that often characterize conventional aerospace experiments, especially in experimental aeronautics.

References

1. Hotelling, H., "Some problems in weighing and other experimental techniques," *Annals of Mathematical Statistics*, Vol. 15, No. 3, pp. 297-306, 1944.
2. Daniel, C., *Applications of Statistics to Industrial Experimentation*, Wiley, New York, 1976.
3. Coleman, H. W., and Steele, W. G., *Experimentation and Uncertainty Analysis for Engineers,* New York: Wiley, 1989.
4. Bevington, P.R., and Robinson, D.K., *Data Reduction and Error Analysis for the Physical Sciences*, 2nd ed, New York: McGraw-Hill, 1992.
5. DeLoach, R., and Ulbrich, N., "A Comparison of Two Balance Calibration Model Building Methods (Invited)," 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, paper AIAA 2007-0147, Jan 8-11, 2007.
6. Morelli, E. A., and DeLoach, R., "Response Surface Modeling Using Multivariate Orthogonal Functions (Invited)", 39th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, paper AIAA 2001-0168, January 2001.
7. IEEE Standard for Binary Floating-Point Arithmetic, ANSI/IEEE Std 754-1985, August 12, 1985.
8. DeLoach, R., "Impact of Loading Selection and Sequencing on a Force Balance Calibration (Invited)," 25th Aerodynamic Measurement Technology and Ground Testing Conference, San Francisco, CA, paper AIAA 2006-3436, June 2006.
9. DeLoach, R., "Tailoring Wind Tunnel Data Volume Requirements through the Formal Design of Experiments," 20th AIAA Advanced Measurement and Ground Testing Technology Conference, Albuquerque, NM, paper AIAA 98-2884, June 1998.
10. Montgomery, D. C., and Peck, E. A., *Introduction to Linear Regression Analysis,* 2nd ed., John Wiley and Sons, New York, 1992.
11. Box, G. E. P., and Draper, N., *Empirical Model-Building and Response Surfaces,* John Wiley and Sons, New York, 1987.
12. Design Expert, Software Package, Ver. 10, StatEase, Inc., Minneapolis, MN, 2016.
13. Minitab, Software Package, Ver. 17, Minitab Inc., State College, PA, 2014.
14. JMP, Software Package, Ver. 12.1, SAS Institute, Cary, NC, 2015.
15. Statistica, Software Package, Ver. 12, StatSoft, Inc., Tulsa, OK, 2013.

16. MATLAB, Software Package, Ver. 9.0 (R2016a), The Mathworks, Inc., Natick, MA, 2016.

17. Ulbrich, N., and Volden, T., "Strain-Gage Balance Calibration Analysis Using Automatically Selected Math Models," 41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Tucson, AZ, paper AIAA 2005-4084, July 2005.

18. Eldred, M. S., Brown, S. L., Adams, B. M., Dunlavy, D. M., Gay, D. M., Swiler, L. P., Giunta, A. A., Hart, W. E., Watson, J.-P., Eddy, J. P., Griffin, J. D., Hough, P. D., Kolda, T. G., Martinez Canales, M. L., and Williams, P. J., "DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 4.0 Users Manual," Sandia Technical Report SAND2006-6337, October 2006.

19. Myers, R. H., and Montgomery, D. C., *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, Wiley Series in Probability and Statistics, 2nd ed., John Wiley and Sons, New York, 2002.

20. Draper, N. R., and Smith, H., *Applied Regression Analysis*, 3rd ed., John Wiley and Sons, New York, 1998.

21. Peixoto, J. L., "A property of Well-Formed Polynomial Regression Models," *The American Statistician*, Vol. 44, No. 1, February 1990.

22. DeLoach, R., Rayos, E. M., Campbell, C. H., Rickman, S. L., and Larsen, C. E., "Space Shuttle Debris Impact Tool Assessment Using the Modern Design of Experiments," 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, paper AIAA 2007-0550, Jan 8-11, 2007.

23. Box, G. E. P., and Cox, D. R., "An Analysis of Transformations," *Journal of the Royal Statistical Society: Series B*, Vol. 26, No. 2, pp. 211-243, 1964.

24. DeLoach, R., Marlowe, J. M., and Yager, T. J., "Uncertainty Analysis for the Evaluation of a Passive Runway Arresting System," 47th AIAA Aerospace Sciences Meeting and Exhibit, Orlando, FL, paper AIAA-2009-1156, Jan 5-8, 2009.

25. Box, G. E. P., and Tidwell, P. W., "Transformation of the Independent Variables," *Technometrics* , Vol. 4, No. 4, pp. 531-550, 1962.

26. DeLoach, R., "The Impact of Truth Surrogate Variance on Quality Assessment/Assurance in Wind Tunnel Testing," 32nd AIAA Aerodynamic Measurement Technology and Ground Testing Conference, AIAA Aviation and Aeronautics Forum and Exposition 2016, Washington, DC, June 2016.

Use of CFD in the Design of an Experiment: The NASA Langley Isolator Dynamics Research Laboratory

Andrew T. Norris

NASA Langley Research Center, Hampton, VA 23681

1. Introduction

When designing experiments to validate computational fluid dynamics (CFD) models, it is important to make sure that the data measured in the experiment is the appropriate data required for the modeling effort. While this statement seems obvious, the process is often not so straightforward when there is minimal communication between the experimentalists and the modeling researchers. This chapter presents a validation experiment that was set up at the Hypersonic Airbreathing Propulsion Branch at NASA Langley Research Center (LaRC), in which a close collaboration between the modelers and experimentalists from the very beginning of the design process led to a well-designed experiment that provided the best quality data for the modeling effort, and where the effect of boundary conditions and measurement techniques were understood by all members of the project.

The experiment was conducted in the Isolator Dynamics Research Lab (IDRL) at LaRC, with the goal to validate CFD models that are used to characterize the flow paths of scramjet isolators. In previous work reporting on the development of this experiment [6, 7], the working of a dual-mode scramjet powered vehicle is described as follows:

"Scramjet-powered hypersonic vehicles require a high degree of integration between the propulsion system and the vehicle's airframe. The forebody surface of the airframe provides a compression surface, which processes the flow passing through the vehicle bow shock. The processed flow near the forebody surface is captured by the inlet and passes into the dual-mode scramjet isolator. The primary function of the isolator is to deliver a high-pressure flow to the combustor with minimal aerodynamic losses and to prevent the pressure rise generated in the combustor from propagating upstream into the inlet, potentially causing the inlet to unstart. In the combustor, fuel is injected, mixed, and reacted with the oxygen in the compressed flow exiting the isolator. The reacted flow expands through the scramjet nozzle and along the aftbody of the vehicle's airframe to produce thrust. Figure 1 shows the components of a typical propulsion-airframe-integrated dual-mode scramjet, and illustrates how the back pressure from the combustion process causes the boundary layer in the isolator to separate and an oblique shock train to form."

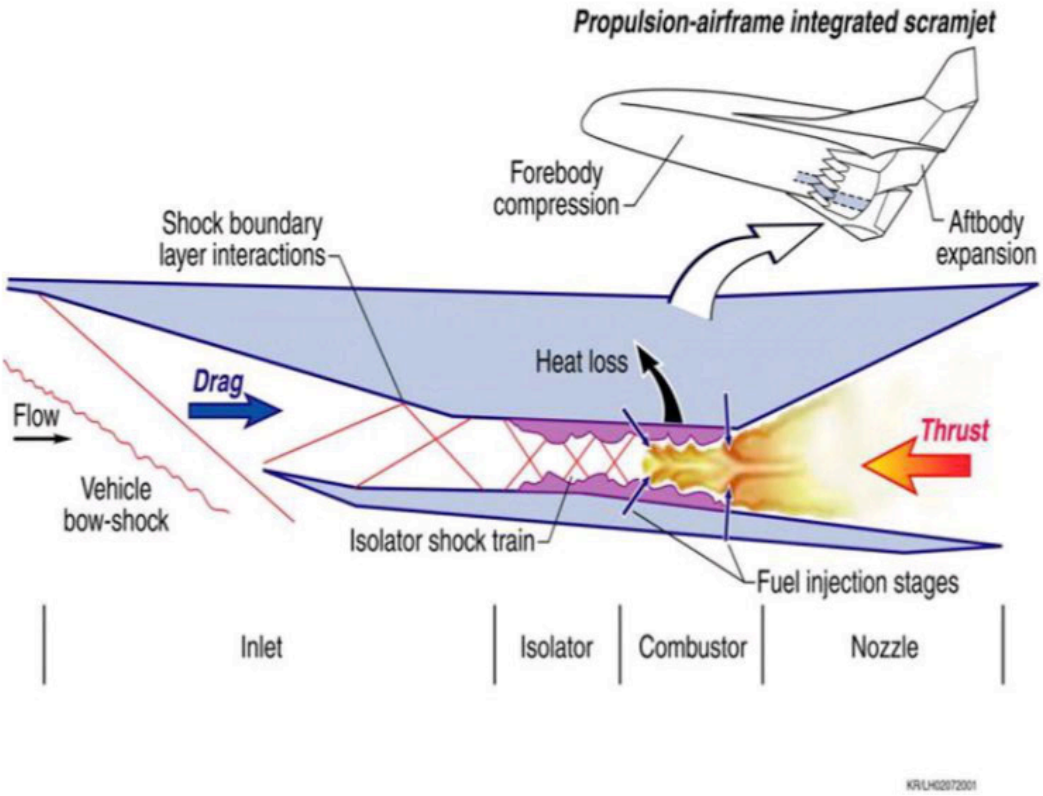


Figure 1. Typical scramjet geometry and operation [6].

When performing ground experiments to simulate a scramjet engine, especially in cold flow experimental configurations, the combustor pressure rise is commonly simulated using mechanical means, such as a flow-restricting valve. The shock pattern that develops in the isolator for a given isolator inflow condition varies as a function of the ratio of the back pressure to the inflow static pressure. For a high-pressure ratio, the boundary layer separates and a pattern of oblique shocks form a shock train within the isolator. The complex flow field has been investigated both numerically and experimentally [1-4]. However, the vast majority of the experimental measurements have been limited to wall pressure, in-stream Pitot pressure, and stagnation temperature with a limited number of nonintrusive measurements, such as shadowgraph and spark schlieren photography [5].

The goal of the IDRL is to obtain accurate off-wall measurements of the flow field to validate CFD models that are employed to characterize scramjet isolator flow paths [6-8]. To accomplish this, the IDRL supports both wall and in-stream measurements for investigating the internal flow of a dual-mode scramjet isolator model. The approximately 28-inch-long Mach 2.5 nozzle- and isolator-test-section assembly is supplied with unheated dry air from a settling chamber.

Three sets of interchangeable sidewalls (glass, metallic, and slotted metallic) are available to support flow visualization and laser-based measurement techniques, as well as wall static pressure, temperature, and high-frequency pressure measurements. Figure 2 shows the key sections that make up the test apparatus. (Note that the assembly is shown in a horizontal orientation for illustrative purpose; in the laboratory, the assembly was installed vertically with a bottom-to-top flow direction.)

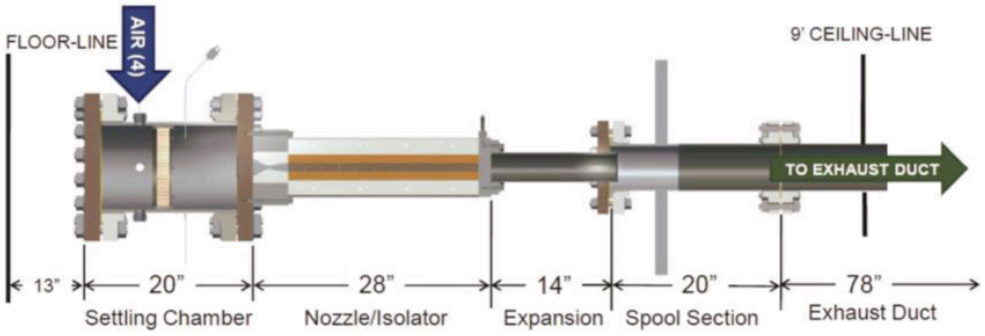


Figure 2. IDRL assembly [6, 7].

2. Design of the IDRL Test Apparatus

The principal elements of the IDRL are the converging-diverging nozzle, the isolator section, and the back-pressure plug. The Mach 2.5 converging-diverging nozzle is shown in Figure 3. The contoured surfaces of the nozzle were designed as inserts that are shimmed at assembly to insure the ratio of the exit area to the nozzle throat is precisely controlled.

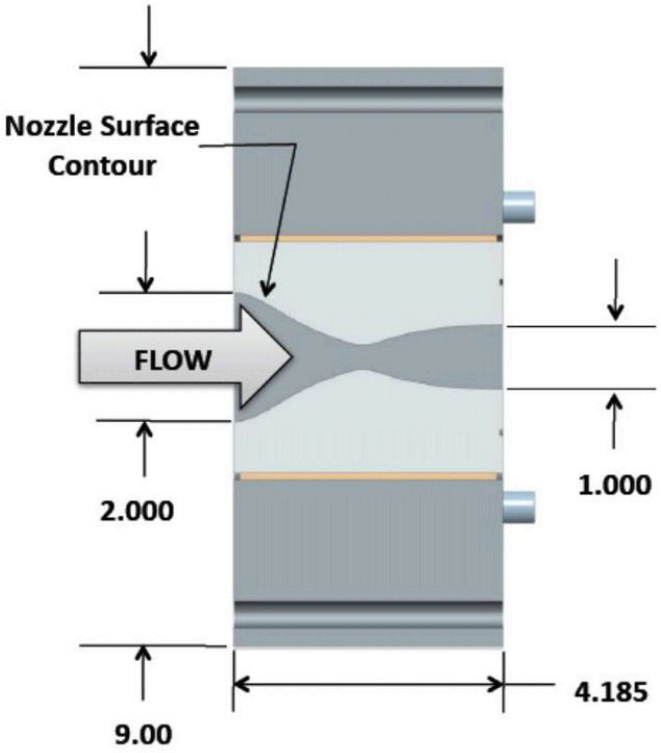


Figure 3. IRDL nozzle assembly [7].

Following the nozzle there is the 24-inch-long isolator test section, which is designed with a 1-inch by 2-inch cross-sectional flow area. The sidewalls are sealed with thin gasket material and sandwiched between four strongback supports.

To structurally support the entire test section, strongback supports are used between the Mach 2.5 nozzle flange and a downstream flange. Identically sized sidewalls of both glass and steel were manufactured and can be interchanged with each other. This provides the ability to obtain both wall and in-stream measurements simultaneously in the experiment.

The assembly and cross-sectional area of the test section are shown in figures 4 and 5.

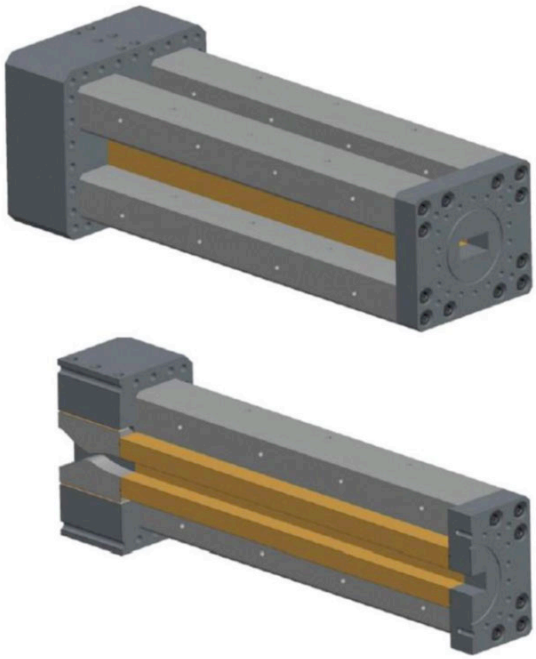


Figure 4. Isolator test section diagram [7].

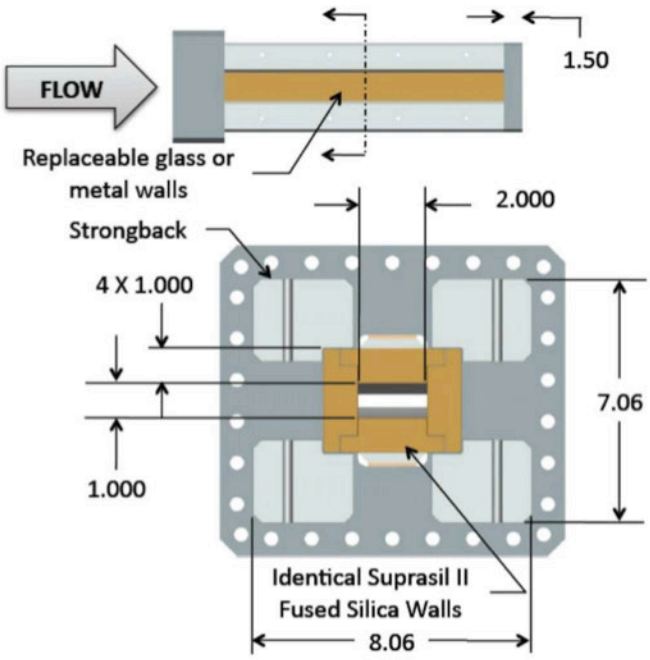


Figure 5. Isolator cross-section diagram [7].

3. Pre-Test CFD Simulations

The IDRL was designed from its inception with CFD validation in mind. Toward this end, both Experimental Fluid Dynamics (EFD) and CFD researchers were involved in all stages of the design process. This collaboration was crucial to ensure that all data requirements (as defined by the CFD team) and measurement capabilities (as defined by the EFD team) were communicated. This communication led to the determination of the diagnostic approaches to employ and the hierarchy of measurements required to maximize the value of the experimental data. As part of this process, CFD was utilized to answer design questions for various components as well as to investigate the sensitivity of the facility to changes in inflow and boundary conditions. Items addressed by the CFD simulations included:

- Design of the Mach 2.5 nozzle contour.
- Sensitivity studies to determine the control authority (precision) required for the back-pressure plug and facility supply pressure.
- Sensitivity studies to determine the influence of facility supply temperature and surface temperature, in order to determine whether a temperature control system should be considered.
- Sensitivity studies to determine the influence of facility supply air turbulence levels.
- Studies to aid with the instrumentation layout, with particular emphasis placed on the measurements required to specify and/or assess boundary conditions.
- Studies to determine the ramification of slot openings present in the specially designed isolator sidewalls required by some of the advanced laser-based diagnostics [8].

The results of these simulations and the effect they had on the design process are described below.

The contour for the Mach 2.5 nozzle was generated using an inviscid method of characteristics (MOC). To account for viscous effects, the nozzle flow field and the exit Mach number profile were then computed using the Full Navier-Stokes (FNS) solver VULCAN-CFD [9]. The results of the FNS simulation were then used to determine whether the core flow at the nozzle exit had the desired Mach number and pressure distribution. If not, then the MOC was reapplied to generate a new nozzle contour and the flow field was again simulated for the new nozzle contour using the FNS solver. This process was repeated until the target Mach number and an acceptable level of exit flow uniformity was achieved.

The Mach contours of the final design are shown in Figure 6. And, the exit Mach number and static pressure profiles from the final FNS solution are shown in Figure 7.

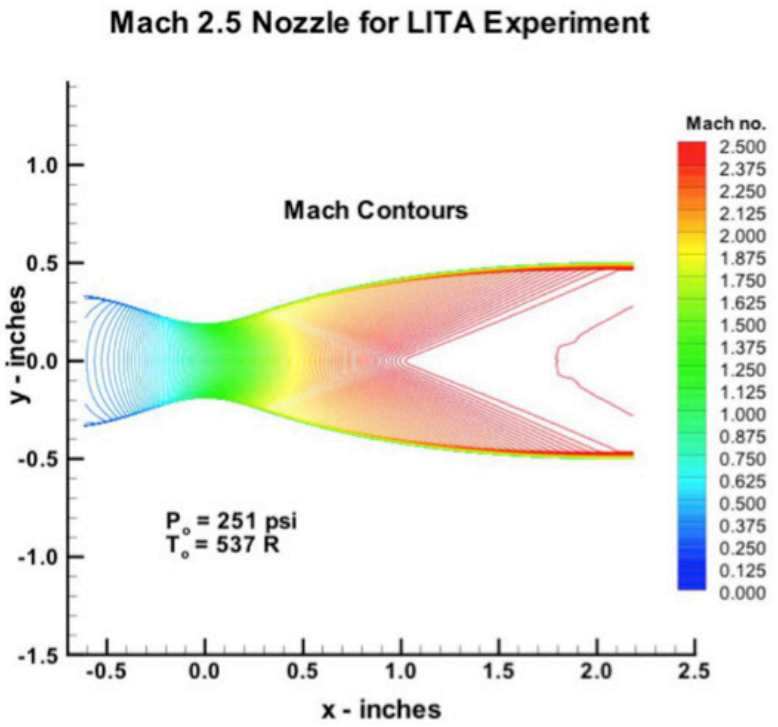


Figure 6. IRDL nozzle flow field [7].

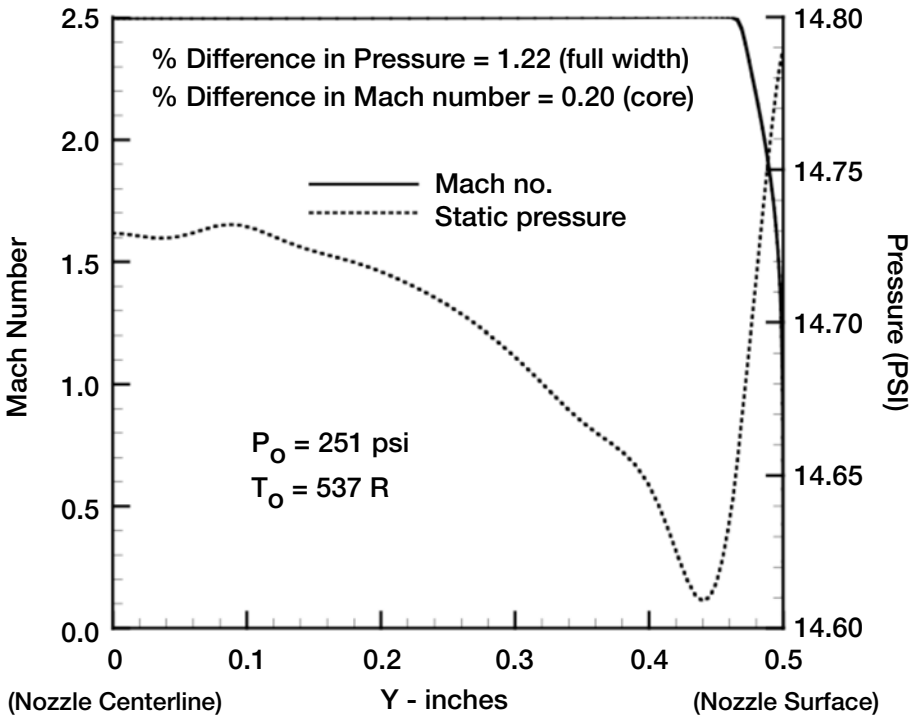


Figure 7. Nozzle exit Mach number and pressure profiles [7].

One lesson learned from the nozzle simulation was the need to investigate the required manufacturing tolerance for the nozzle surfaces before construction. After the nozzle was manufactured, the surfaces were scanned and a CFD simulation on the as-built profile of the manufactured nozzle was performed. The change in the resulting exit pressures and Mach profiles were compared to the as-designed nozzle simulations, as shown in Figure 8.

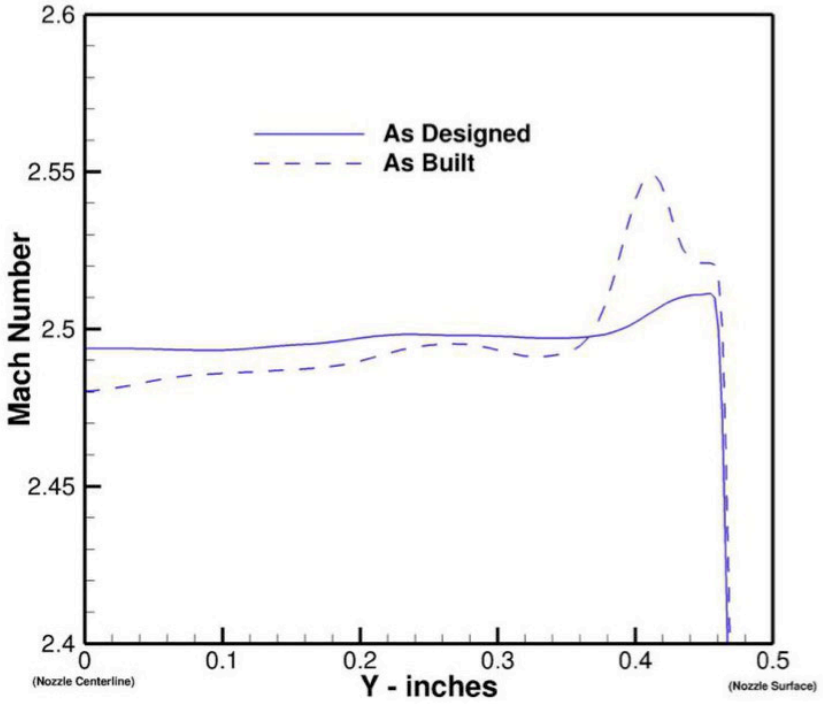


Figure 8. As-built and as-designed Mach number exit profile results [8].

The results from this post-manufacture simulation provided an example of the sort of changes that could be expected in the flow exiting the nozzle for a given fabrication tolerance. This information would have been very useful if it had been available before the manufacturing process. Providing a specified manufacturing tolerance could have ensured that the flow quality at the nozzle exit was within a desired range.

In order to evaluate the effect of the tunnel operating conditions on the facility, five cases were simulated to examine the sensitivity of the CFD solution to pressure, temperature, and turbulence intensity. It was anticipated that there would be variability in the supply pressure, so the effects of changes in air pressure needed to be quantified. In addition, the temperature of the air supply is not controlled and varies from winter to summer. And finally, due to the simple design of the air supply system, the turbulence levels cannot necessarily be well defined.

In these simulations, the back pressure was set to 99.35 psi (approximately 95% of the normal shock value) to model the likely worst-case scenario with regard to flow field sensitivities. The details of these cases are given in Table 1, where P_o is the total pressure, T_o is the total temperature, P_b is the back pressure, P is the inflow static pressure, C_f is the skin friction coefficient, X_s is the axial location of the leading shock within the isolator, and h is the height of the isolator.

Table 1. CFD simulations of isolator.

Case	Input Parameters				Results			
	P_o (psi)	T_o (R)	Turbulence Intensity	P_o P_b	Mach Number	P (psi)	C_f	X_{sh}
0	250.0	536.67	0.0001	2.51635	2.475	14.6011	1.831e-03	1.6920
1	250.0	536.67	0.0500	2.51635	2.475	14.5974	1.829e-03	1.6588
2	250.0	549.67	0.0001	2.51635	2.475	14.55.67	1.807e-03	1.7252
3	246.25	536.67	0.0001	2.47860	2.475	14.3819	1.830e-03	N/A
4	246.25	536.67	0.0001	2.51635	2.475	14.3819	1.830e-03	1.6920

Case 0 represents the baseline condition, and Cases 1–4 involve perturbations of a single parameter from this baseline condition. Mach number contours of the Case 0 simulation are shown in Figure 9.

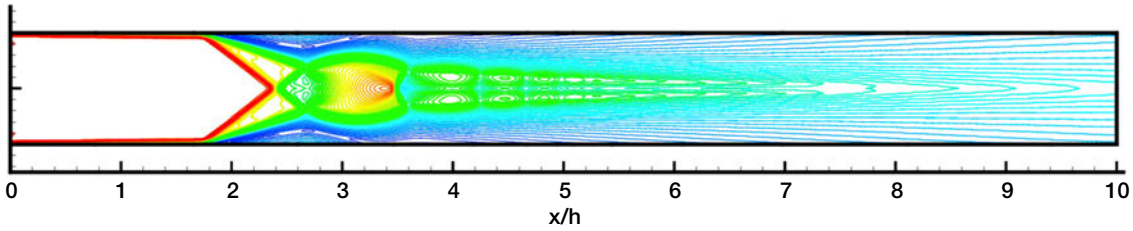


Figure 9. Baseline CFD Mach number contours (Case 0) [8].

Case 1 considers the effect of increasing the inflow turbulence intensity by a factor of 500. Case 2 considers the effect of reducing the supply temperature from a typical summer month value to a typical winter month value, evaluating the sensitivity of the experiment to variations in the air supply temperature. Case 3 considers the effect of a supply pressure variability of approximately 1.5% with the applied back pressure held constant, while Case 4 considers the effect of the supply pressure and back pressure variability where the ratio of these pressures is held constant.

The results for Case 1 show that increasing the inflow turbulence intensity by a factor of 500 had a negligible effect on the shock train position. This is a desirable result, as it shows that the level of inflow turbulence does not need to be controlled (such as in a quiet tunnel), nor do the future validation CFD runs require an extremely accurate specification of the turbulence inflow conditions. The reduction in supply temperature (Case 2) also had a negligible impact on the shock train position. This is important as well, since the air supply temperature is not controlled.

The largest deviation from the baseline condition occurred for Case 3, where the supply pressure was reduced by 1.5% while maintaining a constant back pressure at the exit of the isolator. The shock system that resulted from this case extended several duct heights upstream of the baseline condition and settled in the facility nozzle. For Case 4, the back pressure was adjusted from Case 3 such that the back-pressure-to-plenum-pressure ratio matched that of the baseline condition (Case 0). The results of this simulation show that the shock train collapsed back to what was observed for the baseline case. The wall pressure distribution for each case is shown in Figure 10.

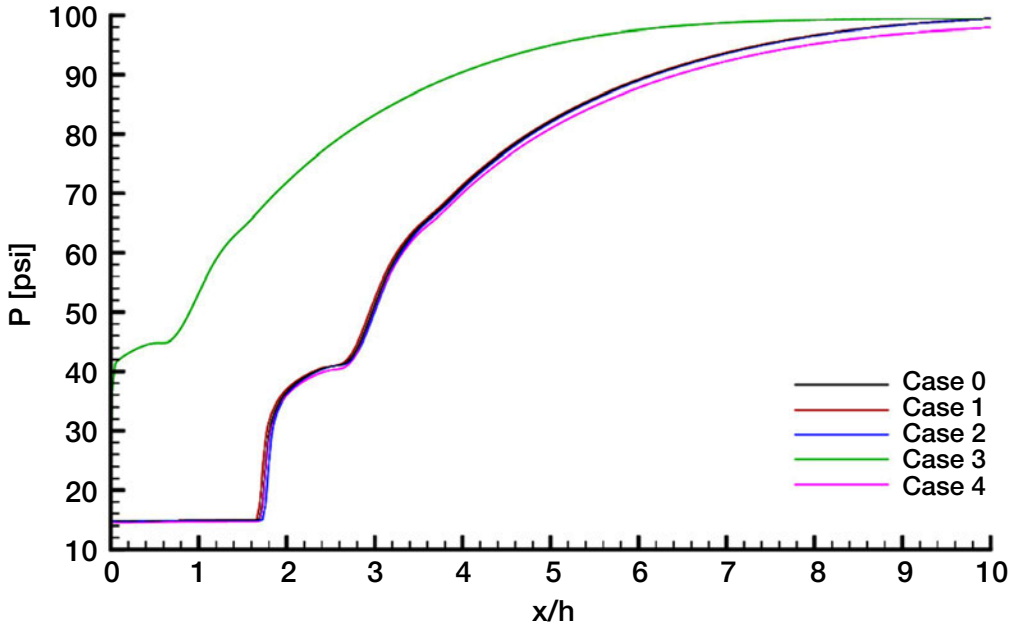


Figure 10. CFD wall pressure comparison [8].

The results of this study indicate that to maintain a constant incipient shock location, care must be taken when setting facility pressure values such that the back-pressure-to-plenum-pressure ratio is maintained at a constant value. Since the goal of the tunnel is to investigate the effect of back-pressuring an isolator, this result should be expected.

The study results also helped determine the control authority required for the back-pressure plug in order to enable repeatable experimental results. It should be noted that the pre-test sensitivity simulations for the isolator section were performed using a two-dimensional (2D) computational model; therefore, sidewall effects have been neglected. The 2D computational domain extended from the plenum, through the facility nozzle, and up to the exit of the isolator. These simulations were run for a 10-inch isolator, as shown in Figure 9. However, in order to capture a larger maximum pressure rise in the isolator, the isolator length was increased to 24 inches during the conceptual design phase. The revised length does not affect the conclusions drawn from these studies.

Since one variation of the isolator portion of the test apparatus is required to have slotted walls to provide optical access, appendages will be mounted on each sidewall, with one end open to the isolator flow field through a wall slot and the other end sealed around a quartz window. To investigate the effects that the slots might have on the isolators flow field, an additional simulation was performed with slots and appendages included in the geometry. The configuration and resulting pressure profiles at three stations and wall pressure distributions are shown in Figure 11. The results show negligible difference in the pressure profiles at the three stations and negligible difference in the wall pressure distributions for simulations run with and without the modeled slots.

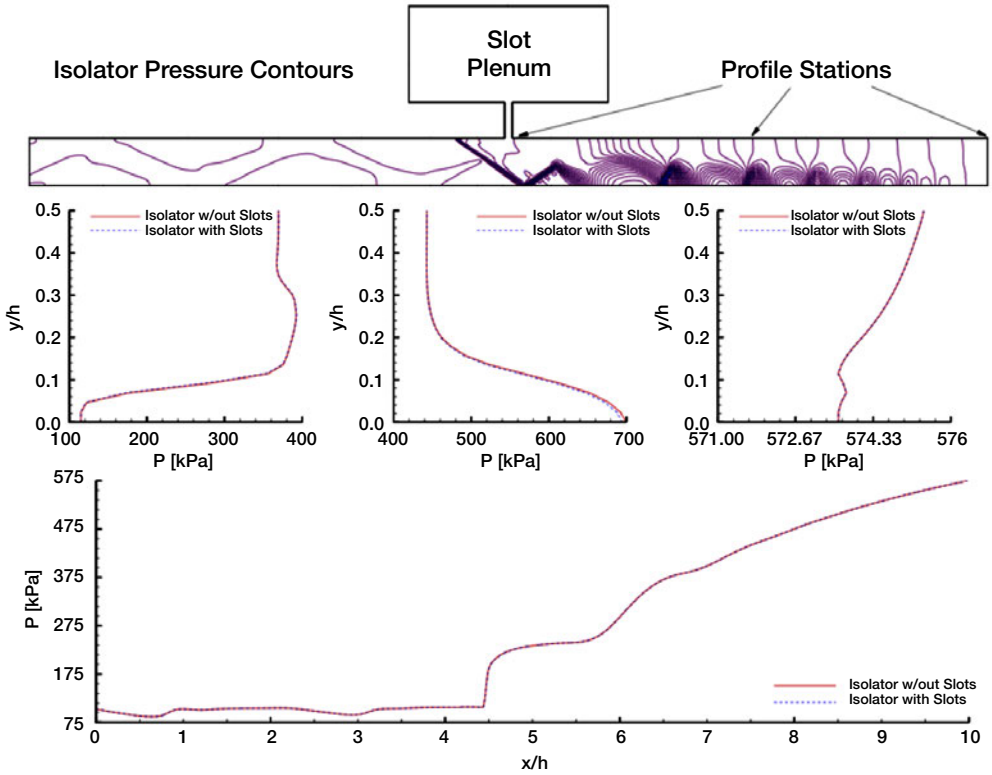


Figure 11. Simulation of the effect of access slots in isolator wall [8].

4. Conclusions

The information provided by the CFD simulations in this study showed that the boundary conditions that could not be controlled, such as the ambient temperature, would not significantly affect the flow properties of interest. In addition, it was shown that the principal effect to be investigated—the effect of back pressure on an isolator section—was highly sensitive to the back pressure ratio, which was the desired sensitivity for this experiment.

Other lessons learned include:

- Communication between experimentalists and modelers is of vital importance to the success of the experiment.
- A rigorous pre-design CFD analysis can help ensure the success of an experiment, ensuring that the experiment remains insensitive to factors that cannot be controlled and sensitive to the factors of interest. In addition, these design simulations help determine the control authority required to ensure experimental repeatability.
- The effect of small changes in geometry can be investigated using CFD analysis to determine the effect of manufacturing tolerances on the experiment.

References

1. Reintarz, B. U., Herrmann, C. D., and Ballmann, J., "Aerodynamic Performance Analysis of a Hypersonic Inlet Isolator Using Computation and Experiment," *Journal of Propulsion and Power*, Vol. 19, No. 5, 2003.
2. Auslender, A. H., "An Analytic Performance Investigation of Mechanically Back-Pressured Ramjet Data," JANNAF 34th CS/PSHS/APS Joint Meeting, October 27-31, 1997.
3. Rice, T., "High Aspect Ratio Isolator Performance for Access-to-Space Vehicles," 12th AIAA International Space Planes and Hypersonic Systems and Technologies, paper AIAA 2003-7041, 2003.
4. Emami, S., Trexler, C. A., Auslender, A. H., and Weidner, J. P., "Experimental Investigation of Inlet-Combustor Isolators for a Dual-Mode Scramjet," NASA Technical Paper 3502, 1995.
5. Le, Daniel B., Goyne, Christopher P., Krauss, Roland H., and McDaniel, James C., "Experimental Study of a Dual-Mode Scramjet Isolator," *Journal of Propulsion and Power*, Vol. 24, No. 5, pp. 1050-1057, 2008.
6. Middleton, T. F., Balla, R. J., Baurle, R. A., and Wilson, L. G., "Laser-induced Thermal Acoustic Measurements in a Highly Backpressured Scramjet Isolator Model: A Research Plan," 30th JANNAF Airbreathing Propulsion Subcommittee Meeting, May 12-16, 2008.
7. Middleton, T. F., Balla, R. J., Baurle, R. A., Humphreys, W. M., and Wilson, L. G., "The NASA Langley Isolator Dynamics Research Lab," JANNAF 31st Airbreathing Propulsion Subcommittee Meeting, December 7-11, 2010.
8. Baurle, R. A., Middleton, T. F., and Wilson, L. G., "Reynolds-Averaged Turbulence Model Assessment for a Highly Back-Pressured Isolator Flowfield," JANNAF 32nd Airbreathing Propulsion Subcommittee Meeting, December 3-6, 2012.
9. Baurle, R. A., Vulcan CFD Home Page, Hypersonic Airbreathing Propulsion Branch, NASA Langley Research Center: http://vulcan-cfd.larc.nasa.gov.

An Introduction to Some Model Validation Concepts and Paradigms and the Real-Space Approach to Model Validation

Vicente Romero

Sandia National Laboratories,[1] Albuquerque, NM 87815

1. Introduction

Model validation is a fast-developing field in engineering and science. A wide variety of viewpoints and methodologies regarding model validation exist in the literature. References 1-14 explore many important philosophical and implementation issues in model validation, and survey various paradigms and tactical approaches for performing model validation for application problems. Some of these references go on to develop a specific approach and implementation framework.

The case study in Chapter 11 uses the "Real Space" (RS) model validation approach. The RS uncertainty accounting and comparison system for comparing experimental and model results in the context of their uncertainties is also useful in model calibration or conditioning (e.g., Romero, et al. [15]). The following philosophical, conceptual, and implementation considerations underlie the formulation of the RS approach.

The rationale for the RS validation approach is presented by Romero [12, 13] and referenced papers therein. In those references, established definitions and paradigms of model validation are evaluated and a new paradigm and "operational" definition of model validation (slightly modified here) are offered:

Model Validation is the compilation of useful indicators regarding the accuracy and adequacy of a model's predictive capability for output quantities of interest, where meaningful comparisons of experiment and simulation results are conducted at points in the modeling space that present significant prediction tests for anticipated uses of the model.

This definition is accompanied by a view that a model can only be "point-validated" (assessed for accuracy and adequacy) at the specific validation points in the modeling space where appropriate validation experiments and assessments exist, and only for assessed prediction outputs and purposes; general statements of model "validity" for extrapolation prediction uses cannot be legitimately established. This position is also expressed in many of the references cited above.

[1] Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Other caveats regarding the interpretation of model validation results are discussed in the last section of this chapter.

The RS model validation approach adopts some elements and constructs from the literature (sometimes adding needed refinement), and adds new pivotal elements and constructs. The approach reflects pragmatism, versatility, and capabilities derived from the author's experience with many industrial-scale validation and calibration applications involving: device thermal response and failure [16, 17]; thermal-structural response and failure [18, 19]; performance of radiation damaged electronics [15, 20]; and combustion in fluids and solids [21, 22]. These problems involved complex physics and constitutive models, steady-state and time-varying nonlinear behavior and boundary conditions, and various sources and types of uncertainty in experiments and models. Among the many model validation paradigms, frameworks, and methodologies considered by Romero [12, 13], the RS methodology was uniquely capable to handle the diverse set of challenging attributes in these validation problems. See References 12, 13, and 23, and Appendix D of Chapter 11 for a discussion of the capabilities of the RS methodology versus other established validation methodologies.

The RS uncertainty accounting and comparison approach has evolved to address various kinds and aspects of uncertainty (aleatory, epistemic, probabilistic, interval, continuous, discrete, correlated, uncorrelated) and combinations of these that commonly arise in experiments and simulations, including:

- Aleatory experimental variability in multiple repeat experiments and the accompanying epistemic uncertainty from limited numbers of repeats.
- Random and systematic uncertainties in measurement and/or processing of experimental and simulation inputs and outputs.
- Boundary condition reconstruction uncertainty from spatially sparse measurement data.
- Numerical solution uncertainties from model discretization and solver effects and other non-experimental uncertainties that are intrinsic to the model.

As with any model validation framework, challenges also arise in other areas:

- Deciding what is adequate in terms of an abbreviated basis of raw and processed output quantities for model validation or conditioning comparisons (not discussed further in this document, see Romero [12, 13]).
- Model *accuracy* characterization—deciding on the formulation or metric for characterizing the discrepancy between model and experiment results.
- Model *adequacy* characterization—deciding on the threshold or criterion for model adequacy (acceptable agreement with reality).
- Dealing with extrapolation of model validation or conditioning information/results/products—deciding how to project results and outcomes from validation or conditioning settings to other prediction settings. (This is not part of model validation, but *is* an essential consideration in the formulation of a workable validation paradigm.)

These items are considered to be strongly interdependent. It is reasoned that only by considering them together can a comprehensive, workable, and relevant model validation framework be formulated.

In contrast to other established validation approaches reviewed by Romero [12, 13], RS does not formulate an assessment of model accuracy and adequacy in terms of "transform" discrepancy metrics and pre-specified acceptance criteria. See Section 2 for elaboration. Besides greater transparency and interpretability from remaining in real space, RS provides a greater ability to discriminate important differences between model and experimental results that affect prediction risk in downstream applications of the model. Furthermore, workable approaches in the literature to assess the *adequacy* of model-experiment agreement in transform space remain elusive, whereas a "zeroth order" bounding approach to this difficult issue can be applied in the RS approach. See the example in Section 5 and in Section 5.4 of Chapter 11 for further discussion.

Also unique to the RS methodology is the introduction and use of the concepts of "traveling," "non-traveling," and "experiment" models and uncertainties. The RS framework clearly defines the model of interest by recognizing two different scopes of the physical problem in model validation, conditioning, and extrapolation. The specific identification of the model to be validated or conditioned is referred to as the *traveling model*. The traveling model delineates which models within a model validation or conditioning activity will be used in subsequent predictions. The traveling model carries an implicit connotation that the physical attributes and phenomena it represents are proposed to carry over consistently to the new prediction settings, such that the traveling model applies there as well.

The traveling model is a subset of the larger set of models employed in the validation activity, referred to as the *experiment model* or *e-model*. This larger set of models cannot be avoided. Whatever the model of downstream or traveling interest is—whether a set of partial differential equations (PDEs), a material-behavior constitutive model, a model of a hardware device, a process model, and so forth—the experiment will have elements that need to be modeled but are auxiliary to the elements in the traveling model of interest. See Section 2 below for elaboration.

The RS approach treats uncertainties in various aspects of the validation experiment according to whether they are traveling uncertainties or non-traveling uncertainties, and according to how these uncertainty types differently affect prediction risk; see Refs. 13, 17, and 24.

It is also recognized that substantial freedom often exists to define the scope of the experiment model in a model validation assessment; and the scope should be selected to minimize uncertainties in the assessment. This is one element of optimized design of model validation experiments discussed in Section 3. The topics introduced here are elaborated in the following, along with a description of generic steps in model validation activities (Section 4) and appropriate interpretation and use of model validation results (Section 5).

2. Traveling, Non-Traveling, Experiment Models and Uncertainties, and Definition of "Model" in Validation and Prediction Contexts

It is important to clearly identify the model or set of models being validated in a validation activity. This seems obvious, but often, validation frameworks and activities are ambiguous in this respect, which can lead to confused interpretation and improper use of validation results. Improper accounting of uncertainties and their propagation to predictions also results. The RS paradigm's concepts of experiment models and traveling models [17, 24, 12, 13] are essential to distinguish what model or set of models is being validated, as explained next.

Consider a 1-D heat conduction experiment involving a heated rod. If a finite-element model is built to simulate the experiment, how does one target or differentiate whether the 1-D heat

diffusion equation (PDE) alone is being validated, or whether the validation applies to the larger set of models (equations and parameter values/ranges) consisting of the PDE and the geometry, material property, and boundary condition descriptions? The model to be validated could also potentially include the affiliated discretization scheme and solution algorithm.

Specifically identifying the model or set of models to be validated is important for planning and performing the validation activity and interpreting and using the results. The traveling model concept is used to define the boundary of the model to be validated. The traveling model delineates the set of models in a validation or calibration activity that will be used (as a set) in subsequent predictions. This is a subset of the larger set of models employed in the validation or calibration activity, referred to as the experiment model or e-model.

For example, a physical system of traveling interest could be an electromechanical device, for which a model is being validated. The physical device is part of a larger hardware system. A finite-element (FE) model of the device is to be validated under loading in the validation experiments. The model is then to be used in a hierarchical system-level model to make predictions of device response under *other* loads that the device may experience within the assembled system subjected to loading at the system level. Here, the hardware device in the validation experiment is the traveling physical system and the FE model is the traveling model, but the applied loads in the validation experiment are *not* part of the traveling system or model.

In contrast to the definite image that a traveling hardware device and FE model evoke, the traveling physical system and model can also be a generic phenomenological behavior such as material property behavior or turbulence. Here, the traveling model is a phenomenological equation set with associated parameter values/ranges, but without a specific traveling geometrical object form. For example, to develop or validate a constitutive equation set for elastic-plastic behavior of a certain material, the experiments must involve a physical specimen of material, shape/geometry, initial state, and applied loads. A corresponding e-model of the experimental instantiation is built. Any e-model aspects such as applied loading and specimen shape/geometry that are not part of a physical setting for which the constitutive model is later used are outside the traveling model.

Although only a portion of the experiment model travels to application prediction settings away from the validation setting, uncertainties from non-traveling sources in the validation experiments and predictions (such as measurement uncertainties and discretization-related solution errors quantified by calculation verification) may contribute significantly to any prediction bias and uncertainty quantified in the validation assessment. It may be appropriate for these uncertainties to accompany the traveling model to predictions as “hitchhiker” uncertainties—depending on the outcome of the validation assessment and the circumstances in subsequent predictions. Hitchhiker uncertainties can be mapped into parameters of the traveling model (e.g., Refs. 15 and 24) or they can be carried with the traveling model as auxiliary functions that operate on model prediction results to adjust the predictions (e.g., Refs. 17 and 25). Uncertainties from the non-traveling sources depend on the design and implementation of the experiments, the diagnostic instrumentation used, and the chosen scope of the e-model—all of which should be optimized through good experiment design to reduce or eliminate the non-traveling uncertainty sources as much as achievable within project constraints.

The traveling model can also contain *model-intrinsic* non-hitchhiker uncertainties that are inherently affiliated with the traveling model, such as an uncertainty range on parameter values in a turbulence or material model. These uncertainties are defined prior to the experiment (they are

not determined by or in the experiment) and come to the prediction model as *a priori* uncertainties in model form and/or parameter values.

Uncertainties are segregated and treated differently in the RS framework, according to whether they come from traveling vs. non-traveling sources in the validation activity. This is because they affect extrapolation risk differently, as demonstrated by Romero [13, 24]. Other model validation frameworks appear to lack the concepts and different uncertainty treatments associated with traveling, non-traveling, and experiment models.

3. Validation Metrics and Real Space Discrepancy Characterization

Validation metrics are defined by the American Society of Mechanical Engineers [7] as "mathematical measures that quantify the level of agreement between simulation and experimental outcomes." The Real Space validation approach is so named because it differs from other established frameworks that employ validation metrics in a discrepancy "transform space," e.g., the Subtractive Difference metric in Refs. 2 and 7, and the cumulative distribution function (CDF)-mismatch 'Area' validation metric in References 11, 26, and 27. Figures 1 and 2 present some differences between these approaches. The physical and decision-making significance of the numerical values yielded by these transform measures is unclear.. The transforms can also constrain what forms and types of uncertainty can be handled appropriately; see Refs. 12, 13, and 23, and Appendix D of Chapter 11 for more information.

RS discrepancy characterization better reveals characteristic differences between model and experimental results that affect prediction risk. Figures 1 and 2 show that the Subtractive Difference and Area transform metrics have non-exclusive mappings between real space and transform space, where the same transform space value can accompany different conditions in real space. Therefore, it can be risky to use transform-space metrics to make validation judgments on model performance and adequacy, and to guide model conditioning and extrapolation.

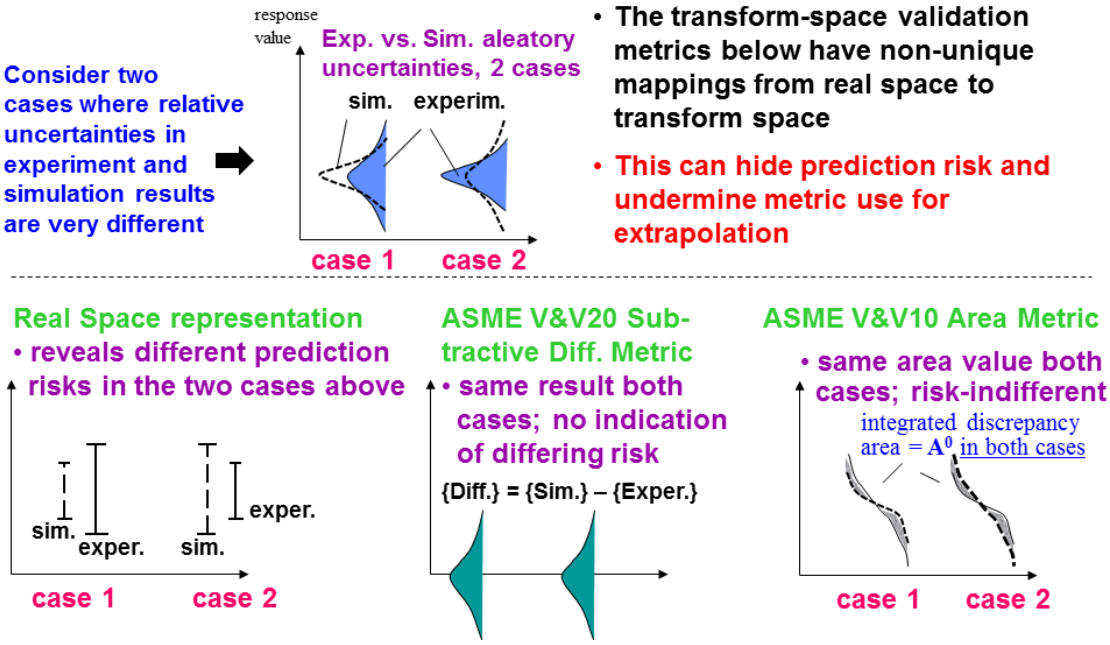


Figure 1. Real Space and transform-space representations of model discrepancy (from Romero [13]).

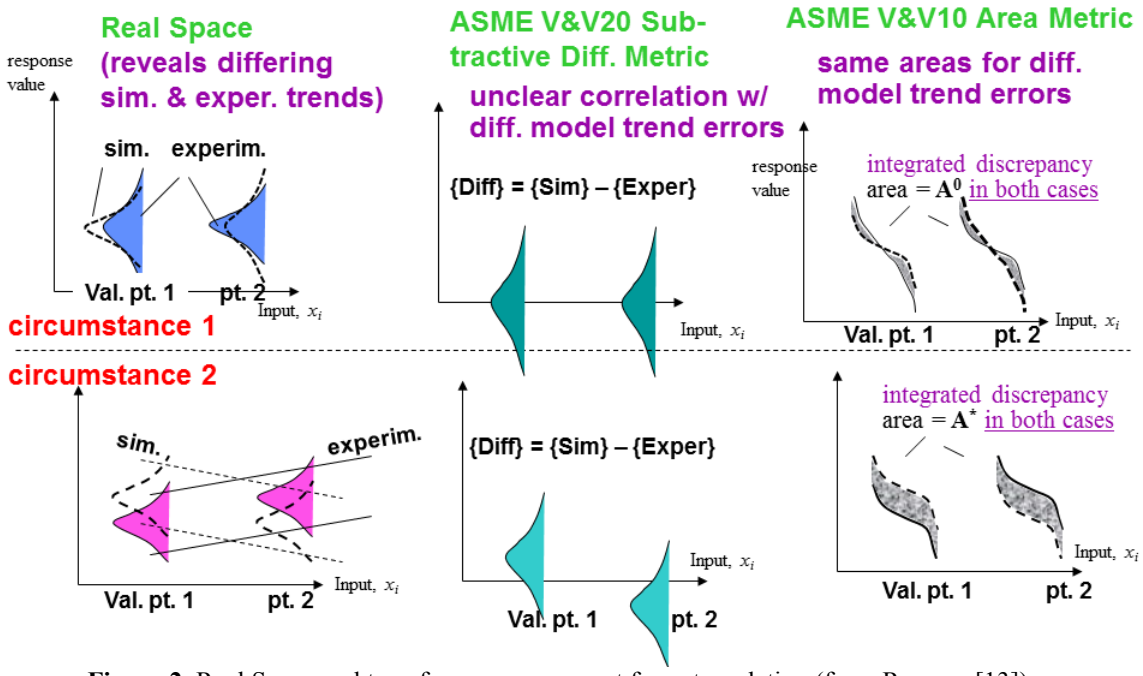


Figure 2. Real Space and transform space support for extrapolation (from Romero [13]).

RS quantification and presentation of discrepancy results segregates aleatory and epistemic uncertainties in the experiments and simulations with affordable "Level 1" Approximate Probability Boxes (see Section 5 of Chapter 11). The RS method presents simulated and experimental results and their uncertainties in an intuitive, visual-quantitative format as illustrated in Figure 41 in Chapter 11. This format helps to more effectively convey the information message from the validation assessment. Figure 41 shows that the prediction discrepancies are different at low percentiles than at high percentiles, so are not adequately expressed by a single discrepancy value. An analogue is the use of visual means like color contour plots to more fully convey the information in spatial fields like temperature and velocity fields than the use of numbers alone can provide. Comparison of percentiles of experimental and simulation response, as shown in Figure 41, above, is especially suited for assessing models to be used in the analysis of performance and safety margins. The RS method provides the ability to perform similar comparisons of other experimental and simulation response statistics, such as means and standard deviations.

Furthermore, workable approaches in the literature to assess the adequacy of model-experiment agreement in transform space remain elusive, whereas a zeroth order bounding approach to this difficult issue can be applied in the RS approach. See the example in the last section of this chapter and Section 5.4 of Chapter 11 for further discussion.

4. Design of Model Validation (and Calibration) Experiments and Scope of the Experiment Model

Balanced judgment must be applied to the following considerations to derive the most benefit from validation experiments and assessments.

Because of the uncertainties that extrapolation presents, it is desirable to plan validation experiments to be as close as possible to the actual conditions of the applications for which a model is to be used. Of course this is also a tactic for the calibration of models. This increases the applicability and relevance of the validation or calibration results to the intended applications.

Countervailing objectives and constraints often cause validation experiments to be relatively simple and far (in modeling parameter space) from the intended model use conditions. One driver for this is the need to control conditions in the validation experiments in order to maximize resolution power to isolate model bias. Cost and technical practicality also often drive validation experiments to be simpler than the eventual applications. In many cases, such as nuclear weapons testing and nuclear power plant accidents, tests near to the intended prediction space are not feasible. Moreover, in many cases, extrapolation robustness depends on whether the model—if it provides acceptable results in the validation conditions—provides these principally "for the right reasons." This can be established to some degree via a *hierarchical* validation approach where the model is decomposed into sub-groupings of phenomenological effects at an affordable level that allows validation of the separated pieces of the model. Optimally cost-effective decomposition depends on many things. Modeling and simulation are very useful for helping to optimize the decomposition. Hierarchical model validation decomposition procedures and information integration from the separate validations are active areas of research. Other aspects of good experiment planning include:

A. Use of modeling to help design the physical experiments and diagnostic measurements as elaborated below;

B. Optimizing the scope/boundaries of the experiment model (*e-model*) and of the traveling model to reduce uncertainty and increase resolution in the validation assessment.

Regarding item B, the validation experiment usually does not dictate a unique scope for the experiment model. For example, a bank of heat lamps in an experiment warms a heat-spreader plate, which then radiates to a test object of interest. The heating conditions on the test object are needed. The scope of the e-model could stop at the heat spreader plate that radiates to the test object, or could extend to include the banks of heat lamps in the model. The former is preferable because it is much easier and less uncertain to paint the heat-spreader plate with an emissivity-controlling paint and instrument it with thermocouples (providing a reasonably accurate radiative heating condition applied to the test object) than to try to model the complex geometries of the heating lamps and accurately determine and specify their effective temperatures and emissivities.

As another illustration, consider a fuel-pool fire being modeled with a fire CFD code. If the fuel-level regression rate of the pool of fuel is measured in the fire experiment, then the fuel vaporization rate that feeds the fire can be calculated from fundamental principles and other experimental measurements of ambient pressure and liquid fuel-pool temperature. The fuel vaporization rate could then be used as input experimental data to the fire CFD model. Alternatively, the fire CFD code may offer a modeling option to calculate fuel vaporization from inputs to the model of measured ambient pressure and initial fuel-pool temperature. From these and the fire simulation's calculated heat flux to the pool surface and the consequent evolution of fuel surface temperature, the fuel vaporization rate that feeds the fire can be calculated without the measured fuel regression rate and its uncertainty, but *with* other uncertainty from the added vaporization model. Thus, various utilizations of measured data, traded off with the potential modeling of various phenomena instead of measuring it, affect the scope and form of the e-model and its uncertainty.

Accordingly, when one sets out to model a validation (or calibration) experiment, there are often multiple options regarding the model's scope. Questions to consider are: What are the model boundaries? What phenomena are to be modeled and what are to be input to the model from experimental measurement or published values or empirical relationships? Although a unique choice usually does not exist with regard to scope of the experiment model, a *best* choice can often be made—that which most reduces uncertainty in the validation (or calibration) activity, given the available freedoms in the experiments and modeling.

The uncertainties of all the options should be assessed up front, with a goal of overall uncertainty minimization through:

- Careful design of the physical experiments, including appropriate ranges and levels of initial and boundary condition inputs, system configurations to be tested, and numbers of experimental repeats, as affordable).
- Appropriate selection of scope of the e-model.
- Design and selection of measurement and sensor types, locations, and spatial and temporal resolutions; and supporting measurement/diagnostics instrumentation and their calibration.
- Possible procurement of more and/or better equipment for measuring inputs and outputs, and control of inputs such as applied boundary conditions.
- Procurement of sufficient information on material properties, geometries, etc. of the elements of the systems being tested.

Optimized experiment design in these areas is very beneficial. Such optimization is enabled immensely by computational modeling and simulation. With model-assisted experiment design**,** the experiments and planned measurements are modeled before the experiments are finalized, with the goal of identifying (to the greatest extent possible) the best experimental settings, measurements, instrumentation, number of experimental replicates, and e-model that minimize validation uncertainty.

Examples of use of modeling and simulation to reduce error and uncertainty in validation experiments and assessments are presented in Section 4 of Chapter 11. Note that model-assisted experiment design includes but goes beyond the concepts of traditional statistical design of experiments (e.g., Box, Hunter, and Hunter [28]). The latter deals with planning input levels and test replications to sample a parameter space of system operation for efficient discernment and analysis of process or product variation over noise variables and exploratory/control/operational variables.

5. Steps in a Model Validation Activity

The following steps within the RS validation framework are thought to generically apply for other model validation frameworks as well, except for the particulars of the RS discrepancy characterization approach and uncertainty treatment. Other validation frameworks may contain their own specific steps.

1. Identify the modeling setting(s), e.g., initial and boundary conditions, use circumstances, and corresponding Quantities of Interest (QOI) for which model predictivity is to be assessed.

2. Decide on specific validation settings and activities (identify achievable experiments and outcomes) according to the following constraints and objectives.
   a. **Constraints:** Project resources (funding, allotted time, personnel, capabilities and expertise), experimental apparatus, and "allowed" and achievable experiments and experimental conditions.
   b. **Objectives:** Validation experiments should be as close as reasonable to the actual conditions of the applications for which the end-use model will be used. This reduces uncertainty associated with extrapolation. A second and often competing objective of establishing a high degree of credibility that the model is principally "right for the right reasons" calls for a hierarchical decomposition and approach to the validation problem, which competes with resource priorities for the integral validation activities. Modeling and simulation should be used as possible to help arrive at the best tradeoff between these objectives, and to design and optimize the physical experiments (geometries, initial and boundary conditions, input and output measurement sensor locations, and so on).
3. Perform the planned experiments and form uncertainty representations for the approximation and inference errors in measurement and processing of the experimental input and output quantities in the multiple replicate experiments. Categorize the errors and corresponding uncertainties as:

Traveling	Non-traveling
Fully correlated ("systematic") over all experiments	Fully correlated ("systematic")
Uncorrelated (randomly varying) from exper. to exper.	Uncorrelated (random)

   If only one experiment is conducted, all uncertainty is categorized as "epistemic" in the Real Space method (a measurement error exists per input or output quantity and the error is unknown but proposed to lie within a distribution or interval range of epistemic uncertainty). For examples in several of these categories, see Section 5.3 of Chapter 11.
4. Process experimental results and uncertainties according to the uncertainty qualification (UQ) methodology explained in Section 5.3 of Chapter 11. Often, construction of a linear or quadratic polynomial response-surface surrogate model for uncertainty propagation is sufficient for model validation purposes. Then, relatively simple spreadsheet procedures can be used for the uncertainty propagation, as shown in Chapter 11. Simpler or more complex experimental UQ procedures can be found in Refs. 15, 19, and 21, depending on the categories of traveling and non-traveling uncertainties involved (see Step 3). Obtain uncertainty of QOI response statistics of interest for validation comparisons against predicted results: mean behavior, standard deviation, and/or selected percentiles of response.
5. Perform solution verification at the reference nominal values of the experimental input conditions (identified in Step 4) that anchor the processed uncertainty of experimental and simulation QOI results. Alternatively, state a plausible modeling scheme to be used to attain consistent (traveling) discretization-related solution bias error in model validation and end-use settings (plausible schemes may exist in some cases, see discussion by Romero [12, 13]). If a solution verification approach is used, perform solution verification at one or more other strategically selected points in the uncertainty parameter space (see Step 6) of the validation exercise. This is used to assess whether the

discretization correction in Step 6 is robust over the UQ parameter space. See e.g., Romero [29] for selection of strategic points for such solution verification studies.

6. Categorize the uncertainties associated with the model into the following structure, and process the uncertainties accordingly using the UQ methodology demonstrated in Section 5.2 of Chapter 11, or see Refs. 15, 19, and 21, as appropriate.

Traveling	Non-traveling
Modeled stochastic variability of the traveling systems	Modeled stochastic variability in the experiments and e_model that are not in the traveling systems or model
Epistemic uncertainties associated with the prediction model of the traveling systems	Modeled epistemic uncertainties associated with the experiments, e_model, and any response surface surrogates for it, and numerical simulations, that are not in the traveling systems or model

Propagate uncertainties with the highest-resolution mesh and solver combination that can be afforded for the chosen propagation procedure. Often, a linear or quadratic polynomial response-surface surrogate model for uncertainty propagation is sufficient for model validation purposes. Regardless, relatively simple uncertainty propagation procedures, like those in Refs. 15, 19, and 21 or Section 5.2 of Chapter 11, are available depending on simulation resource constraints and uncertainty descriptions in the problem. From the propagated and aggregated uncertainties, obtain uncertainty of QOI response mean, standard deviation, and/or selected percentiles for validation comparison.

7. Compare uncertainties of experimental and predicted statistics of QOIs. Quantify prediction bias uncertainties for these statistics for each QOI. See Section 5.4 of Chapter 11 for an example comparison and interpretation of RS validation results.

6. Interpretation and Use of Model Validation Results

Model validation information like that given in Chapter 11, Section 5.4 helps to contextualize model predictiveness at the validation conditions. In the VV/UQ case study, inferred experimental failure pressures at the lower end of the inferred failure pressure spectrum were significantly lower than the model predicted—indicating that the model may be unsuitable for design and safety analysis purposes. At the upper end of the spectrum, the experimental and simulation uncertainties were not sufficiently small to determine whether the model is biased high or low relative to experimental failures, but at least the potential range of prediction bias error was quantified.

As in the case study, the assessment of model adequacy often comes in the form of whether the model yields conservative predictions—but not overly conservative—at the validation points in the modeling space. If so, the possibility that this circumstance will be preserved at interpolation or extrapolation points in the modeling space is referred to as "zeroth-order" adequacy establishment [12, 13]. Very frequently in model validation projects, more definitive statements on the adequacy of the model for anticipated uses are not rigorously supported or allowed by the information in the validation activity. Refs. 12 and 13 contain extended discussions of the necessary information, problem structure, and implements for proceeding beyond a zeroth-order indication of model adequacy for proposed predictions.

In particular, the common paradigm (e.g., Refs. 14 and 27) of proposing pre-specified accuracy requirements against which a model is to be validated for adequacy is argued by Romero [12, 13] to be unworkable in practice. A primary reason is that non-uniqueness and "parametric disconnect" usually make it is impossible to derive quantitative accuracy requirements that are not substantially arbitrary even if a prediction end-use of the model has been identified. Rather, *relative* (ordinal) determinations like the zeroth-order adequacy approach that are mutable to various uses of the model are found in practice to be more workable.

Thus, validation assessments rarely yield definitive yes/no statements about whether a model is valid for a proposed use. Rather, validation provides a crucial source of information in a larger overall evaluation of model suitability for the proposed use. Decision makers can use the quantitative and qualitative validation information to supplement other project information, such as risk-reward stance, to make more informed—although perhaps still highly subjective—decisions on model acceptability for identified use purposes.

For model uses that involve significant extrapolation beyond the validation conditions, subject-matter experts are vital to make judgments concerning model adequacy for the proposed predictions. If predictions are made, the prediction bias and uncertainty information from validation can be used to nominally bias-correct the predictions. How to best do this to mitigate prediction risk beyond the validation conditions is beyond the scope of model validation methodology. This is a very difficult issue and an active area of research. See Chapter 5 for further discussion and references on this topic.

Although validation assessments yield limited circumstantial corroboration of model predictiveness, carefully designed and executed validation experiments and assessments can add greatly to the modelers' and the model users' knowledge about the performance of a model. This is an essential element of due diligence and applied risk management in developing and using models. Likewise, although accuracy of predictions cannot be guaranteed, contextualizing and improving the predictions as much as possible through appropriate methodology is important. We can seek to maximize accuracy through careful design and analysis of experiments and model development, validation, and extrapolation procedures optimized for the desired prediction tasks. Having said this, quality assessment and control in modeling and simulation are still in the early stages of development. Much still needs to be done to bring these young engineering sciences to maturity.

References

1. Mehta, U. B., "Guide to Credible Computer Simulations of Fluid Flows," *Journal of Propulsion and Power*, Vol. 12, No. 5, pp. 940-948, 1996.

2. Coleman, H. W., and Stern, F., "Uncertainties in CFD Code Validation," *Journal of Fluids Engineering*, Vol. 119, pp. 795-803, Dec. 1997.

3. American Institute of Aeronautics and Astronautics, *AIAA Guide for the Verification and Validation of Computational Fluid Dynamics Simulations*, AIAA Standards, AIAA-G-077-1998, 2002.

4. Roache, P. J., Verification and Validation in Computational Science and Engineering, Hermosa Publishing, 1998.

5. Anderson, M. G., and Bates, P. D., editors, *Model Validation—Perspectives in Hydrological Science*, Wiley, 2001.

6. Trucano, T. G., Pilch, M., and Oberkampf, W. L., "General Concepts for Experimental Validation of ASCI Code Applications," Sandia National Laboratories Report SAND2002-0341, March 2002.

7. American Society of Mechanical Engineers, *Guide for Verification and Validation in Computational Solid Mechanics*, American Society of Mechanical Engineers (ASME) V&V 10-2006, 2006.

8. Bayarri, M., Berger, J., Rui, P., Sacks, J., Cafeo, J., Cavendish, J., Lin, C.-H., and Tu, J., "A Framework for Validation of Computer Models," *Technometrics*, May 2007, Vol. 49, No. 2, pp. 138-154, 2005.

9. Babuska, I., Nobile, F., Tempone, R., "Reliability of Computational Science," *Numerical Methods for Partial Differential Equations*, Vol. 23, pp. 753-784, 2007.

10. American Society of Mechanical Engineers, *Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer*, American Society of Mechanical Engineers (ASME) V&V 20-2009, 2009.

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

12. Romero, V. J., "Comparison of Several Model Validation Conceptions against a 'Real Space' End-to-End Approach," *Society Automotive Engineers International Journal of Materials and Manufacturing*, June 2011.

13. Romero, V. J., "Elements of a Pragmatic Approach for dealing with Bias and Uncertainty in Experiments through Predictions: Data and Model Conditioning; 'Real Space' Model Validation and Conditioning; and Hierarchical Modeling and Extrapolative Prediction," Sandia National Laboratories report SAND2011-7342, Nov. 2011.

14. National Research Council, Assessing the Reliability of Complex Models: The Mathematical and Statistical Foundations of Verification, Validation, and Uncertainty Quantification, 2012. doi: 10.17226/13395

15. Romero, V., Rutherford, B., and Castro, J., "Model Calibration, Extrapolative Prediction, and Validation with Segregated Aleatory and Epistemic Uncertainties—Demonstration on Prototype Silicon Device," Sandia National Laboratories report (to be published).

16. Romero, V. J., Sherman, M. P., Dempsey, J. F., Johnson, J. D., Edwards, L. R., Chen, K. C., Baron, R. V., and King, C. F., "Development and Validation of a Component Failure Model," 45th AIAA/ASME/ ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Austin, TX paper AIAA-2005-2141, April 18-21, 2005.

17. Romero, V. J., "Validate Model? Not So Fast. The Need for Model 'Conditioning' as an Essential Addendum to Model Validation," 9th Non-Deterministic Approaches Conference, Honolulu, HI, paper AIAA-2007-1953, April 23-26, 2007.

18. Romero, V., Dempsey, J. F., and Antoun, B., "UQ and V&V Techniques applied to Experiments and Simulations of Heated Pipes Pressurized to Failure," Sandia National Laboratories report SAND2014-3985, May 2014.

19. Romero, V., Black, A., Breivik, N., Orient, G., Suo-Anttila, J., Antoun, B., and Dodd, A., "Advanced UQ and V&V Procedures applied to Thermal-Mechanical Response and Weld Failure in Heated Pressurizing Canisters," Sandia National Laboratories document SAND2015-3005C presented at the Society of Automotive Engineers 2015 World Congress, Detroit, MI, April 2123, 2015.

20. Rutherford, R., Romero, V., Castro, J., and Hoekstra, R., "Methods Used in Uncertainty Quantification, Model Calibration, and Prediction for the QASPR Silicon Device Prototype Demonstration," Sandia National Laboratories report SAND2011-7940, October 2011.

21. Romero, V. J., Luketa, A., and Sherman, M., "Application of a Versatile "Real Space" Validation Methodology to a Fire Model," *AIAA Journal of Thermophysics and Heat Transfer*, Vol. 24, No. 4, pp. 730-744, Oct.-Dec. 2010.

22. Ricks, A., Nicolette, V., Romero, V., and Erickson, W., "Fuego Solid-Propellant Fire Model Verification and Validation," Sandia National Laboratories report (to be published).

23. Romero, V., "The Real-Space Model Validation Approach as a (Unifying?) Extended Hybrid of the ASME VV10 and VV20 Approaches," Sandia National Laboratories document SAND2015-3752C presented at the American Society of Mechanical Engineers 2015 Verification and Validation Symposium, Las Vegas, NV, May 12-15, 2015.

24. Romero, V. J., "Type X and Y Errors and Data & Model Conditioning for Systematic Uncertainty in Model Calibration, Validation, and Extrapolation," presented at the Society of Automotive Engineers 2008 World Congress, Detroit, MI, SAE paper 2008-01-1368, April 14-17, 2008.

25. Chen, X., Zhanpeng, S., Qinshu, H., and Liu, X., "Prediction Considering Multi-Model and Model Form Uncertainty in the Parameter Space," Society of Automotive Engineers 2015 World Congress, Detroit, MI, paper 15IDM-0015, April 21-23, 2015.

26. Ferson, S., Oberkampf, W. L., and Ginzburg, L., "Model Validation and Predictive Capability for the Thermal Challenge Problem," *Computational Methods in Applied Mechanics and Engineering,* Vol. 197, pp. 2408–2430, 2009.

27. American Society of Mechanical Engineers, *An Illustration of the Concepts of Verification and Validation in Computational Solid Mechanics*, ASME Codes & Standards V&V 10.1-2012, April 16, 2012.

28. Box, G. E. P., Hunter, W. G., and Hunter, J. S., Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building, John Wiley & Sons, 1978.

29. Romero, V. J., "Model-Discretization Sizing and Calculation Verification for Multipoint Simulations over Large Parameter Spaces," 9th AIAA Non-Deterministic Methods Conference, Honolulu, HI, paper AIAA2007-1953, April 23–26, 2007.

Application of UQ and V&V to Experiments and Simulations of Heated Pipes Pressurized to Failure[1]

Vicente J. Romero[*], J. Franklin Dempsey[*], Bonnie Antoun[†]

[*]Sandia National Laboratories,[2] Albuquerque, NM 87815
[†]Sandia National Laboratories, Livermore, CA 94550

Overview

A case study involving experiments and model calibration, verification, validation, and uncertainty quantification associated with heated pipes pressurized to failure is presented in this chapter. The case study includes:

- Material strength experimental characterization and associated constitutive-model calibration.
- Model-assisted design and analysis of validation experiments to characterize and reduce uncertainty.
- Mesh and solver discretization studies (simulation verification) to control and characterize solution error and uncertainty.
- Model validation comparison of experimental and simulation results with uncertainties and interpretation of the results.

1. Introduction

The coupled thermal-mechanical application problem in this case study involves heated steel pipes pressurized to failure (see figures 1 and 2). The pipes are heated by a hot Inconel plate, creating a hotspot on each pipe. Each pipe is pressurized in a controlled manner until it bursts at its hot spot in a fairly repeatable manner, tearing back along upper and lower thickness-transition shoulders in the pipe and leaving "butterfly wings" as shown in Figure 1.

The driving purpose for this work is the validation assessment of a constitutive model for stainless-steel response, up to initiation of failure, in pressure vessels at high pressures and temperatures. The instantiation of the constitutive model involves simple pipe geometries [1]. The pipe is approximately 14 inches high and 3.5 in. diameter, with mid-region wall thickness of nominally 0.02 in., and upper and lower shoulder section thicknesses of nominally 0.05 in. For

[1] Sandia National Laboratories document SAND2014-4870P.

[2] Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

safety reasons, the inner slug (appears as shiny silver) fills up most of the inside volume of the pipe, lessening the pressurized volume of explosive energy built up prior to pipe breach failure. Only details of "pipe bomb" (PB) geometry and experimental conditions and results important to the model validation procedures and comparisons are presented here. Further details of the design of the hardware and experiments, and execution and results of a larger set of experiments with pipe geometries and various heating and pressurization conditions, are given in the work by Antoun [2].

The project required the formulation and development of an approach for including the significant temperature dependence of material strength (stress-strain response) of 304L stainless steel over the temperature range of interest, nominally 25 degrees Celsius (C) to 800°C. Information on the mathematical and algorithmic formulation of the temperature-dependent, multilinear, elastic-plastic (MLEP) constitutive model for material behavior is available from Wellman [3] and Scherzinger et al. [4]. The constitutive model was implemented in the Sierra solid mechanics code [5] (massively parallel 3-D implicit nonlinear quasi-statics) to model pipe response in the tests.

Population of the constitutive model required experimental characterization of stress-strain (s-s) behavior with tension tests of material samples at temperatures spanning the range of interest, as described in Section 2. Several replicate tests were performed with nominally identical material samples for each test, in order to characterize the effects of material variability on exhibited strength. For each measured s-s curve, optimization techniques were used to solve the inverse problem of obtaining best-fit transformed s-s curves for the constitutive model. The inverse problem and solution procedures and results are briefly described in Section 2 and are more fully documented in the work by Wellman [3].

For uncertainty quantification (UQ) purposes, it is important to note that each s-s curve comprises a *discrete* random function that has no readily identifiable parametric relationship to other s-s curves at that temperature. A novel UQ approach that operates from limited (sparse) numbers of replicate material tests and compensates for just a few test samples from the larger population of discrete random functions is described in Section 2.

Sections 3-5 describe the specific implementation of the generic verification, validation, and UQ procedural steps summarized in Section 4 of Chapter 10.

Section 3 describes the project's finite-element (FE) models, geometries, mesh and solver choices, and calculation verification studies to characterize discretization-related solution errors in both the constitutive-model material characterization/inversion procedure and the heated pressurized pipe simulations.

Section 4 describes the use of modeling and simulation to help design the PB validation experiments and thermocouple locations, in order to minimize errors and uncertainty associated with the experiments and modeling of the boundary conditions from spatially sparse sensor information.

Section 5 presents the PB validation experiments and simulations, their uncertainties, and processing of results and uncertainties for comparison within the "Real Space" (RS) model validation framework. Chapter 10 provides a brief introduction to some important concepts and considerations in model validation and the RS model validation paradigm, in particular. Appendices D and Chapter 10 discuss the features and capabilities that make the RS approach

more applicable to the challenging PB validation problem than other available validation frameworks.

Figure 1. Model validation experiments.

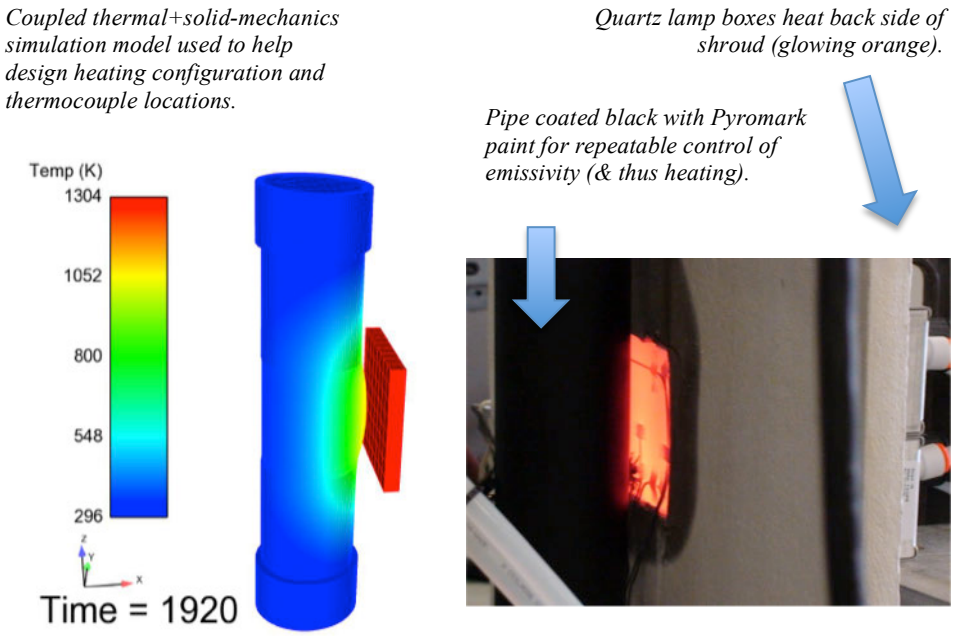


Figure 2. Thermal modeling helped the design, instrumentation, and analysis of experiments. Thermocouples with wire leads can be seen on the inconel heating shroud (glowing orange).

The RS framework segregates aleatory and epistemic uncertainties to form straightforward model validation comparisons especially suited for validation assessment of models to be used in the analysis of performance and safety margins. The methodology also handles challenges in the PB validation problem associated with representing and propagating interval and probabilistic uncertainties from multiple correlated and uncorrelated sources in the experiments and simulations including:

- Material variability characterized by non-parametric random functions (discrete temperature dependent s-s curves).
- Very limited (sparse) experimental data at the material testing and/or characterization level and at the pipe-test validation level.
- Boundary condition reconstruction uncertainties from spatially sparse sensor data.
- Normalization of pipe experimental responses for measured input-condition differences among multiple replicate tests and for random and systematic uncertainties in measurement/processing/inference of experimental inputs and outputs.
- Numerical solution uncertainty from model discretization and solver effects.

Section 5 culminates in the comparison of uncertainty ranges of experimental and predicted 2.5 and 97.5 percentiles of failure pressure. Analysis and interpretation of the results are provided, along with a presentation of the relatively sophisticated but inexpensive UQ spreadsheet procedures used to obtain the results. The RS validation and UQ approach can be leveraged to a large set of challenging model validation problems beyond the example in this chapter.

Section 6 closes with a brief summary of our main findings and conclusions.

2. Quantification and Propagation of Material Behavior Variability and Epistemic Uncertainty Associated with the Constitutive Model

2.1. Experimental Stress-Strain Variability of Tested Material Samples

Figure 3 shows elements of the rod tensile tests and results from which the s-s curves for the constitutive model were derived. Several replicate tests at each of the indicated temperatures were conducted, through necking and failure of the rod specimens.

The cylindrical material samples were machined from the same 304L stainless steel tubular stock (3.5 in. diameter, ¼ in. wall thickness) that the validation-experiment pipes were machined from. The rod specimens are 3 in. long and have a long thin "gage" section of diameter 1/8 in. (see Reference 2 for detailed drawings). The specimens were vacuum annealed at 1,000°C for 30 minutes to produce the same anneal conditions present in the pipe vessels. The ends of the rods were displaced in axially opposing directions to produce axial tension at an engineering strain rate of 0.001/s, which is slow enough to be considered quasi-static, and were estimated from hand calculations to be representative of pipe wall membrane strain rates in the PB validation tests. The measured axial displacement and resisting force were transformed to the experimental "engineering" stress-strain response curves plotted in Figure 3. More details of the testing apparatus, experimental conditions, measurement and control instrumentation and calibration, and so forth, are given in the work by Antoun [2].

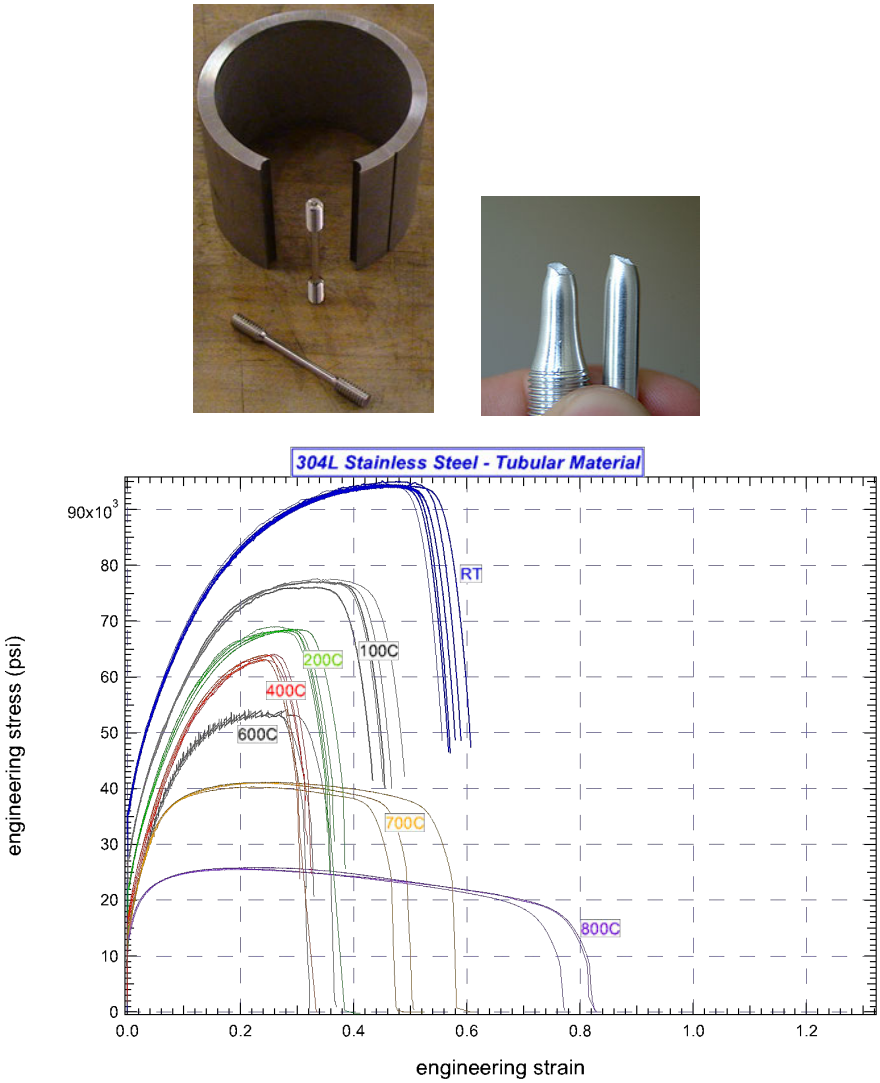


Figure 3. Rod tensile test material samples and measured stress-strain curves at the labeled temperatures. (Note: "RT" in the plot stands for "room temperature", nominally 20C).

2.2. Inverse Calculations to Convert From Measured Stress-Strain Curves to Constitutive Model Stress-Strain Curves

For each measured engineering s-s curve, optimization techniques were used to solve the inverse problem of determining the constitutive model's corresponding "true" s-s curve [3]. This enables an FE model of the rod to reproduce the measured engineering s-s response in the tension tests. The modeled rod deforms and necks when pulled to experimentally measured displacement and resisting force vs. time, in a manner that closely matches the test results. An example is presented in Figure 4.

The solid mechanics code Adagio [5] was used for the FE model forward simulations in the inversion procedure. In Section 3.1 we discuss the FE mesh and solver discretization choices. The model results are insensitive to significant perturbations from the model's discretization settings used in the inversion procedure. Therefore, the mesh and solver settings are presumed adequately refined for the purposes here.

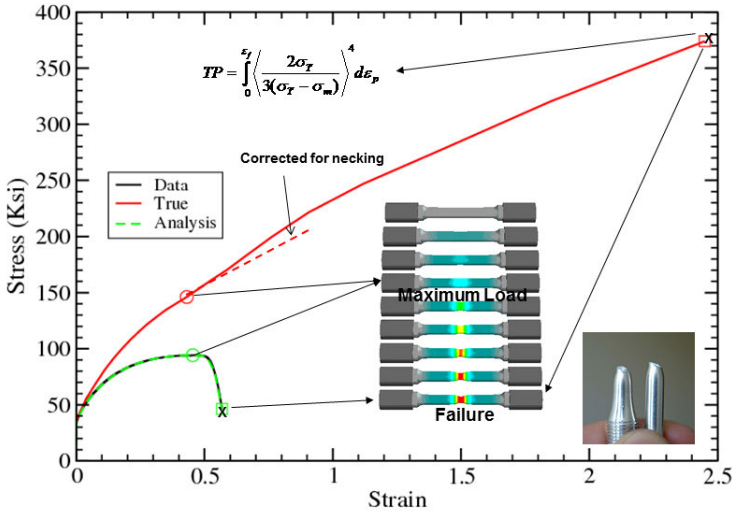


Figure 4. Relationship between experimental and constitutive-model stress strain curves obtained from a tensile test. The red curve is the Cauchy-Stress/Logarithmic-Strain "True-Stress/True-Strain" curve appropriate for the constitutive model. The black curve is the engineering-stress/engineering-strain curve from the test. The red curve is inversely calculated such that its use with the constitutive model in FE simulations of the cylinder tensile test yields the calculated green dashed s-s curve, which closely matches the experimental s-s curve.

The inversion procedure involves many steps and decision points in which analyst judgment is used. To check the sensitivity of the inversion results to different analysts applying the procedure we asked another seasoned solid mechanics analyst to try to reproduce one of the 700C True-Stress/True-Strain curves that was originally obtained by Wellman [3], who retired shortly after generating the material curves for our project. The other analyst was familiar with the inversion procedure from applying it for other projects; otherwise, this was a relatively independent spot-check on our derived material curves. The differences were insignificant below about 230% strain (see Reference 18), which far exceeds the point of pipe structural failure initiation in our calculations. We assume that the relevant portions of the other True-Stress/True-Strain curves in our project are also free of analyst variability effects in the material inversion process.

2.3. Incorporating Multiple Stress-Strain Curves of Material Variability and Accompanying Uncertainty from Small Numbers of Material Tests

The numbers of material variability tests (numbers of replicate s-s curves at each temperature) are relatively small, as seen in the plot in Figure 3. When only a few samples of a random variable or function are available, the variability properties of the full population of random values or functions generally cannot be accurately constructed from the few samples. Thus, substantial epistemic "sampling uncertainty" exists in addition to the aleatory uncertainty due to stochastic variability in the source population.

The likely error that accompanies sparse sampling has a bias toward underestimating the true full-population variance (at least for distribution types and combinations shown in Figure 7 and

investigated in the work by Romero, Swiler, Mullins, and Urbina [9-11]). This unconservative bias is undesirable for many engineering purposes. If a structure or pressure-vessel model were perfect in every other way, use of the constitutive model would likely underestimate the strength or displacement response variance of the real system. In design and risk analysis, one would normally want to avoid such underestimation.

Two approaches were tried to avoid underestimation of the larger-population variability. In the first approach, we assumed that the variability of s-s curves at a given temperature could be parameterized. Then, the parameter range corresponding to variations between the curves could be appropriately increased to correct for small-sample underestimation bias. The adjusted parameter range representing material curve variability could then be propagated through the pipe response model along with the other parametric uncertainties in the model. However, the attempted parametric representation of material curve variability failed an important "sanity test" for physical consistency as shown in Ref. 18.

Consequently, we devised and implemented a different approach, in which the uncertainty associated with material curve variability is decoupled from the other (parametric) uncertainties in the problem. The variability is represented and propagated alone, as described here. After propagation, it is combined with the other (propagated) parametric uncertainties, as described in Section 5.2.

This approach treats the s-s curves at a given temperature as discrete random functions with no readily identifiable parametric relationship between them, but also recognizes that the s-s curves issue from the same temperature-characteristic population of discrete random functions. Furthermore, because only a small number of experimental curves of behavior are usually available, the approach mitigates the chances of underestimating the full-population variability with relatively few data samples. Figure 5 conveys the approach for a random variable, but the idea also applies to random functions.

Employing the approach shown at right in Figure 5, the multiple s-s curves at a given temperature are individually propagated through the applicable system model (here, the PB model) to yield corresponding samples of output quantities of response, such as displacement and failure pressure. Tolerance Intervals of the response quantities are then constructed from the response samples for further analysis purposes, as explained below.

This approach also accommodates the propagation of model parameter sets that are discrete (that is, not parametrically continuous). For instance, in electronics modeling applications the Gummel-Poon (GP) model parameters (often 10 or more) are determined unique to each particular device tested. When multiple repeat tests on nominally identical devices are performed, the resulting sets of GP parameters define different points in the parameter space. However, the parameter space is generally not considered to be continuous, e.g., as discussed by Kambour, Hembree, and Keiter [12]. The model is generally not "trusted" to yield suitable results when run with parameter values at other points in the space, e.g., at points interpolated on a line between any two established points (parameter sets) in the space. Therefore, device-to-device and other experimental variability resident in the discrete GP parameter sets can be treated by the paradigm illustrated at right in Figure 5.

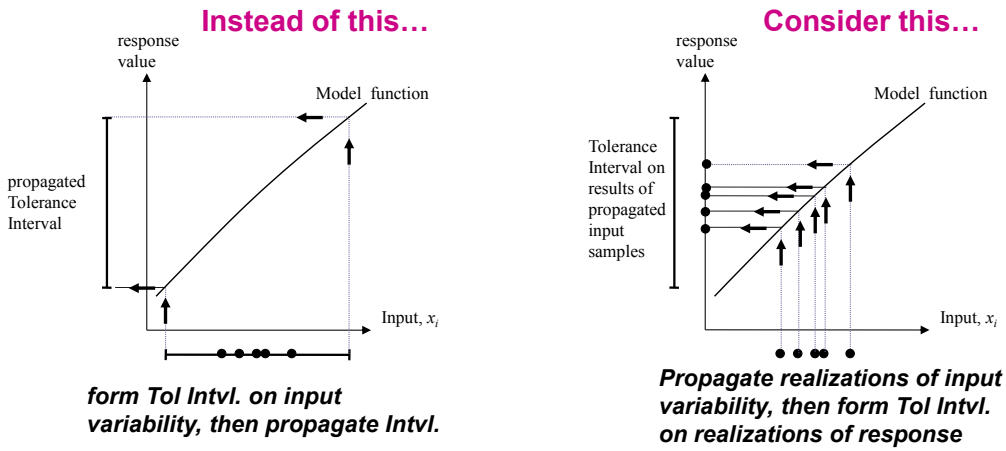


Figure 5. Two ways of propagating uncertainty from sparse samples of an input random variable. In the approach shown at left, the sparse samples of the input quantity are fit with a tolerance interval that is then propagated to an uncertainty of response. Alternatively, an equivalent Normal probability density function (explained below in connection with Figure 8) is fitted to the tolerance interval and propagated. The approach shown at right individually propagates each sample of the input quantity and then forms a tolerance interval (and perhaps an equivalent Normal PDF) from the propagated results.

Consider the samples of output response on the vertical axis at right in the figure. Various approaches can be taken to compensate for small numbers of samples. Investigations were undertaken, as reported in Refs. 9-11. It was found that a classical statistical tolerance interval (TI) approach, e.g., in Reference 13, provides reliably conservative estimates of the combined epistemic and aleatory uncertainty associated with limited data. The approach is also very easy to use. A different approach [14] also worked well for sparse samples in many cases, but is somewhat more involved to implement and its performance remains to be broadly tested and characterized because the method is very new. The rather common practice of simply fitting the random data with a Normal distribution was found to have significant risk of under-estimating the true variability of the population being sampled, especially if the sampled distribution is a Normal distribution.

For our purposes, the TI approach was used. The length of tolerance intervals accounts for both the epistemic and aleatory elements of uncertainty due to the limited data samples. Hence, TIs are characterized by two user-prescribed attainment levels: one for *coverage* of a subset of the variability, and one for *statistical confidence* in covering or bounding at least that subset of variability. For instance, a 0.95-coverage/0.90-confidence TI prescribes lower and upper values of a range of response that is said to have at least 90% odds that it covers or spans the 0.025 and 0.975 percentiles of the "true" PDF from which the random samples were drawn, for a large array of PDF types.

A 0.95/0.90 TI is constructed by multiplying the calculated standard deviation $\tilde{\sigma}$ of the data samples by the factors f in Table 1 to create an interval of total length $2f\tilde{\sigma}$, where the interval is centered about the calculated mean $\tilde{\mu}$ of the samples. Table 1 and Figure 6 reveal that the TI size decreases quickly with the number of data samples. For 0.95/0.90 TI, a knee in the rate of uncertainty decrease per added sample occurs somewhere between 4 to 6 samples, with the rate of decrease being fairly small after 8 samples. The tolerance interval has an asymptotic standard-deviation multiplier of 1.96 for an infinite number of samples. This gives a TI that corresponds to the exact 0.95 central percentile range of a normal PDF with μ and σ the same as $\tilde{\mu}$ and $\tilde{\sigma}$ from

the ∞ samples. That is, the multipliers f for 0.95/0.90 TI are effectively constructed from randomly sampling a Normal PDF a number of times M in each of a large number of random trials, and finding the multiplier f_M that gives TI, which, in approximately 90% of the trials, spans the true generating normal PDF's 0.025 to 0.975 percentile range.

Table 1. 0.95/0.90 tolerance interval factors (standard deviation multipliers), vs. the number of samples of random quantity (selected results from tables from Montgomery and Runger [13]).

# samples	$f_{0.95/0.90}$
2	18.80
3	6.92
4	4.94
5	4.15
6	3.72
8	3.26
12	2.86
20	2.56
30	2.41
40	2.33
∞	1.96

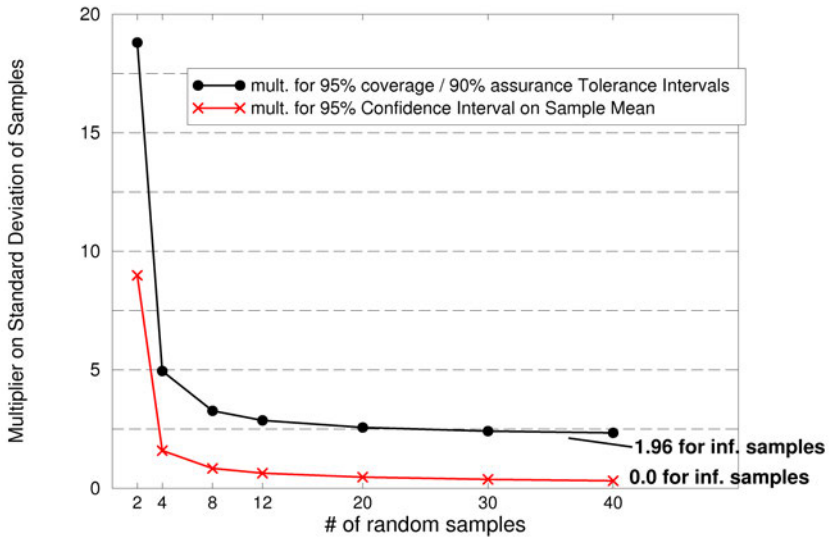


Figure 6. Multiplier on calculated standard deviation used to form 0.95/0.90 tolerance interval ranges vs. number of random samples. (Figure reproduced from Romero, Rutherford, and Newcomer [15], ignore confidence interval curve.)

Although constructed with respect to Normal PDFs, 0.95/0.90 TI will also span, with approximately 90% odds, the 0.025 to 0.975 percentile ranges of many other PDF types when sparsely sampled. This has been empirically established by Romero et al. [10, 11] for uniform and right-triangular PDFs and for PDFs resulting from convolving various types of PDFs, as depicted in Figure 7.

For further uncertainty representation and analysis purposes, a normal PDF is constructed such that its 0.025 and 0.975 percentiles coincide with the end points of the established 0.95/0.90 Tolerance Interval (e.g., Figure 8). Therefore, the normal PDF has approximately 90% odds that its 0.025 and 0.975 percentiles contain the 0.025 and 0.975 percentiles of the true PDF from which the random samples come (for a large array of PDF types). Furthermore, because the constructed PDF is normal, its characteristically long tails will have extended percentiles like 0.01 and 0.99 that, in most cases, extend beyond the same percentiles of the true PDF from which the data samples come. In References 9-11, this was found to be true for all tested PDF types and combinations shown in Figure 7.

The TI approach described here is presumed likely (at the said odds) to exaggerate the effects of the actual material variability. Such exaggeration can sometimes be egregious when very few samples are involved, depending on the particular samples obtained; see Refs. 9-11. The Pradlwarter-Schueller approach [14] has much smaller chances of egregious conservatism but averaged only 70% reliable in bracketing the true 0.025 to 0.975 percentile ranges of the PDF shapes and combinations shown in Figure 7 (compared to an average of 92% reliability for TIs).

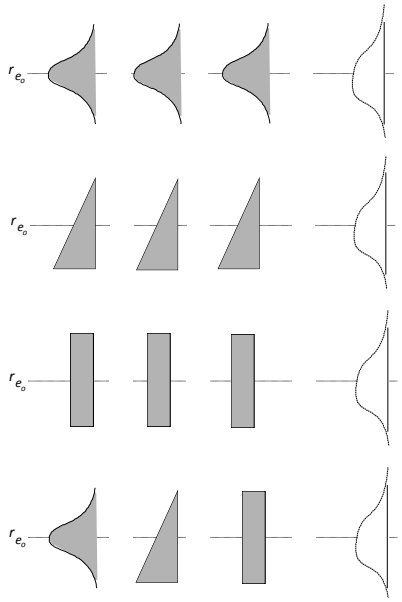


Figure 7. Test matrix for PDF representation study from Romero et al. [10, 11].

2.4. Results for Constant-Temperature Pressurized Pipes

At each value of material characterization temperature shown in Figure 3, the derived constitutive model s-s curves are used in simulations of the pipe vessel held at that temperature and pressurized at a linear ramp rate of 1psi/sec until pipe wall failure is indicated. The rate of linear pressure rise is representative of the PB tests but does not impact these calculated quasi-static

failure pressures; creep is not an aspect of the current constitutive model and is assumed to be unimportant in the PB tests, based on a test for creep effects in Ref. 16. The pipe model, mesh and solver settings, pipe ends fixturing/loading conditions, simulations, and failure criteria are described in Section 3.2. Table 2 gives the calculated failure pressures for the simulation runs at the various characterization temperatures. The results at a given temperature are listed in order of predicted failure pressures, and this orders the effective strengths of the s-s curves at that temperature. Note that because structural response depends on the path that a given s-s curve entails, it is not clear *a priori* how the curves rank in effective strength in a given application until application-model simulations are run.

For the six predicted failure pressures at 20°C in Table 2, the mean is 1482 psi and the standard deviation σ is 3.97 psi. Table 3 lists these summary statistics and those for the other characterized temperatures. As indicated in the table, the upper and lower extents of the 0.95/0.9 TIs are determined by adding/subtracting the quantity $f\sigma$ to/from the mean failure pressure, where the appropriate values of f come from Table 1. The tolerance intervals are plotted in Figure 8.

2.5. Extension to Other Application Conditions

Later validation simulations with the PB model employ different initial and boundary conditions than those underlying the TIs in Figure 8. Differences include wall thickness, fixturing/loading conditions on the pipe ends, pressure loading history, and non-uniform pipe temperature. A two-part approach explained in the following two sections is used to represent PB failure pressure variability under these different initial and boundary conditions.

2.5.1. Temperature Dependence of Material Variability, Parameterized by High and Low Strength Material Curves

Consider a pipe simulation with the same pressure and end-loading conditions underlying the results shown in Figure 8, but let the pipe have a spatially uniform temperature that increases in time. Let the simulation start at the first characterization temperature (20°C) in the material data set. When the pipe temperature rises from 20°C to 100°C, the s-s curves characterized at 20C gradually become less applicable and the s-s curves at the next characterization temperature of 100C gradually become more applicable until at 100°C they are exclusively the applicable curves. This prompts a generalized question of how to best (best balance of effectiveness and economics) transition between two sets of s-s curves from two adjacent characterization temperatures that bracket the local pipe temperature. In general, how does the simulation model represent material constitutive behavior for the local pointwise pipe temperature (in time and space) that lies between adjacent characterization temperatures at which s-s curves exist?

If a single s-s curve exists at each bracketing temperature, then a linear interpolation procedure is used to interpolate the s-s state from the bracketing s-s curves, given the local temperature and the bracketing curve temperatures. However, when multiple s-s curves of material variability exist at each bracketing temperature, it must be decided how to handle this. The issues and our approach are discussed next.

Table 2. Predicted pipe failure pressures (sorted from highest to lowest) at listed temperatures where multiple replicate stress-strain curves were characterized.

Test and Temperature (degrees C)	**Fail Pressure** (psi)
try5-20	1485.2
try6-20	1485.0
try3-20	1484.5
try39-20	1483.9
try4-20	1482.8
try40-20	1474.8
try14-100	1227.1
try36-100	1226.3
try16-100	1225.3
try37-100	1222.9
try15-100	1208.7
try11-200	1102.1
try34-200	1089.9
try13-200	1088.6
try12-200	1085.8
try35-200	1081.7
try33-400	1014.0
try17-400	1010.3
try18-400	1007.2
try19-400	1005.7
try32-400	1001.9
try24-600	884.7
try23-600	880.1
try22-600	869.2
try25-700	714.0
try27-700	704.2
try26-700	703.7
try31-800	448.8
try29-800	448.0
try30-800	440.8

Table 3. Statistics of predicted failure pressures at temperatures where replicate stress-strain curves were characterized.

Temperature	Mean, μ fail press. (psi)	Std. dev., σ (psi)	Upper & lower extents of 0.95/0.9 TI	Upper & lower extents of 0.95/0.9 TI (psi)
20C	1482.70	3.97 = 0.27% of mean	μ + 3.72σ = μ + 1% =	1497.5
			μ - 3.72σ = μ - 1% =	1467.9
100C	1222.06	7.63 = 0.63% of mean	μ + 4.15σ = μ + 2.6% =	1253.7
			μ - 4.15σ = μ - 2.6% =	1190.4
200C	1089.62	7.65 = 0.7% of mean	μ + 4.15σ = μ + 2.9% =	1121.4
			μ - 4.15σ = μ - 2.9% =	1057.9
400C	1007.82	4.59 = 0.46% of mean	μ + 4.15σ = μ + 1.9% =	1026.9
			μ - 4.15σ = μ - 1.9% =	988.8
600C	878.00	7.96 = 0.91% of mean	μ + 6.92σ = μ + 6.3% =	933.1
			μ - 6.92σ = μ - 6.3% =	822.9
700C	707.30	5.81 = 0.82% of mean	μ + 6.92σ = μ + 5.7% =	747.5
			μ - 6.92σ = μ - 5.7% =	667.1
800C	445.87	4.41 = 0.99% of mean	μ + 6.92σ = μ + 6.8% =	476.4
			μ - 6.92σ = μ - 6.8% =	415.4

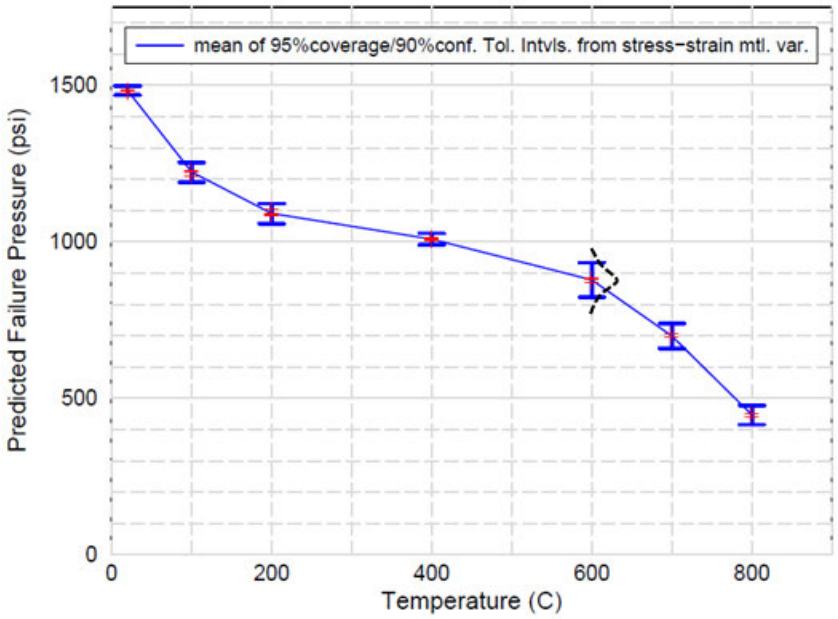


Figure 8. This graph depicts 95% coverage/90% confidence tolerance intervals from variations of pipe failure pressures at tested pipe temperatures (for uniform temperature throughout the pipe). Individual failure pressures predicted with the multiple replicate stress-strain curves are plotted as red crosses. Depicted at 600°C is an illustrative Normal PDF constructed from the 0.95/0.9 TI.

Consider a PB simulation with a spatially uniform temperature pipe that increases from 20°C to 200°C over time. Figure 9 is an illustrative representation (not to scale) of the TIs and underlying PB failure-pressure data points for the first three material characterization temperatures in Table 2 and Figure 8. Each dot on each TI in Figure 9 corresponds to a particular s-s curve. The temperature transient in the simulation can be negotiated with linear interpolation after selecting one dot/s-s curve at each of the relevant characterization temperatures, 20°C, 100°C, and 200°C. Several possible selection combinations are shown in Figure 9. A vast number of other combinations exist. All combinations could be sampled (exhaustive sampling), and a PB simulation could be run for each combination. This would amount to 6x5x5 = 150 runs of the PB model. With the full set of material data there would be 20,250 combinations/runs for a simulation involving a uniform pipe temperature that increases from 20°C to 800°C. This is clearly unaffordable and is not necessary in the present circumstances. Alternatively, a random sampling of, say, 30 to 50 random combinations could be performed. This might be affordable in some circumstances, but not for the validation simulations to be described in Section 5. Thus, the following two-run approach was taken in an effort to bound things.

Consider a local region of material on the pipe. At a given characterization temperature, say 20°C, this local region has particular s-s properties with an effective strength variability reflected by the six data samples at 20°C in Figure 9. Now consider a case where the effective strength of the local material lies near the highest dot at 20°C. If the local material region undergoes a temperature transition from 20°C to 200°C, for example, it is physically plausible[3] that a local material region with a high relative effective strength at the starting temperature 20°C will retain high relative effective strength as it transitions temperatures to 200°C. Thus, a material realization with effective strength that starts in the neighborhood of the highest-strength dot at 20°C will tend to correlate with the highest-strength dots at 100°C and 200°C. So the high-strength (HS) combination path shown in Figure 9 is highly reasonable physically. Similar reasoning is applied to the lowest dots at 20°C, 100°C, and 200°C (low strength (LS) path).

Material realizations corresponding to mid-range dots at 20°C in Figure 9 will also tend to correlate with the mid-range dots at 100°C and 200°C. The PB response in simulations using mid-range path combinations is assumed to be bounded by the consistently HS and consistently LS combinations. Therefore, computational resources are not expended investigating moderate paths.

We now discuss spatial material variability in the pipe. Figure 10 illustrates some possible combinations at three neighboring (> millimeters apart) locations on the pipe at uniform temperature 20°C. Any of an exhaustive number of possible combinations would seem to be equally likely. To bound this problem we assume that PB simulation results from the consistently HS and consistently LS combinations in Figure 10 would bound the simulation results from any other possible combinations like the others shown in Figure 10. Due to lack of time and resources we have not verified this assumption, but recommend that it be assessed in future projects that would seek to rely on this assumption.

[3] This assumes that material weakening mechanisms and percentage of weakening are roughly similar with increasing temperature whether the material is initially of higher, medium, or lower relative strength.

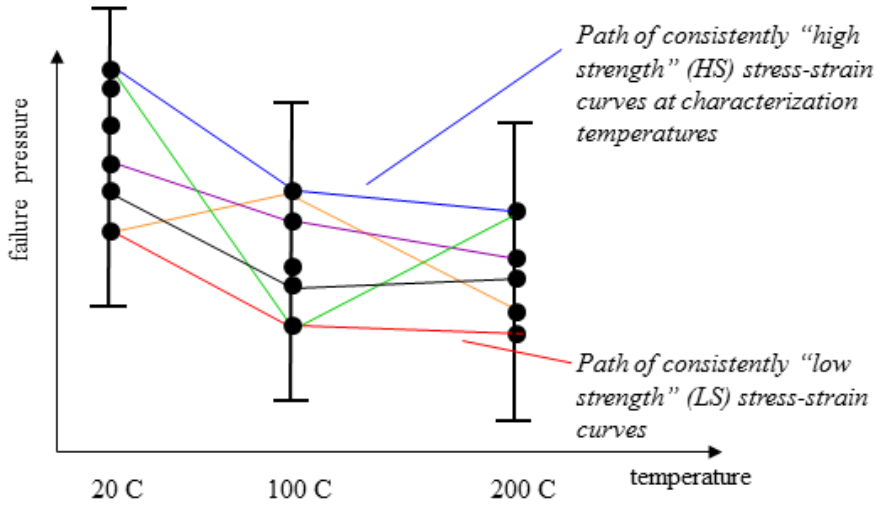


Figure 9. Illustrative PB failure pressure data (for uniform pipe temperature), and corresponding 0.95/0.9 tolerance intervals (not to scale), for the first three material characterization temperatures in Table 2 and Figure 8. Several possible combinations or paths of material stress-strain curves are shown that could be used in a computational simulation to negotiate a temperature transition over the depicted temperature range.

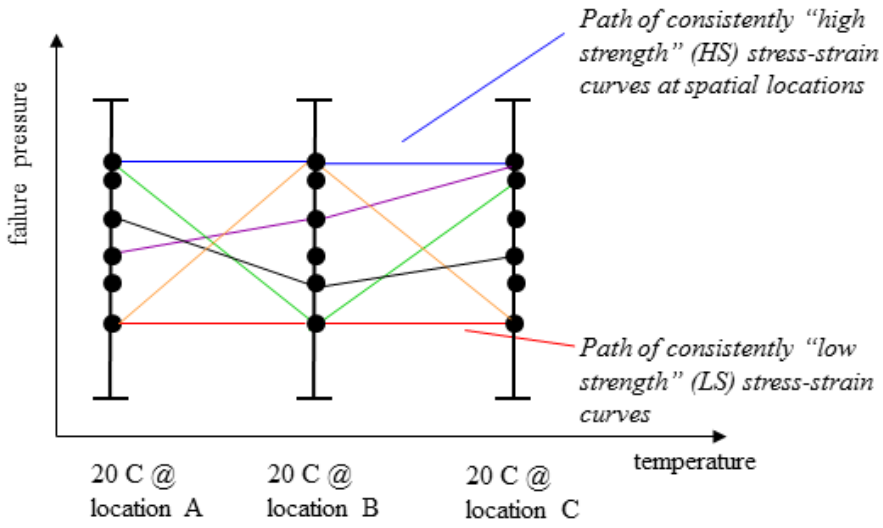


Figure 10. Illustrative PB failure pressure data and associated 0.95/0.9 tolerance intervals for three neighboring locations on the pipe at uniform temperature 20°C. Several possible stress-strain curves exist at each of three sample locations on the pipe. Some possible combinations of the stress-strain curves at these locations are shown that could be used in a computational simulation of a uniform-temperature 20°C pipe. However, only the high-strength and low-strength combinations are used in the present work.

We use similar reasoning for steady or transient spatially non-uniform temperature fields. Consider a spatial temperature transition between 20°C and 200°C at two locations on the pipe. Figure 9 applies to this spatial temperature transition (and not just to the temporal transition previously considered with the figure). For spatial temperature transitions, strong correlation of ss-curve strengths is not expected between different locations because of physical material variations that can occur from point to point. Therefore, any ss-curve computational path

combinations, such as all those shown in Figure 9, are physically plausible. Again, we assume that the consistently HS and consistently LS combinations bound the simulation results from any other possible combinations. But we acknowledge that a spatially stochastic uncertainty representation would be relevant here and may even be necessary for certain analysis requirements. A follow-on investigation is recommended. We conclude similarly for spatially varying pipe wall thickness in Section 5.3.

Thus, in view of the constraints in this project, material variability effects are parameterized in terms of just two simulation runs of the PB model. One simulation uses only the high strength s-s curves at each characterization temperature, using linear interpolation to transition between the high-strength s-s curves. The other simulation uses only the low strength s-s curves at each characterization temperature. Next we explain how this parameterization is used in Section 5 to scale PB failure pressure variability TIs in Figure 8 into TIs for the validation application conditions.

2.5.2. Scaling of PB Failure Pressure Variability to New Application Conditions

Let the HS and LS simulations described above yield the respective failure pressures P_fail-HS and P_fail-LS. These define a range of predicted failure pressure given by:

$$\Delta = \text{P\_fail-HS} - \text{P\_fail-LS}. \tag{2.1}$$

For the case of a pipe at uniform temperature 700°C, the range $\Delta_{700\text{Cunif}} = \text{P\_fail-HS}_{700\text{Cunif}} - \text{P\_fail-LS}_{700\text{Cunif}}$ can be determined from the results in Table 2. This range is labeled in Figure 11 on the left TI, which corresponds to the 700°C uniform-temperature PB tolerance interval defined in Table 3. The 700°C-uniform TI is composed of three segments: $L_{\text{lower-700Cunif}}$, $\Delta_{700\text{Cunif}}$, and $L_{\text{upper-700Cunif}}$, as shown in Figure 11.

The separation of the TI into three segments according to the HS and LS simulation results is handy for scaling the TI to other application conditions, as illustrated in Figure 11. For example, let $\text{TI}_{\text{app700C}}$, as shown at right in Figure 11, be for a non-uniform temperature pipe where failure occurs at the pipe hot spot with temperature 700°C at its peak. In validation simulations with the PB model, several conditions will differ from those Figure 8 is based on, such as pipe wall thickness, pressure loading history, non-constant temperature, and end fixturing conditions on the pipe. Even if the failure point is at 700°C at the time of failure, the calculated HS and LS failure pressures will differ (perhaps greatly) from the values $\text{P\_fail-HS}_{700\text{Cunif}}$ and $\text{P\_fail-LS}_{700\text{Cunif}}$ in Table 2 for the 700°C uniform-temperature pipe. But, the spans $\Delta_{700\text{Cunif}}$ and Δ_{app700C} in Figure 11 both issue from the same sets of low-strength and high-strength s-s curves. Therefore the spans Δ_{app700C} and $\Delta_{700\text{Cunif}}$ scale with each other. The scaling factor is:

$$S = \Delta_{\text{app700C}} / \Delta_{700\text{Cunif}}. \tag{2.2}$$

Everything on the right side of Eq. 2.2 can be obtained from Table 2 and the HS and LS simulation results under the new application conditions. The lengths of the other segments that make up the TIs in Figure 11 are—assumed to scale similarly. Mirroring Eq. 2.2, $S = L_{\text{upper-app700C}} / L_{\text{upper-700Cunif}}$ — or:

$$L_{\text{upper-app700C}} = S \times L_{\text{upper-700Cunif}}. \tag{2.3}$$

Analogously:

$$L_{\text{lower-app700C}} = S \times L_{\text{lower-700Cunif}}. \tag{2.4}$$

Then $\text{TI}_{\text{app700C}}$ is fully defined by the three known segments: $L_{\text{lower- app700C}}$, Δ_{app700C}, and $L_{\text{upper-app700C}}$, as shown in Figure 11. Tolerance intervals constructed this way come from a loose but traceable basis of 0.95-coverage/0.90-confidence TIs at reasonably nearby points in the problem parameter space. Because of several assumptions made in the construction, including interpolation and extrapolation (see next paragraph), it cannot be assured that the constructed TIs present conservative 95% ranges of failure pressure variation due to the underlying material property variability. Nonetheless, we consider the constructed TIs to be reasonable working estimates and use their lower and upper ends for 0.025 and 0.975 percentiles of constructed normal PDFs of failure pressure variability in the new application, as indicated in Figure 11.

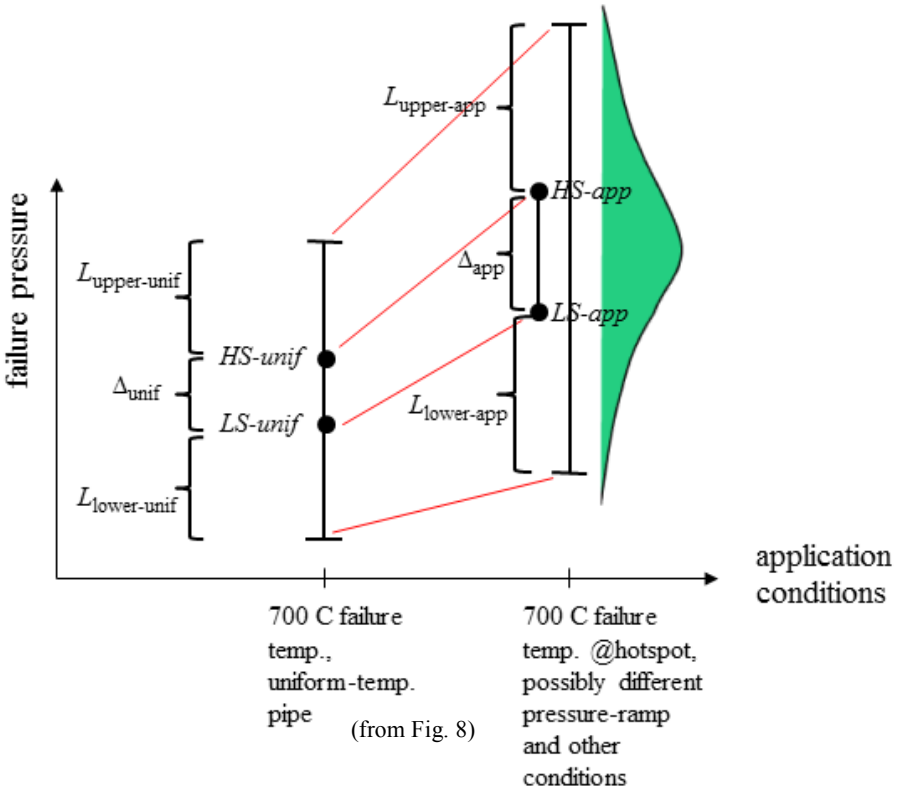


Figure 11. Proposed scaling between 0.95/0.9 tolerance intervals parameterized by failure pressures calculated with low- and high-strength material curves for (at left) a uniform-temperature pipe at 700°C and (at right) a non-uniform temperature pipe where failure occurs at pipe hot spot with temperature 700°C.

This procedure can be applied to the general problem, where the failure temperature is some value T between the specific characterization temperatures in Figure 8. Then, the subscript 700°C in the above equations is replaced by the applicable temperature T. The span $\Delta_{T\text{unif}}$ in the new version of Eq. 2.2 is linearly interpolated from the spans at the immediately surrounding characterization temperatures given in Figure 8. The upper and lower segment lengths $L_{\text{upper-}T\text{unif}}$ and $L_{\text{lower- }T\text{unif}}$ are likewise interpolated.

Note that in the PB validation simulations, which involve both spatially and temporally varying temperature fields, the HS and LS predicted failures generally do not occur at exactly the same location or at exactly the same temperature. But failure locations and temperatures are within tenths of a percent for the HS and LS simulations, so we used the average predicted failure temperature Tavg-app in the above procedure, where the applicable subscripts are related by Tunif $= T$avg-app.

3. FE Models, Geometries, Mesh and Solver Choices, and Calculation Verification

3.1. Rod Tension Test Simulations

We now consider the finite element (FE) model and simulations of the cylindrical test specimens in the constitutive model material characterization/inversion procedure described in Section 2.2. Given the geometry and testing conditions described in Section 2.1, the simulations emulate rod response through necking and failure, as depicted in Figure 4.

The geometry and mesh of the corresponding FE model are shown in Figure 12. We model only a 1-in. middle portion of the gage length, where the loading and s-s response is considered to be axisymmetric. The FE model consists of only about 2.77% of this 1-in. middle section of gage length, invoking symmetry to model a 20-degree portion and only the top half, as shown in the figure. The model's radial dimension is 1/16 (0.0625) in. at the top cross-section, varying linearly to 0.0620 in. at the bottom. The bottom cross-section is modeled with slightly less than actual radius so that computed failure will dependably occur at this minimum cross-section location.

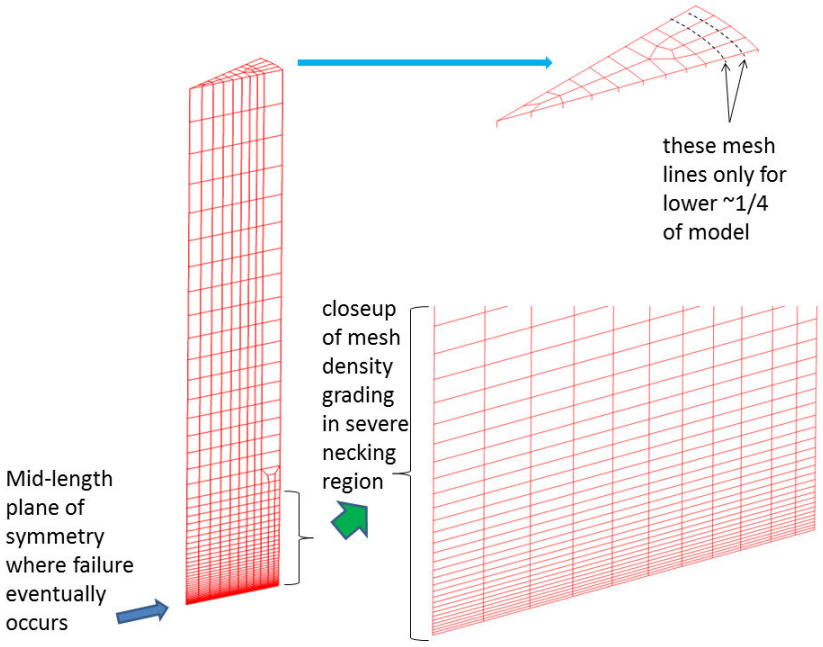


Figure 12. FE model geometry and mesh for a modeled section of circular rod in tension tests.

By seeding the location of most deformation (and failure) in this way, a mesh can be tailored for the simulations. Linear 8-node hexagonal finite elements are used. The mesh is most dense in the severe necking region. In this region the mesh is graded so that grid density increases substantially where the grid approaches the mid-length plane of symmetry. Necking entails

substantial axial stretching and radial compression of elements. To compensate, the initial mesh that the simulations start from is made with axially compressed and radially stretched elements, as shown. The mesh resolution appears to be adequate when compared to the much coarser meshes in a mesh sensitivity study by Romero, Dempsey, and Antoun [18] for a similar material inversion problem that shows no mesh effects for the relevant portion of the derived s-s curves.

The solver and hourglass-control settings for the FE calculations in the material inversion procedure are also catalogued in Ref. 18. These were chosen by Wellman [3] based on years of experience performing similar material inversion procedures for similar types of rod tensile tests. The settings were also deemed adequate by a second analyst with similar experience during a spot-check of reproducibility of the inversion procedure and results (see Section 2.2).

3.2. Isothermal Pressurized Pipe Simulations at the Material Characterization Temperatures

Isothermal PB simulations were performed to calculate the pipe failure pressures given in Table 2. When the uniform-temperature pipes are pressurized to failure, they reach a state in which the material can no longer resist the next increment in applied pressure. This is termed the "structural instability" point [17]. In the PB simulations, this coincides with a mathematical instability, signified by the Adagio calculation going numerically unstable (that is, unable to converge [17]). The instability occurs because the Adagio quasi-static governing equations have no inertial terms to balance the increment in pressure force by an acceleration of the material when its strength can no longer resist the internal pressure—and the pipe bursts.

Identifying the burst pressure by arriving at the quasi-static instability point is sufficient for the purposes here. If one wants to calculate the ensuing structural breakup of the pipe, a structural dynamics code can be used. It has the required inertial terms for continuation of the simulation from a starting point just prior to quasi-static instability and continuing through structure breakup. Such continuation was demonstrated by Wellman [3].

Figure 13 shows representative Adagio calculation results associated with Table 2. A signature difficulty in these calculations is the large computational expense required to creep up to the structural instability point. As this point is neared, the solver must work harder to advance by taking smaller timesteps and performing more iterations per timestep to solve the increasingly nonlinear material response and force-balance problem.

The large majority of CPU time is spent in the final stages of the calculation. Alternate approaches are being investigated to more efficiently identify the structural/mathematical instability point or to identify alternative indicators of material damage and critical failure values. The latter approach would ideally suspend the calculation at a point simultaneously low on the CPU cost curve and near the final failure pressure. Such a point occurs, for example, at the sharp knee in the curve of the lower left CPU plot in Figure 13. This point is very near the ending (instability) time in the simulation, and because pressure is linearly ramped with time in these simulations, the knee also corresponds to a pressure that is very near the final pressure at calculation failure.

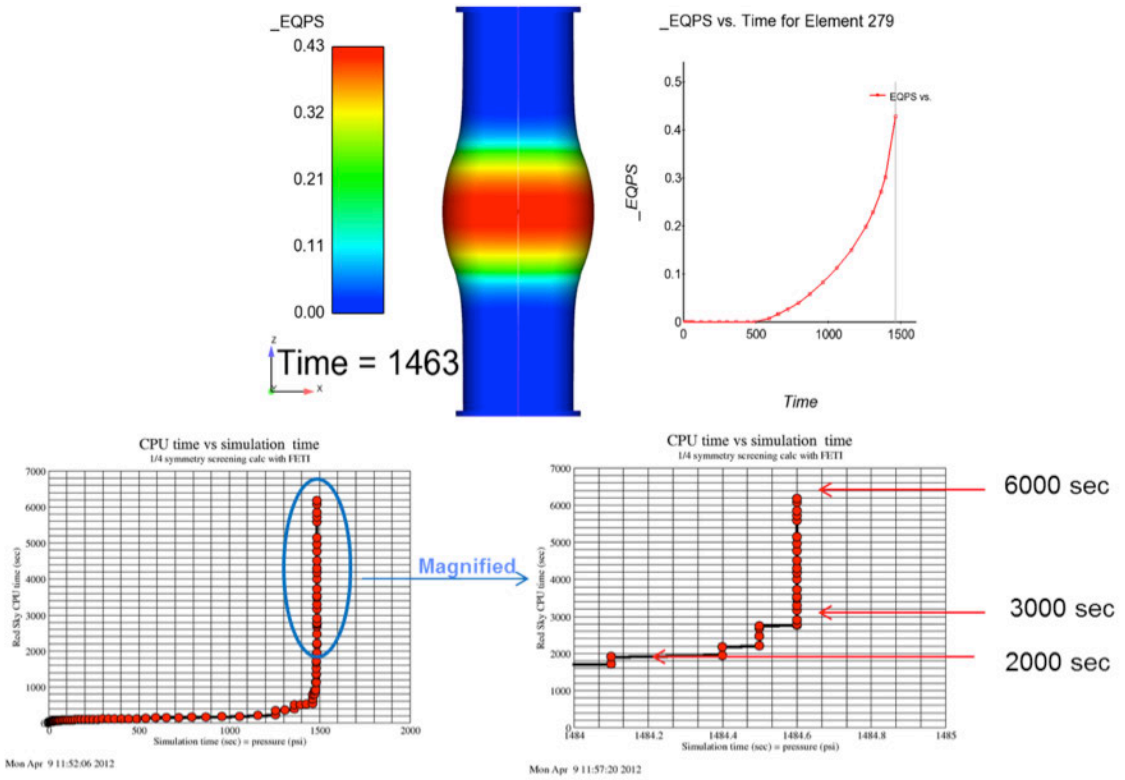


Figure 13. Representative Adagio calculation results associated with Table 2 and a ¼ symmetry model. EQPS is the equivalent plastic strain of the material.

3.2.1. Geometry and Mesh

The FE model has a pipe wall thickness of 0.02 in. and four finite elements through the thickness of the wall. Mechanical end-loading conditions are fixed. The full-geometry model is shown at left in Figure 14 (see Ref. 2 for detailed drawings). For the calculations in section 3.3, a geometry change was made in the modeled length of the pipe. The full length of the PB cylinder is 14 inches. The truncated model shown at right in Figure 14 contains only the middle 10.8-in. section of the pipe. Most of the thick-walled shoulder sections at either end are truncated because previous simulations found they are so strong relative to the thin-wall sections of the pipe that truncating them does not materially affect failure pressure results. This reduces the finite-element count by about 30%. To further save on computational resources, a 1/8-symmetry portion of the truncated geometry was used, which takes advantage of the symmetry of geometry and uniform-temperature and fixed-end boundary conditions in the problem. See figures 15 and 16 for details on the mesh structure and element aspect ratios.

Later, in Section 3.4, we investigate the effects of different mesh densities and element aspect ratios for simulations of the spot-heated PB experiments. From that investigation, we infer that the mesh in figures 15 and 16 appears quite adequate for simulations of the uniform-temperature pressurized pipes described in this section. These simulations are used to calculate the failure pressures in Table 2.

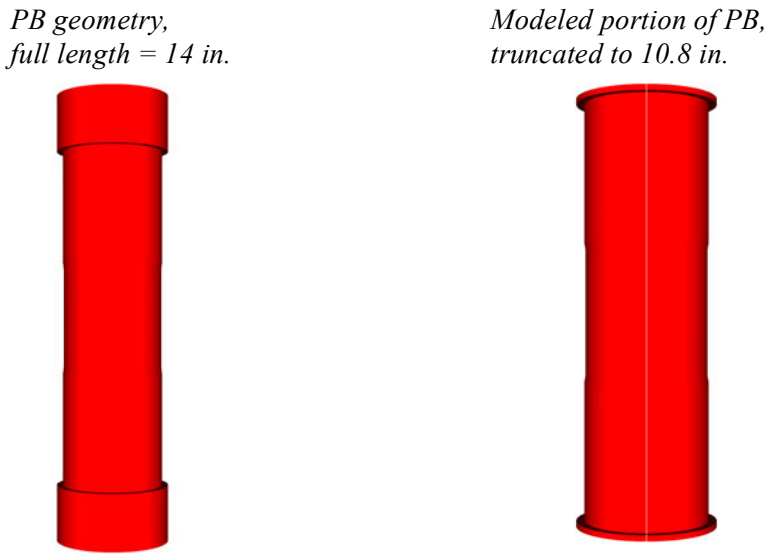


Figure 14. Actual and truncated-model pipe geometries. The full length of pipe is 14 inches. The modeled portion omits 1.6 inches from each end of the pipe, leaving a ¼ inch length of stiffening collar at the pipe ends. This decreases the finite-element count by almost 30%.

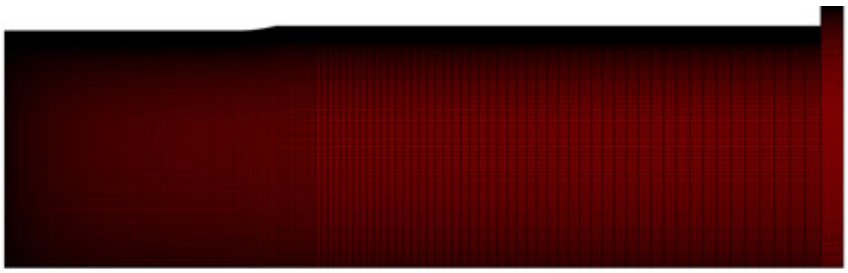


Figure 15. Finite element mesh of the 1/8 symmetry model of truncated pipe section shown in Figure 14. The 1/8 symmetry model contains 200,322 8-node linear hex elements. Grid density increases significantly in going from the truncated end of pipe to the mid-length of pipe. Grid density also increases in going from back of pipe to front of pipe. (The front-to-back grid density variations were imposed in anticipation of simulating scenarios with the hotspot from non-uniform heating conditions in the PB experiments, but different meshes were ultimately used; see Section 3.5.)

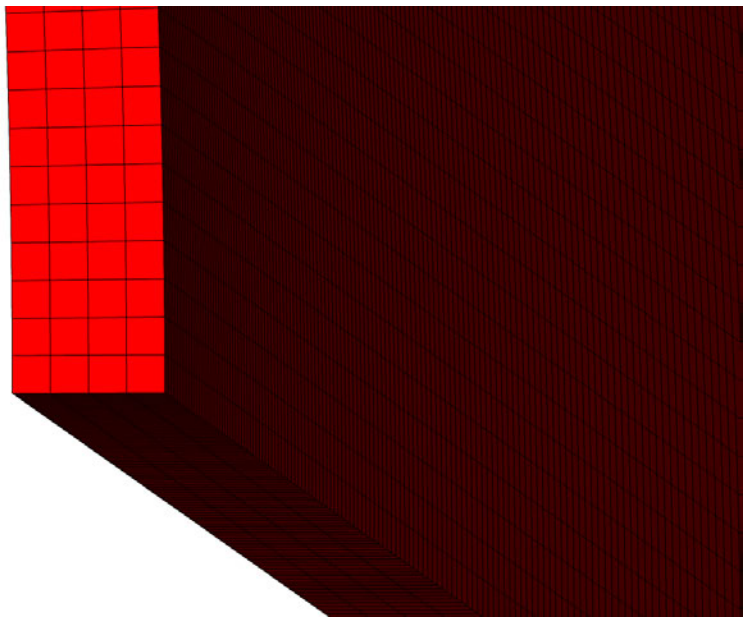


Figure 16. Mesh with four elements through the thickness of the pipe wall. The starting aspect ratios of the elements at pipe mid-length are 4:4:1 in the radial, axial, and circumferential directions, respectively.

3.2.2. Solver, Control Settings, and Error Tolerance Refinement Study

For the PB simulations, we used the conjugate gradient (CG) solver in Adagio with the finite-element tearing and interconnecting (FETI) pre-conditioner. The solver settings in Table 4 were used to produce the uniform-temperature PB results in Table 2.

The settings in Table 4 were obtained by performing the solution cost vs. accuracy study summarized in Table 5. Selected cases from Table 2 are listed in Table 5. The cases span a large temperature range, from 20°C to 700°C, and the s-s curves at each temperature yield closely coinciding failure pressures. These selected cases provide a tough test for the solver variants in Table 5: can the strength orderings of the selected s-s curves be accurately determined despite the small separations in the failure pressures, and is this successfully done at the low end 20°C and high end 700°C temperatures?

The error tolerances in column two of Table 5 are customary for Adagio analyses, according to analyst experiences in balancing calculation cost and accuracy in many projects prior to the PB project. Relatively small average differences (of about 2.8 psi or 0.24%) exist between results in column 2 and column 3 (the latter has significantly stricter error tolerances). But a difference in curve-strength orderings occurs for try3-20C and try6-20C cases. In further tightening the error tolerances from column 3 to column 4, the curve-strength orderings do not change and the calculated failure pressures are virtually unchanged—only the results for try26-700C change, and only by 0.1 psi or 0.01%. Therefore, the error tolerances in column 2 are not sufficient for curve strength ranking purposes, and the tolerances in columns 3 and 4 both appear adequate; results are stable when error tolerances are tightened to these levels (for this particular type of PB analysis).

Table 4. Solver error tolerances and algorithm controls in uniform-temperature PB calculations.

Solver Error Tolerances	CG
Target relative residual	1.00E-06
Acceptable relative residual	2.00E-04
Max iterations	10,000
Min iterations	3
Max cutback	10
Cutback factor	0.5
Growth factor	1.1
Iteration window	100
Target iterations	400
Max multiplier	100
Min multiplier	1.00E-12
Max timestep	1.0
Hourglass effective moduli	Elastic
	FETI
Residual norm tolerance	1.00E-03
Iteration updates	125

We note that the CPU costs of the simulations in Table 5 do not necessarily increase with tightened error tolerances. Complex numerical interactions in adaptive solvers sometimes result in *slower* progress overall when looser iteration-convergence tolerances are used. This can allow larger timesteps and disproportionately increased iterations to converge the nonlinear solution over the larger timesteps. Furthermore, sometimes the timestep has to be retried with a smaller step size if convergence does not occur within the specified iteration limits for the original timestep. So, larger steps can result in non-convergence followed by much numerical rework as the size of the steps are cut down until convergence can be achieved. Because column 3 only saves about 3.4% on average vs. the stricter settings in column 4, we used the more strict tolerances (see Table 4) in the simulations for Table 2.

To get a sense of the effectiveness of the FETI pre-conditioner, we tried the CG solver without pre-conditioning. The CG solver settings from Table 4 were used, except for the following increased iteration targets and limits because of no preconditioning: *iteration window* = 10,000, *target iterations* = 35,000, and *max iterations* = 50,000. For the five cases in Table 5, the non-pre-conditioned simulations took two to five times longer than with FETI preconditioning, and results were insignificantly different, within 0.1%. Therefore, on the PB problem, the FETI pre-conditioner yielded large computational savings with no appreciable drawbacks, so we used it in all that follows.

Table 5. Comparison of predicted failure pressures with uniform-temperature pipe model and FETI-CG solver using the listed error tolerances. Other settings not specified are the same as in Table 4. Cases are listed in order of increasing failure pressure per the last two columns, which use the tightest error tolerances. (8/8 geometry model used, see Section 3.3)

Test and Temperature Cases	**Rel.Resid. = 10^{-4} Accep.Resid. = $2x10^{-4}$** Failure psi (CPU hrs.*)	**Rel.Resid. = 10^{-5} Accep.Resid. = $2x10^{-4}$** Failure psi (CPU hrs.*)	**Rel.Resid. = 10^{-6} Accep.Resid. = $2x10^{-4}$** Failure psi (CPU hrs. *)
try26-700C	702.0 (20.3)	703.8 (5.87)	703.7 (5.24)
try27-700C	704.1 (19.1)	704.2 (5.28)	704.2 (6.21)
try3-20C	1490.70 (12.1)	1484.5 (7.8)	1484.5 (9.78)
try6-20C	1487.20 (4.6)	1485.0 (2.9)	1485.0 (4.39)
try5-20C	1492.60 (41.3)	1485.2 (20.7)	1485.2 (8.26)
* CPU times reported in Adagio output file via global output variable *cpu_time*. Simulations were run on 192 processors of Sandia's Red Sky computer.			

3.3. Check of 8/8 Truncated Geometry Model vs. 1/8 Truncated Geometry Model

As a spot check, the 1/8 truncated geometry model depicted in Figure 15 was run at 700°C with the try26 and try27 s-s curves. (The experiments we will later validate the model against have a failure temperature of approximately 700°C so the s-s curve rankings at 700°C are of high relevance.) From Table 5 a very small difference of 0.5 psi separates the failure pressures for try26-700C and try27-700C. We assessed how well this small separation was preserved when running the 1/8 geometry model. The calculations were run with the FETI solver and control settings listed in Table 4. The 8/8 model has 1.6 million elements and runs more than an order of magnitude slower than the 1/8 geometry model with 200,322 elements. Table 6 reveals that, for the try26 curve, a shift of -3.1 psi in failure pressure occurred with the 1/8 model. For the try27 curve, a shift of -3.7 psi occurred with the 1/8 model. These ~0.5% shifts are very small, supporting a conclusion that the mechanical loading symmetry boundary conditions (BCs) were applied correctly in the 1/8 model.

Furthermore, because the 1/8 model is only used for relative curve-strength ranking purposes, and not where prediction accuracy on an absolute basis is sought, the shift between the 8/8 and 1/8 models is not a problem if the shifting is relatively uniform for the various s-s curves. Reasonably similar shifts of 3.1 psi and 3.7 psi occur for the try26 and try27 curves. The curve-strength rankings are the same with either model. From this spot check we assume that the 1/8 model also gives accurate curve-strength rankings in Table 2 for the other curves and characterization temperatures.

Table 6. Comparison of predicted pipe failures with the 8/8 and 1/8 geometry models (all calculations with FETI/CG solver settings in Table 4).

Test cases	8/8 Model Failure psi	1/8 Model Failure psi	Difference (8/8 – 1/8) Failure psi	% Diff. Failure psi
try26-700C	703.7	706.8	–3.1	–0.44%
try27-700C	704.2	707.9	–3.7	–0.52%

3.4. Model Variants for Pipe Hotspot Heating and Pressurization to Failure

Four different categories of PB simulations with hotspot heating and pressurization to failure are summarized here.

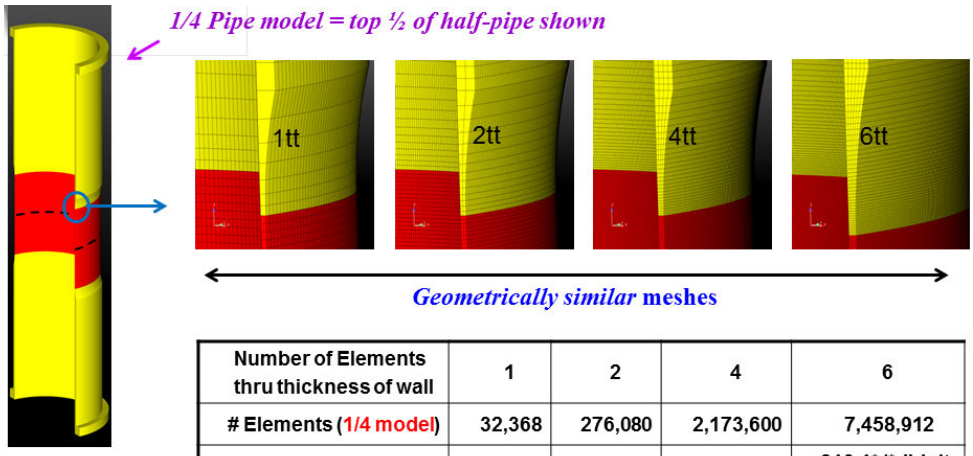
- **PB calculation verification study**: ¼ symmetry Adagio model with mapped/interpolated temperature field BCs from test PB1 thermocouples (see next section).
- **Coupled thermo-mechanical simulations**: ¼ symmetry Adagio-Aria (coupled solid-thermal mechanics) model with heating shroud and thermal radiation and conduction included in model—used to model the validation experiments for experiment design and improvement to minimize validation uncertainty (see sections 4.1 and 4.3).
- **"Self check" simulations to characterize temperature mapping/interpolation error**: ¼ symmetry model with mapped temperature field BCs from *virtual* thermocouples in coupled thermo-mechanical simulations—used to characterize *experimental* BC temperature field reconstruction errors from thermocouple interpolation procedure (see Section 4.4).
- **Full-model validation/UQ simulations**: Full 8/8 pipe model Adagio simulations with temperature field BCs from thermocouple data in the PB tests (see Section 5).

3.5. Calculation Verification of Simulations of Pipe Hotspot Heating and Pressurization to Failure

This section concerns a calculation verification study using a ¼ symmetry model of the pipe with temperature field BCs that closely resemble those in the pipe validation tests but slightly modified to be ¼ symmetric. This allows a more affordable, in-depth calculation verification study to high mesh densities with the less expensive ¼ symmetry model of the pipe, but gives results approximately applicable to the much larger full-geometry pipe model used for the actual, slightly non-symmetric temperature field BCs mapped/interpolated from thermocouples in the validations tests.

3.5.1. Mesh Refinements and Solver Settings in Calculation Verification Study

Figure 17 shows the ¼ pipe geometry and meshes used in the mesh refinement study. The temperature field BC on the pipe comes from the test PB1 thermocouple temperature mapping and interpolation, described in Section 4.2. This BC is used to maximize the applicability of the mesh study and discretization error/uncertainty estimates in the current section to the validation simulations in Section 5. Section 4.4 shows that the temperature field in the experiments varies markedly from front to back and axially. This variation is captured in the ¼ symmetry model, except for minor circumferential and axial asymmetries characterized in Section 4.4. A ¼ pipe model was required for affordability of the mesh refinement study described next.



Number of Elements thru thickness of wall	1	2	4	6
# Elements (1/4 model)	32,368	276,080	2,173,600	7,458,912
Pressure at Fail (psi)	1069	955	850	819.1* (*didn't finish, 36 days on 400cpu's)

Figure 17. The above shows ¼ truncated-length pipe geometry, meshes, linear HEX element counts, and failure pressure results in mesh refinement study.

The 1tt (1 finite element through the thickness of the wall) mesh in Figure 17 was created with approximately equal cell lengths in the axial, circumferential, and through-wall dimensions in the 0.020-in. thin-wall extent of the pipe where the hotspot exists and hence large deformation and ultimately failure occur. Use of 1:1:1 aspect ratio elements is probably not optimal because pressurization-induced deformations cause the walls to thin in the radial direction because of wall stretching predominantly in the circumferential dimension, but in the axial direction as well. Therefore, finite elements near the hotspot that start with equal cell lengths have highly non-uniform aspect ratios by the end of the simulation. We did not study the effects of trying to compensate for element thinning and stretching by using non-1:1:1 aspect elements in the starting meshes.

In the pipe wall-thickness transition region zoom-shots in Figure 17, the 1tt mesh has axial and through-wall cell dimensions that increase simultaneously according to wall thickness as it transitions from 0.02 in. to 0.05 in. The cell axial dimensions are coordinated to approximately match cell widths (through-wall) so that a nearly 1:1 aspect ratio is maintained in these element dimensions in the wall transition region, and therefore throughout the model. However, to maintain an i-j-k structured hex mesh throughout the model, cell size in the circumferential dimension cannot likewise scale with wall thickness. Cell size in the circumferential direction is constrained to be constant along the entire length of the pipe at its inside-diameter (ID) surface. This (constant) arc-length cell dimension of approximately 0.02 in. at the pipe ID is set by the imposed target of the 1:1:1 starting aspect ratios where the hotspot, large deformation, and failure occur.

Our mesh refinement study followed rules for constructing *geometrically similar* meshes (explained next) that are necessary [19] for legitimate Richardson Extrapolation (RE) estimates of the order or rate of convergence of the solution as the mesh is refined. Based on the rate of convergence, the asymptotic solution in the limit of infinitesimal mesh size is estimated and

assigned an uncertainty band. Then, the solution error for any mesh in the sequence of refined meshes can also be estimated with assigned uncertainty.

The 2tt mesh in our refinement sequence doubles the number of elements across the wall. The two elements across the wall are equally spaced, having equal width in the through-wall direction. The refinement rules require that a mesh refinement in one dimension be matched by similar refinements in the other dimensions of the mesh. Therefore, nominally, each element of the 1tt mesh is subdivided into eight elements to get the 2tt mesh and each element in the 2tt mesh has essentially the same aspect ratio as the parent element in the 1tt mesh. "Nominally" here means that the mesh refinement objectives were met as well as reasonably possible by manual trial-and-error iteration with the present meshing facilities in CUBIT [20]. The 4tt and 6tt meshes in Figure 17 were constructed with similar considerations in mind**.**

Another refinement rule is to maintain any grading of mesh density when subdividing elements. For example, in the 1tt mesh, the seven elements that vertically span the wall-thickness transition region have graded (non-constant) cell sizes as the region is traversed in the vertical/axial direction. This grading is required to meet the objective of a 1:1 aspect between the axial and through-wall cell dimensions, as the wall thickness increases in the axial direction. To meet the mesh refinement rule that the vertical mesh-density grading function in this region be preserved under mesh refinements, the seven elements that vertically span the transition region are not halved when subdivided in the axial direction. Rather, they are vertically subdivided such that the mesh density grading in the vertical direction is nominally preserved.

The calculations were run using the solver settings in Table 7. The FETI *residual norm tolerance* was tightened to 10^{-4} vs. Table 4, and experimentation for improving solver efficiency (reducing CPU times) led us to lower some of the iteration limits so that timestep cutbacks and retries would occur sooner instead of allowing high numbers of iterations that ultimately were often unsuccessful.

3.5.2. Mesh Refinement Results, Convergence Assessment, and Estimation of Solution Error and Uncertainty

The failure pressures calculated with each mesh are plotted and listed in Figure 18. As noted in Figure 17, the calculation with the 6tt mesh did not finish within an allowed 36 days on 400 processors. If the calculation was allowed to run until it reached instability failure, the associated failure pressure would be greater than the 837 psi level at the end of the 36-day calculation. Therefore, 837 psi is the lowest potential failure pressure plotted in Figure 18 for the 6tt mesh. The calculated material damage quantities of equivalent plastic strain and tearing parameter [3] at the point when the 6tt simulation timed out were very low relative to values from 1tt, 2tt, and 4tt results (see Table 8). So the 6tt simulation was probably far from reaching failure instability. Therefore, 837 psi is certainly a lower bound, but not a plausible value of failure pressure that would be yielded by a finished 6tt mesh simulation.

For an upper bound, the plotted trend of decreasing computed failure pressure with mesh refinement shown in Figure 18 indicates that failure pressure for the 6tt mesh can reasonably be assumed to be less than the 874 psi result for the 4tt mesh. So, representative potential failure pressures within the lower (837 psi) and upper (874 psi) assumed limits are chosen as 845, 855, and 865 psi as plotted in Figure 18.

Table 7. Solver error tolerances and algorithm controls in mesh study simulations and all subsequent PB simulations.

Solver Error Tolerances	CG
Target relative residual	1.00E-06
Acceptable relative residual	2.00E-04
Max iterations	20
Min iterations	3
Max cutback	10
Cutback factor	0.5
Growth factor	1.25
Iteration window	100
Target iterations	20
Max multiplier	500
Min multiplier	1.00E-12
Max timestep	10
Hourglass Effective Moduli	Elastic
	FETI
Residual norm tolerance	1.00E-04
Iteration updates	10

Failure pressure is not a nodal quantity or a quantity computed "on the mesh." It is determined by the point of non-convergence in the calculation. We are not aware of any theory that would establish a theoretical rate of convergence for computed failure pressure. However, both empirically convergent results in Table 5 and empirically well-behaved and consistent results in figures 18 and 36 provide a basis for treating failure pressure as a convergent quantity yielded by the computations. Therefore, we apply RE to determine the empirical rate of convergence of computed failure pressure as the mesh is refined [21, 22]. Although this use of RE is on somewhat thin ground from a theoretical standpoint, we link computed failure pressure to computed stress state in the pipe because the stress state (and the solver algorithm and error tolerances) govern when the point of non-achievable equilibrium occurs in the calculation as an indication of material failure. We venture to say that failure pressure might therefore be expected to converge at the same rate as computed stresses, i.e., at an ideal rate of 1.0. This is the theoretical rate at which calculated stresses converge with mesh cell size for the second-order spatial discretization of displacement fields in Adagio.

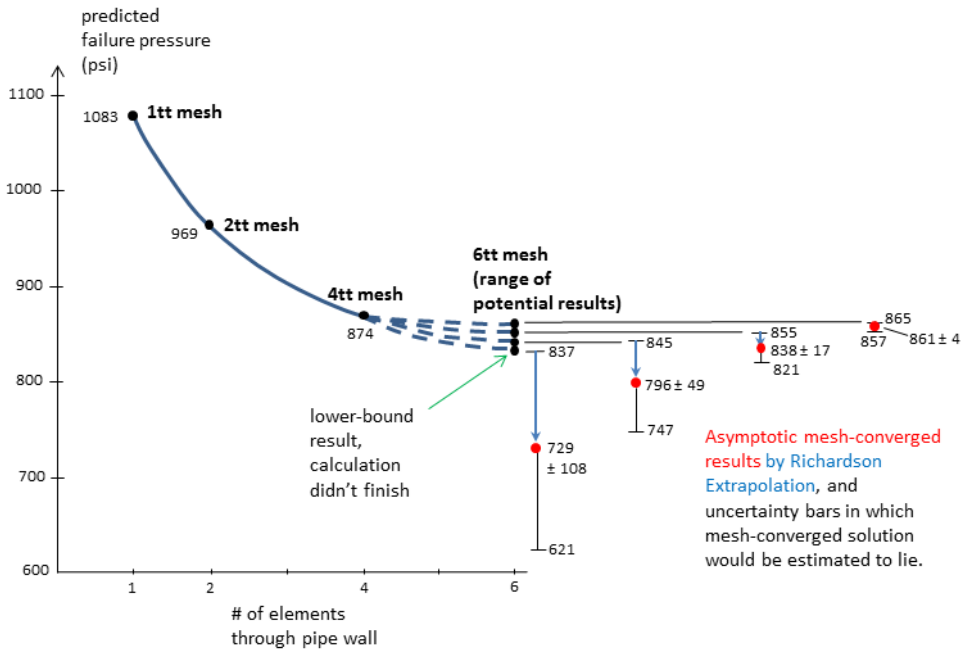


Figure 18. Calculated pipe failure pressures in the mesh refinement study. The lowest result for the 6tt mesh is from a calculation that did not finish within an allowed 36 days on 400 processors. Various plausible finishing results are plotted for the 6tt mesh, along with Richardson Extrapolation estimates of asymptotic mesh-converged values (red dots) and associated solution uncertainty bars.

Table 8. Calculated material damage quantities Equivalent Plastic Strain and Tearing Parameter for simulations in mesh study (spatial maximums over pipe, see, for example, Figure 27).

Mesh	Maximum Equivalent Plastic Strain in Simulation	Maximum Tearing Parameter Value in Simulation
1tt	138%	5.5
2tt	208%	8.89
4tt	116%	3.53
6tt *sim. timed out	27%	0.71

Results from the three coarsest meshes, 1tt, 2tt, and 4tt, yield an empirical rate of convergence of 0.26. (See Reference 22 for a straightforward recipe for calculation of empirical rate of convergence.) This is far from the surmised ideal rate. The consensus of the solid mechanics analysts in the project was that the 1tt-mesh results do not lie in the asymptotic regime of convergence for this problem—a requirement for RE to perform well. This is because structures modeled by single-element-thick linear hex elements have no resistance to bending moments. Therefore we pursued the more refined triplet of 2tt, 4tt, and 6tt meshes/results to apply RE. As Table 9 indicates, uncertainty in the finished 6tt mesh result yields uncertainty in the empirical rates of convergence and, therefore, uncertainty in the asymptotic failure pressure estimated by RE (see Ref. 22) in the limit as mesh size decreases to zero. The estimated potential asymptotic failure pressures are plotted in Figure 18.

Table 9. Potential empirical orders of convergence based on 2tt, 4tt, and 6tt meshes, and Richardson Extrapolation estimates of asymptotic mesh-converged failure pressures.

6tt Mesh Potential Finished Result (failure pressure, psi)	Potential Empirical Order of Convergence	RE Estimates of Potential Grid-Converged Failure pressure results (psi)
837	0.73	729
845	1.15	796
855	1.87	838
865	3.1	861

In Figure 18 an uncertainty bar is shown about each estimated potential asymptotic value. Each uncertainty bar has a half-length equal to the distance between the asymptotic estimate and the corresponding 6tt mesh value. The half-length of each estimated asymptotic solution's uncertainty bar is about 80% of that given by Roach's Grid Convergence Index [21, 22] when the recommended factor of safety 1.25 is used. This safety factor accompanies: 1) empirical rates of convergence used for RE; and 2) structured grids refined into geometrically similar meshes. The present study meets these criteria. Nonetheless, we used the simpler uncertainty bar sizing approach described at the beginning of this paragraph.

Recalling our prior arguments that the 6tt mesh potential results are considered plausible, we further consider only the results in the bottom three rows of Table 9. The bottom row shows a potential order of convergence that seems implausibly high, whereas the two middle rows have much more reasonable values. Therefore, we discount the last row's potential value of failure pressure (865 psi). We are left with the two middle rows in Table 9. We make no further judgment that one of these rows has more or less plausibility or probability than the other. Therefore, we obtain a combined uncertainty range that which spans the two uncertainty bars in Figure 18 that correspond to the two middle rows in Table 9. Thus, the asymptotic grid-converged failure pressure for the modeled conditions in this mesh study is estimated to lie within an uncertainty range between 747 psi and 855 psi. This uncertainty estimate will be used in the model validation uncertainty analysis discussed in Section 5.2. To prepare for this, we note that this uncertainty range (747, 855) psi corresponds to a range of potential corrections (-228, -336) psi added to the 1tt-mesh result.

4. Modeling and Design of the Pipe Bomb Validation Experiments to Minimize Error and Uncertainty in the Experiments and Validation Simulations

In accordance with model validation best practices (see e.g., Refs. 23 and 24), a model of the envisioned validation experiments was constructed and used to help refine the experiment conditions and the locations of measurement sensors to best support the model validation objectives. One of the test objectives was to test the temperature-dependent constitutive model over large ranges of pressures, temperatures, ramp rates, and large temperature gradients on the pipes. Coupled thermal-structural modeling (Section 4.1) was used to help define the pipe length and the experimental heating configuration to achieve the spatial temperature variations sought. The model was also used to devise a thermocouple (TC) placement scheme suitable to capture the

anticipated temperature field variations and support an interpolation procedure to reconstruct the temperature field boundary condition on the pipe surface (Section 4.2). Results from the temperature field reconstruction prompted an experimental adjustment to compensate for convection effects (Section 4.3). Representative simulations were then performed to test the temperature mapping and interpolation capability and characterize associated error and uncertainty (Section 4.4).

The design of the pipe wall thickness and the size of the internal slug for internal volume reduction (explosive energy reduction) were based on simplified handbook models and formulas. These design aspects were driven by experimental safety restrictions regarding explosive rating limits of the test facility.

4.1. Coupled Thermal-Mechanical Modeling to Help Design the Validation Experiments

Coupled thermal-structural modeling was used to help define the pipe length, and the size and temperature of the heating shroud and its location relative to the pipe, in order to achieve the spatial temperature variations sought. The model included thermal radiation exchange between the heating shroud, the dynamically bulging pipe, and the surroundings. Radiation exchange between the pipe and the internal slug (see Figure 1) was also modeled, along with heat conduction along and through the pipe walls. Convection was ignored because of the difficulties mentioned in Section 4.2. Nonetheless, the simplified thermal model was deemed sufficient to aid in designing and planning the thermal aspects of the validation experiments.

During pressurization, the hotspot bulges outward because of temperature-related weakening. The bulging brings the hotspot closer to the heating plate. During this process, the view factors for radiative heat exchange on the inside and outside of the pipe must be updated after each timestep or after a prescribed sufficiently small number of timesteps. This brings a challenging and expensive computational aspect to the coupled simulations. Fortunately, the bi-symmetric temperature field shown in Figure 19 allows a reduced 1/4 geometry to be used for all the calculations in Section 4. However, the 1/4 model used here has full shoulder extensions like the version at left in Figure 14.

The simulation provided a rough indication of the temperature pattern on the pipe in the planned experiments. This was useful in devising locations of thermocouples on the pipe surface for approximate reconstruction of the experimental temperature field. The thermocouple location and interpolation procedures are described next.

4.2. Thermocouple Placement and Interpolation Scheme for Temperature Field Boundary Condition Reconstruction on Pipe

Early in the design and planning of the validation activity we judged that modeling the thermal aspects of the experiments would contribute prohibitively large uncertainty to the validation study and unacceptably degrade the precision with which we could resolve the accuracy of the constitutive model. In particular, for thermal modeling purposes, the emissivities of the radiating surfaces of the shroud, pipe, and internal slug were not known to within ±10%. We also estimated that modeling convective heat losses from the interior and exterior surfaces of the pipe could be in error by ±25% or more. Thermal modeling difficulty is compounded by the fact that pressurization of the pipe entails continually introducing new gas (mass) and its associated enthalpy into the pipe. These factors not only affect the gas temperature inside the pipe in a complex way, but also foreseeably affect convection inside the pipe. It was not even known if

convection correlations exist that account for continually introduced gas and/or for the very high pressures in the experiment (several hundred atmospheres).

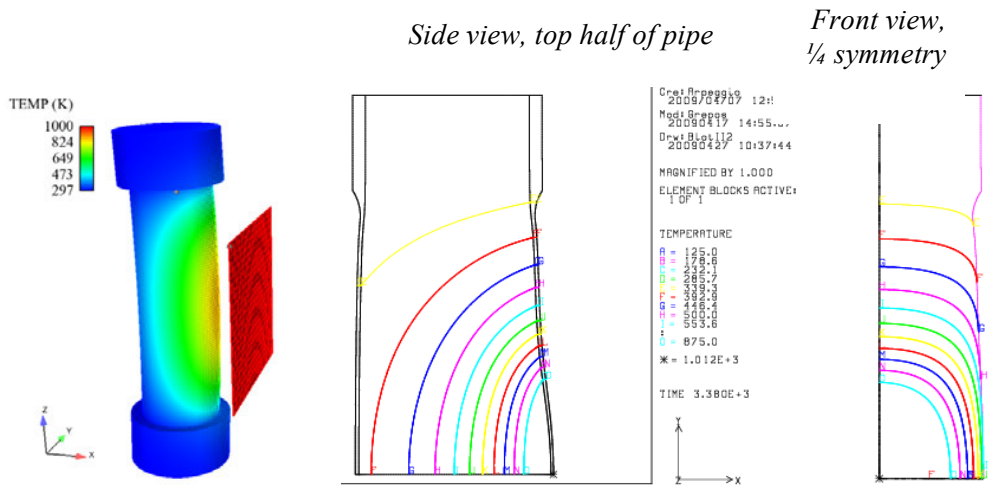


Figure 19. Coupled Thermal-Solid Mechanics simulation results for sizing and positioning heating plate in the tests.

Fortunately, error and uncertainty associated with thermal aspects in the validation problem were largely circumvented by measuring pipe surface temperatures at specifically designed thermocouple locations based on model simulations. Then, spatial interpolation was used (with interpolation error/uncertainty factored in) to provide pipe wall temperature BCs for the pipe structural response simulations. The procedures for thermocouple placement, temperature interpolation, and associated UQ are explained as follows.

An initial constraint was that 18 or fewer thermocouples (TCs) were allowed per experiment because of limited data channel availability in the data acquisition system. Freedoms in the sensor placement and interpolation design problem were: (1) the 18 TCs could be placed anywhere on the pipe surface; and (2) any method of spatial interpolation could be used that was compatible with proposed sensor patterns, was expected to yield reasonably good interpolation accuracy, and was achievable within project time and resource constraints.

Another consideration for sensor placement and interpolation was that the quantities we wanted to predict (such as failure time, pressure, and temperature) were anticipated to be most affected by the hotspot region on the pipe, where the steel would experience the greatest temperature-associated material strength loss, bulging, and tendency for failure. It was recognized, however, that the response prediction problem is a global one. Stress, strain, and failure at or near the hotspot are affected by material "give" everywhere else in the pipe. Therefore, temperature-dependent material relaxation had to be modeled everywhere on the pipe. Even so, it is reasonable to assume that it is most crucial to model the effects most accurately in the hotter regions of the pipe, where most of the stress/strain/failure action occurs. Accordingly, the temperature contour information in Figure 19 was used as an indicator for relative concentration of TC sensor coverage over the pipe—greater concentration of TCs in the hotter regions, less in the cooler regions. Within this generalized objective, the locations of the individual TCs were determined as explained next.

A quasi-Hermite 2D bi-cubic polynomial interpolation scheme was identified as a method that could be quickly developed to meet our needs. The bi-cubic interpolating flexibility of 2D

Hermite polynomials can conform reasonably well to the temperature field over the surface of the pipe if the field is subdivided into a suitable set of interpolation "patches" as shown in Figure 20. The assemblage of all interpolation patches yields a C^0 interpolated temperature field that is continuous (but not necessarily smooth) across patch boundaries. So, temperature *slope* across patch boundaries is not necessarily continuous.

Generation of a Hermite bi-cubic interpolation function over a quadrilateral interpolation domain (patch) requires input of temperature and three slope-affiliated terms at each corner of the patch: T, $\partial T/\partial z$, $\partial T/\partial \theta$, and $\partial^2 T/\partial z \partial \theta$. If the derivative information at the patch corners is from numerical approximation, as employed here, then the overall interpolation scheme is referred to as quasi-Hermitian. The objective then becomes one of laying out a TC pattern that supports the most accurate determination of temperature and the three derivatives at the corners of the patches in Figure 20—given the importance of obtaining the greatest accuracy in the vicinity of the hotspot and accuracy importance drops as temperature drops with distance from the hotspot.

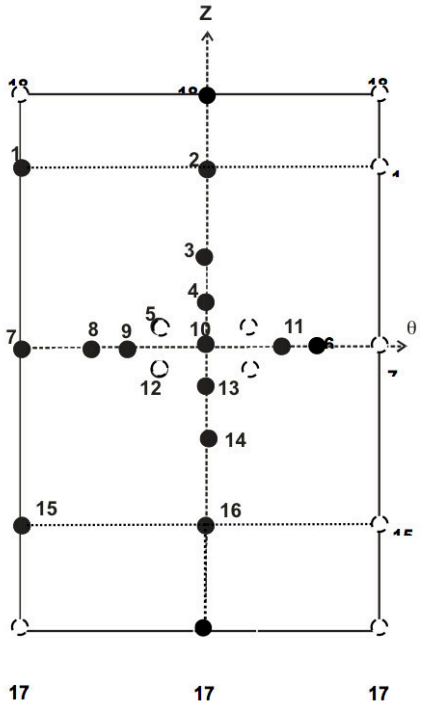


Figure 20. Division of pipe exterior surface into eight quadrilateral "interpolation patches." The surface of the pipe is represented as though sliced at the back, unwrapped, and laid flat in θ-z space, as shown. TC locations on the pipe surface are shown (not drawn to scale, see Table 4.2 in Romero et al. [18] for numerical coordinates). TC 10 marks the origin ($\theta = 0$, $z = 0$) of the θ-z pipe-surface coordinate system, where pipe circumference ranges from $-\pi < \theta \leq \pi$. Note that the TC numbering here is from the convention used in the interpolation subroutine (see Ref. 18) and *not* the ID numbering of TCs in the experiments (see Figure 21).

In Figure 20, the θ-z pipe-surface coordinate system has its origin at the hotspot at pipe mid-height. A preferential TC placement scheme increases the concentration of TCs as the hotspot location ($\theta = 0$, $z = 0$) is approached, as shown in the figure. The location of the origin also coincides with an experimental pipe temperature distribution expected to be nominally symmetric about the vertical z-axis at $\theta = 0$, and to a lesser extent (because of convection) to be approximately symmetric about the horizontal θ-axis at $z = 0$. Nonetheless, the TC placements

were designed to enable temperature non-symmetry in either or both directions to be detected and to be approximately modeled by the interpolation scheme.

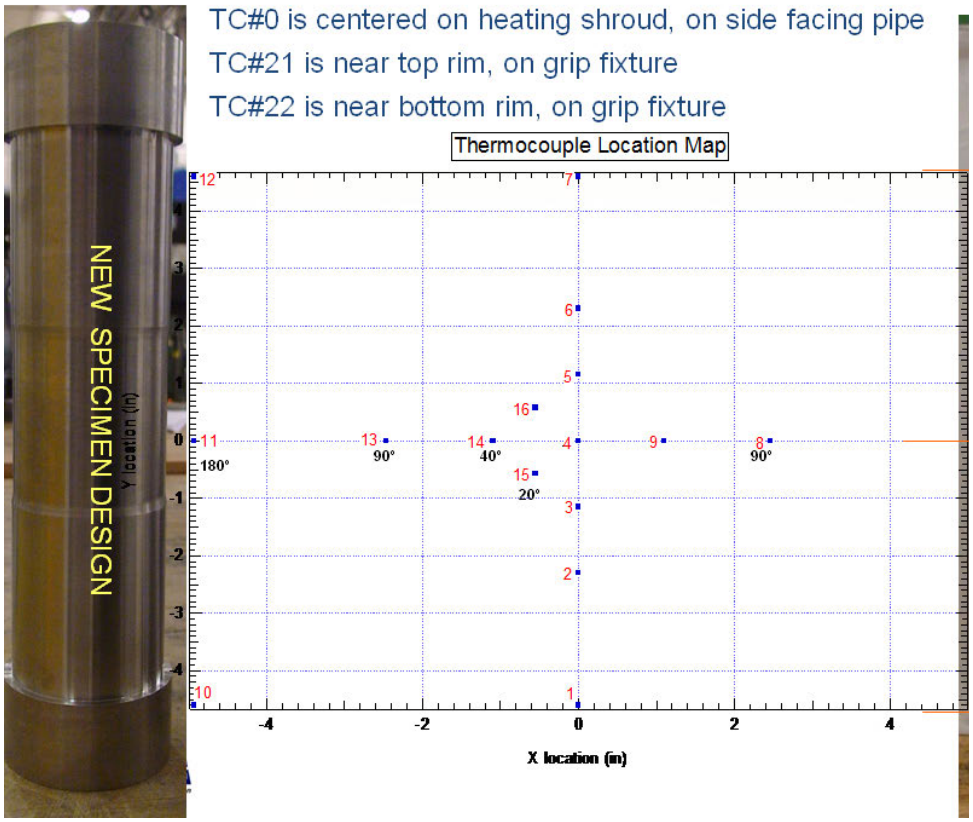


Figure 21. TC numbering used in the experiments, which is different from the interpolation code's numbering scheme shown in Figure 20.

The layout of TCs in Figure 20 allows some of the derivatives to be evaluated non-uniquely; any of several choices could be made. For example, $\partial T/\partial\theta$ at TC10 could be evaluated from a Lagrange cubic polynomial fitted to the temperatures of TCs 7, 8, 9, 10 or TCs 8, 9, 10, 11, for instance; or simply be set to zero according to a physical argument of C^1 smoothness (continuous slope across patch boundaries) + temperature field symmetry about the z axis. Testing each of these alternatives showed that the first option yielded the least interpolation error on the test problems described in Section 4.4. It was similarly found that setting $\partial T/\partial\theta$ at TC7 according to the option described in Appendix C of the work by Romero et al. [18] gave less overall interpolation error than using a C^1 smoothness + symmetry condition $\partial T/\partial\theta = 0$ at TC7.

Symmetry + C^1 smoothness conditions $\partial T/\partial\theta = 0$ at the front and back of the pipe ($\theta = \pm\pi$) were also not used for patch corners at TCs 1, 2, 15, 16, although this would have been simpler for the interpolation code. Instead, the upper TCs 2, 1 and lower TCs 16, 15 were used to prescribe linear temperature drops with circumferential distance from front to back. We did not assess whether this linear temperature decrease caused smaller overall interpolation error than a nonlinear decrease from front to back would if $\partial T/\partial\theta = 0$ were specified at front and back. An assessment was not performed because temperatures in this region varied only slightly around the pipe because its top and bottom rims were held to approximately 293K by active cooling from water jackets.

TCs 5 and 12 in Figure 20 were not planned for use in constructing the interpolated temperature field. Rather, they were used to provide a check on interpolation accuracy. We figured that the accuracy check would be most useful if conducted near where the hotspot was anticipated to be. We also decided that one TC should be above the z=0 plane and one symmetrically below the z=0 plane so that information could be obtained on non-symmetric interpolation errors in the anticipated convection-induced asymmetric temperature fields about the z=0 plane. We envisioned that the information on interpolation error might be useful to inform uncertainty estimates in the model validation phase of the project. Unfortunately, the very limited point information did not prove sufficient to make useful inferences about global interpolation error effects on computed failure pressures—the quantity of greatest validation interest. So we used the strategy described in Section 4.4 instead.

4.3. Use of Temperature Mapping/Interpolation to Adjust Experiments to Compensate for Convection Effects

Figure 22 shows a tested pipe (0.05 in. uniform wall thickness) heated and pressurized to a bulging state and then depressurized before failure. The thermocouple leads can be seen on the tested pipe. The accompanying simulation results show a significantly non-symmetric temperature field with greater temperatures above pipe mid-height than below. The skewed temperature distribution was attributed to external and internal heat convection. The hottest point on the pipe was significantly above the center TC (#4 in Figure 21). The experiment was therefore reconfigured to lower the heating plate by about ½ inch to better center the hotspot about the center TC. Thus, the temperature mapping and interpolation capability was directly useful in fine-tuning the heating configuration in the experiments.

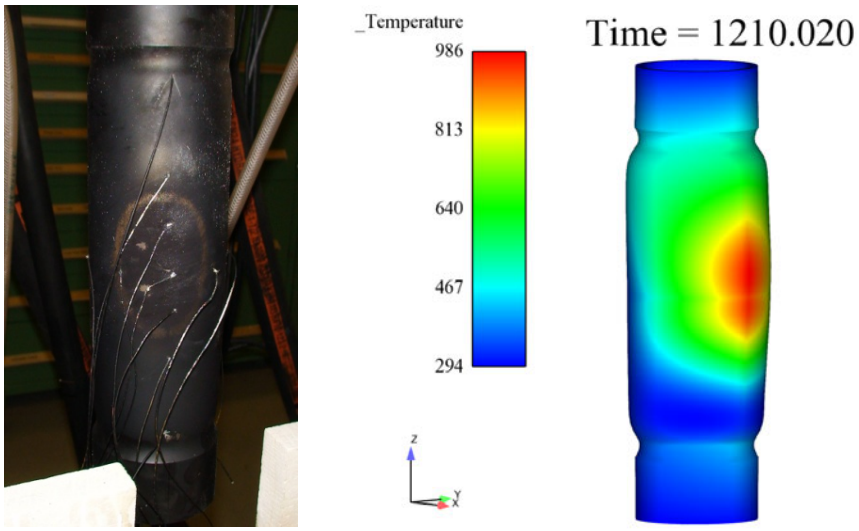


Figure 22. Convection-caused non-symmetric temperature field in early test setup. The heating shroud was subsequently lowered to better center the hotspot about the center thermocouple.

4.4. Simulations and Analysis to Test Temperature Mapping and Interpolation Procedures and Characterize Associated Error and Uncertainty

Here we describe some relevant tests to determine the accuracy of the temperature interpolation and mapping procedure that transforms TC point-temperature data into temperature field BCs for model simulations. Further, we quantity the impact of interpolation error caused by sparseness of

TC spatial coverage. We then describe how the results are leveraged to estimate a correction (with uncertainty) for interpolation-induced error in predicted failure pressures in the validation simulations.

Such quantification of induced error can also be used to actively adjust and/or optimize the number of TCs, their locations on the pipe surface, and the interpolation scheme (quasi-Hermite or others) to minimize the temperature BC reconstruction error and its effects on calculated failure pressure. This was done at an informal level only, using expert judgment because of time and resource limitations.

4.4.1. Synthetic "Nearby Problem" Representative of PB Validation Experiments

A synthetic "nearby problem" was constructed to closely emulate the tests that the model will be validated against in Section 5. In these tests (referred to as "700C-hold" PB1 and PB4), a pipe is heated to produce a hotspot with a steady target temperature of 700°C at the hottest point. Once this temperature is reached, the pipe is pressurized until failure. See Section 5.1 and Reference 2 for a further description of the test conditions.

Convection is ignored in the model, so it predicts artificially high temperatures for a given heater plate temperature. Therefore, the plate temperature from a different experiment (PB2) was used because it had a plate temperature that maintained the pipe hotspot at 672°C in the test. It was anticipated that use of this lower plate temperature would nominally offset the lack of convective cooling in the simulation such that the pipe temperature field in the simulation would be close to the temperature field in the 700C-hold experiments. We later show that this is indeed the case.

The pipe pressurization and heating-plate temperature inputs to the coupled thermal-mechanical simulations are plotted in Figure 23. The 4tt mesh (Figure 17) and solver settings in Table 7 were used in the simulations described next.

Two variants of the stainless-steel constitutive model were used in the simulations: s-s curves of high and low strength, as explained in Section 2.5.1. With these and the radiative emissivity values described next, the coupled simulations yielded a representative set of time-developing spatial temperature fields on the pipe.

The following emissivities were nominal estimates from consulting various references in the literature. The emissivities are not used in the eventual validation simulations, so their accuracy is not critical from this standpoint. The values only need to be representative enough to obtain temperature fields on the pipe that are close to the fields in the PB1 and PB4 experiments.

The outside surface of the solid slug inside the pipe (Figure 1) was assigned an emissivity of 0.5. The painted-black Inconel heating plate was assigned an emissivity of 0.7. Except for the heating plate, the surroundings that exchange radiation with the outside surface of the pipe were modeled with an emissivity of 1.0 and ambient temperature of 296 K. Emissivity of the outside and inside surfaces of the pipe is assigned a value stated in the first column in Table 10. The lower and upper emissivities of 0.7 and 0.84 in the table were deemed a reasonable range to represent the radiative heating uncertainty in the problem.

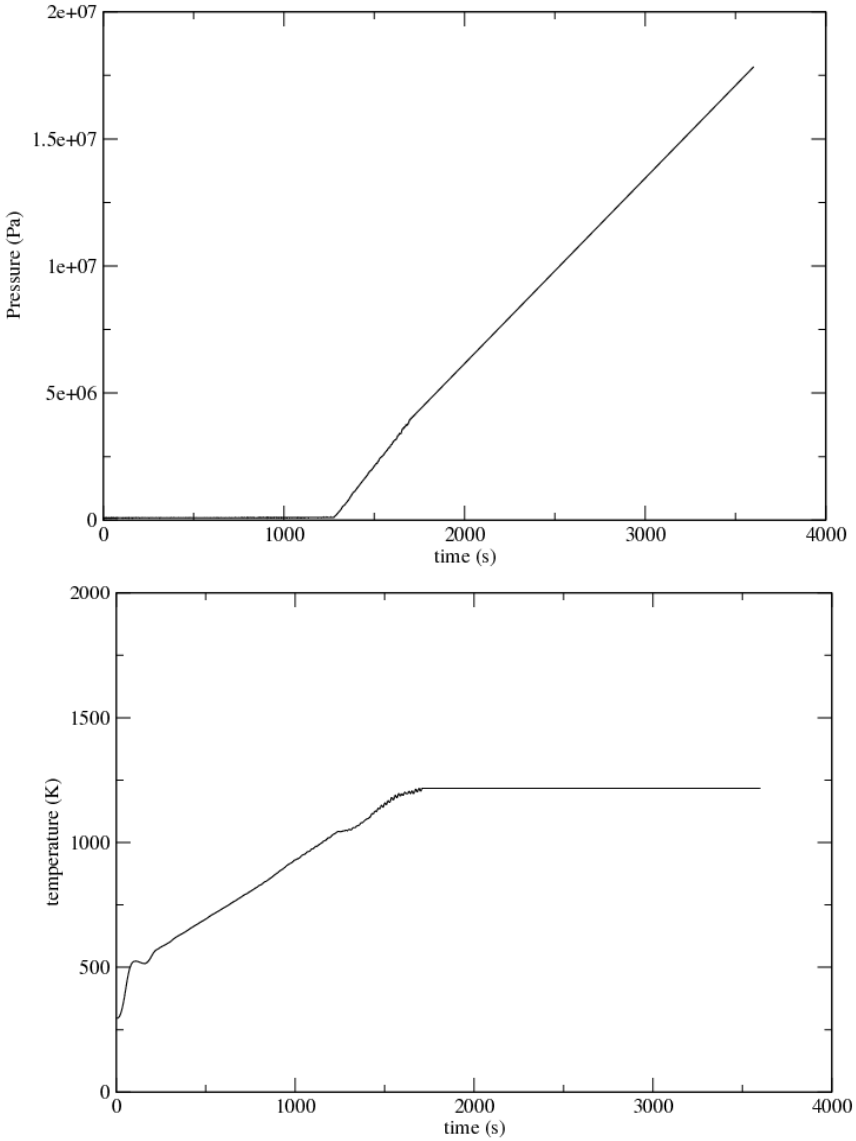


Figure 23. Test PB2 pressure and heating-plate temperature inputs to coupled thermal-mechanical simulations.

Table 10. Failure-related quantities in coupled-simulation test cases.

Simulation case	Time @failure (sec.)	Temperature @failure (C)	Pressure @failure (psi)	Tear Param. @failure
Coupled-high-0.84	2277.1	759.0	809.9	6.23
Coupled-low-0.84	2255.9	759.0	791.6	5.40
Coupled-high-0.7	2386.0	706.9	903.7	3.9
Coupled-low-0.7	2350.4	706.8	873.0	3.50

The combinations of high and low material strengths and high and low radiative heating in Table 10 give a representative range of temperature fields in the simulations to assess temperature mapping and interpolation error effects on calculated failure pressures. The hotspot peak temperatures (at TC 10 location in Figure 20) are essentially the same for high and low strength variants when emissivity = 0.84. Peak temperatures are also the same for high and low strength when emissivity = 0.7. Moreover, Figure 24 shows that the computed temperatures at other TC locations essentially plot over each other for emissivity=0.84 regardless of whether high or low strength material curves are used, and likewise for emissivity=0.7. Thus, peak temperatures and the entire temperature fields are essentially indifferent to changes between high and low strength material curves. However, the temperature fields are significantly different when emissivity is changed from 0.7 to 0.84. Hotspot temperatures in Table 10 are 52°C higher when emissivity = 0.84 than when emissivity = 0.7, whether high or low strength material is involved. Figure 24 shows the computed temperatures at other TC locations are also hotter when emissivity = 0.84 than when emissivity = 0.7, as expected.

Although changing from high to low strength curves negligibly impacts the temperature field, Table 10 reveals that failure pressure decreases by an average of about 3% (2.3% when emissivity=0.84 and 3.5% when emissivity=0.7). But the curve-to-curve material strength variations affect failure pressure far less than changing emissivity from 0.7 to 0.84. The latter increase raises the temperature field, thereby reducing the strengths of the applicable high and low strength curves, reducing failure pressure by 11.5% for high material strength and 10.3% for low strength.

4.4.2. Temperature Mapping and Interpolation Errors and Impact on Calculated Failure Pressures

Figure 25 shows, for a representative simulation case, the front and back views of the temperature fields of the coupled and the “Interp” temperature-mapped/interpolated simulations at a common time—the time of failure in the coupled simulation. The Interp results are obtained from a mechanics-only simulation with temperature field BCs obtained from mapping/interpolation of synthetic temperature data (at the 18 TC locations) generated by the coupled simulations.

Noticeable differences exist between some areas of the coupled and Interp temperature fields in Figure 25. Figure 26 better reveals the spatial variation of error in the interpolated temperature field. Because the interpolated field is constructed from data at the TC locations, interpolation error at these locations is zero. Substantial error exists in many regions, especially where the temperature field varies quickly in space and, simultaneously, TC coverage is sparse.

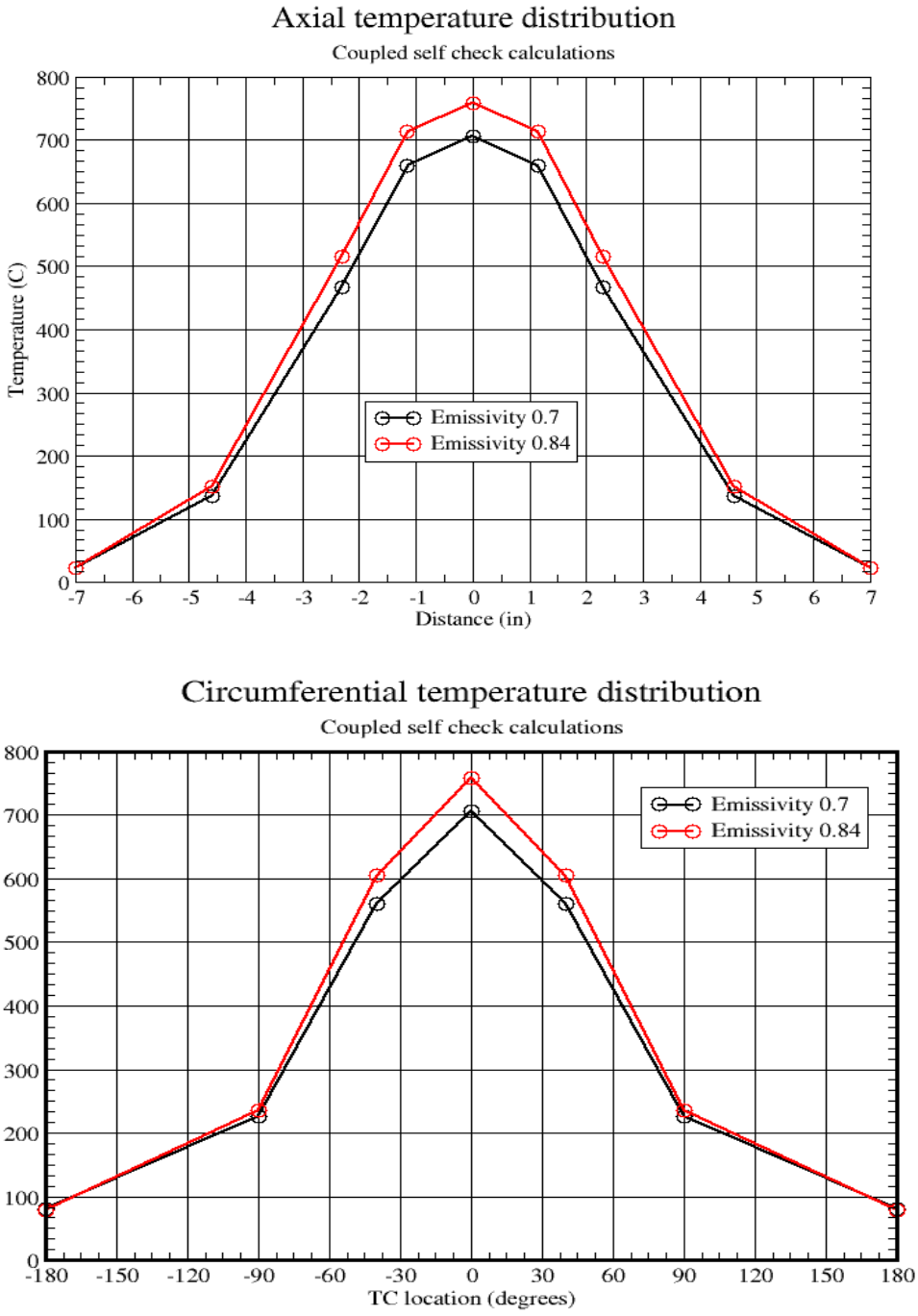


Figure 24. Coupled simulations temperature results (degrees C) plotted at thermocouple locations up and down the front of the pipe (left plot) and around pipe at mid-height (right plot). These locations correspond to TCs on *z* and θ axes in Figure 20 that the temperature interpolation scheme is built on. Temperatures are at time of failure in the computations (see Table 10). Temperatures at TC locations are spanned by linear connecting segments for visualization purposes only and are not actual (see Figure 25).

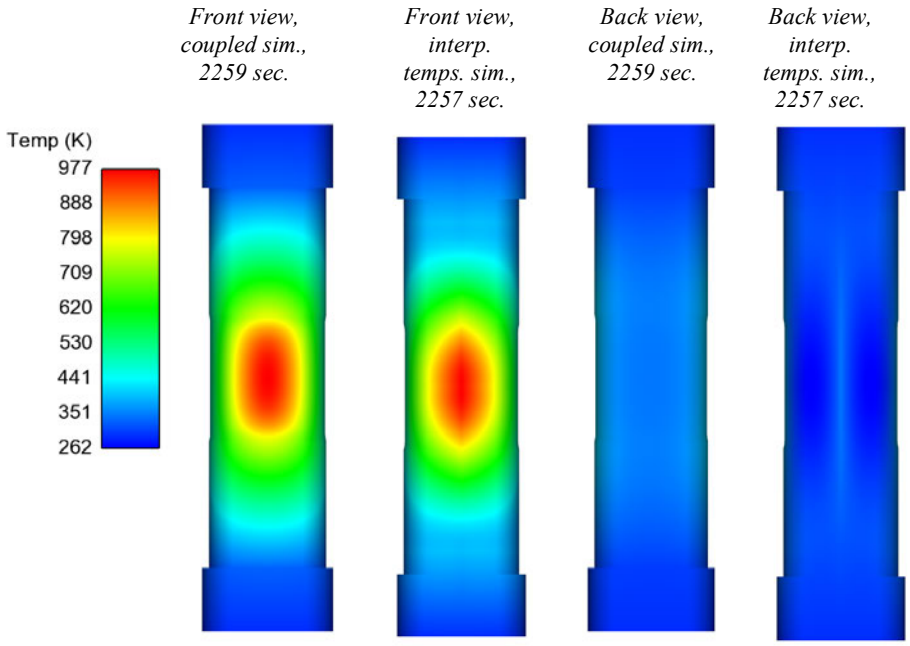


Figure 25. Reference and interpolated temperature fields for simulations with emissivity=0.7 and low-strength material curves.

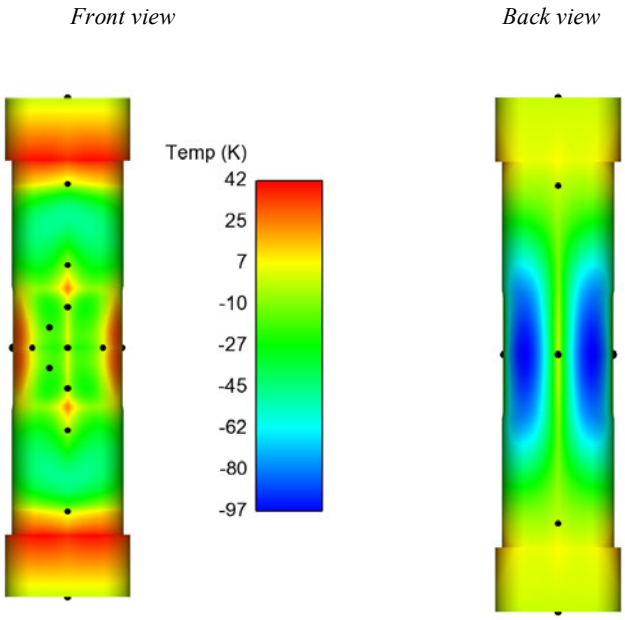


Figure 26. Difference plot of the interpolated minus coupled-simulation temperature fields in Figure 25. Zero interpolation error exists where yellow fades to green, in particular at the TC locations where the temperature data is input to the interpolation scheme (and not interpolated).

The interpolation error also varies over time. The integrated effect of the spatial-temporal interpolation error can be assessed by comparing the coupled simulation results in Table 10 to the Interp simulation results in Table 11. Various comparisons are discussed next.

Table 11. Failure-related quantities in coupled simulations using mapped/interpolated temperature field boundary conditions.

Simulation case	Time @failure (sec.)	Temperature @failure (C)	Pressure @failure (psi)	Tear Param. @failure
Interp-high-0.84	2332.8	758.9	857.9	7.27
Interp-low-0.84	2299.3	759.0	829.0	7.46
Interp-high-0.7	2400.9	706.9	916.2	4.43
Interp-low-0.7	2353.6	706.8	875.8	3.86

The hotspot peak temperatures in Table 11 are effectively the same as those in Table 10; the interpolation scheme ensures that the reference and interpolated temperature fields have the same temperature at all TC locations, including at the TC at the pipe hotspot.

Figure 27 shows that the magnitude of material damage correlates with elevated temperature and that failure occurs at the hotspot where the peak temperature exists. Damage contours and location of failure are shown to be very similar for the coupled and the Interp simulations.

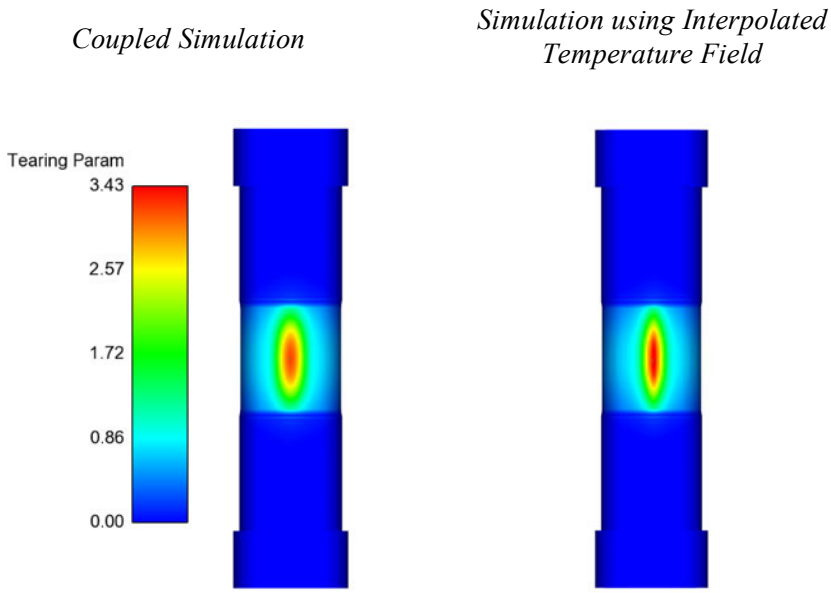


Figure 27. Spatial plots of computed tearing parameter indicating material damage at time of failure in emissivity=0.7 low-strength simulations. Damage is effectively confined to the thin-wall section of pipe where the wall is 0.02 in. thick (wall is 0.05-in. thick elsewhere, except at thicker top and bottom shoulders). White dots locate maximum damage in the coupled simulation. These are negligibly offset from the pipe front-center, where the hottest point on the pipe is. The black dot locates the point of maximum damage in Interp simulations and is essentially the same location as failure in the coupled simulation.

Both the coupled and Interp simulations predict failure at the same location and at the same hotspot temperature. Even though the two models have the same temperatures at the 18 TC locations, spatial interpolation errors in the rest of the temperature field causes the Interp simulation values to differ from those of the coupled simulation. Tables 10 and 11 show that failure occurs earlier and at lower pressures and tearing parameter levels in the coupled simulation than in the Interp simulation.

Nevertheless, the *trends* in output responses vs. changes in emissivity and material strength are in the same *directions* as those for the coupled calculations cited in the paragraph below Figure 24.

- Changing from high to low strength negligibly impacts the interpolated temperature field.
- Changing from high to low strength reduces failure pressures, as shown in Table 11, by an average of ~4% (3.5% when emissivity=0.84 and 4.6% when emissivity=0.7). These impacts are in the same direction as in the coupled simulations, but somewhat larger.
- Increasing emissivity from 0.7 to 0.84 raises the temperature field, which decreases failure pressure by an average of ~6% (6.8% at high strength and 5.6% at low strength). These decreases are significantly less than in the coupled simulations, but in the same direction.
- These emissivity changes affect failure pressure significantly more than the high vs. low curve-strength variations.

Failure pressure is the quantity of interest for the validation comparisons in Section 5. Accordingly, Table 12 lists the failure pressure errors caused by temperature interpolation for the four test cases. The errors are always positive; failure pressures in the Interp simulations are higher than in the coupled simulations by 0.3% to 5.9%. The errors are much larger for the higher-temperature 0.86 emissivity cases than for the lower-temperature 0.7 emissivity cases, regardless of whether high- or low-strength material curves are involved. Errors are slightly greater for high-strength curves than for low-strength curves, regardless of whether emissivity is 0.7 or 0.84.

Table 12. Overestimation of failure pressures caused by interpolation of temperature field for synthetic test cases.

Sim. Case	△ = P_fail_interp - P_fail_coupled	△% = △/ P_fail_coupled
high-0.84	48.0 psi	5.9 %
low-0.84	37.4 psi	4.7 %
high-0.7	12.5 psi	1.4 %
low-0.7	2.8 psi	0.3 %

4.4.3. Application to Model Validation Simulations to Correct Calculated Failure Pressures, with Uncertainty on the Correction

For actual temperature fields that closely resemble the synthetic fields in figures 24 and 25, the characterized pressure prediction errors in Table 12 will approximately apply. Figure 28 compares the synthetic temperature fields against the actual temperature fields at time of failure in the validation tests PB1 and PB4. The temperature fields are similar enough that the results in

Table 12 are used to estimate failure pressure prediction errors and associated corrections (with uncertainty) in Section 5.2, due to reconstruction of the pipe temperature BCs from the tests' TC data.

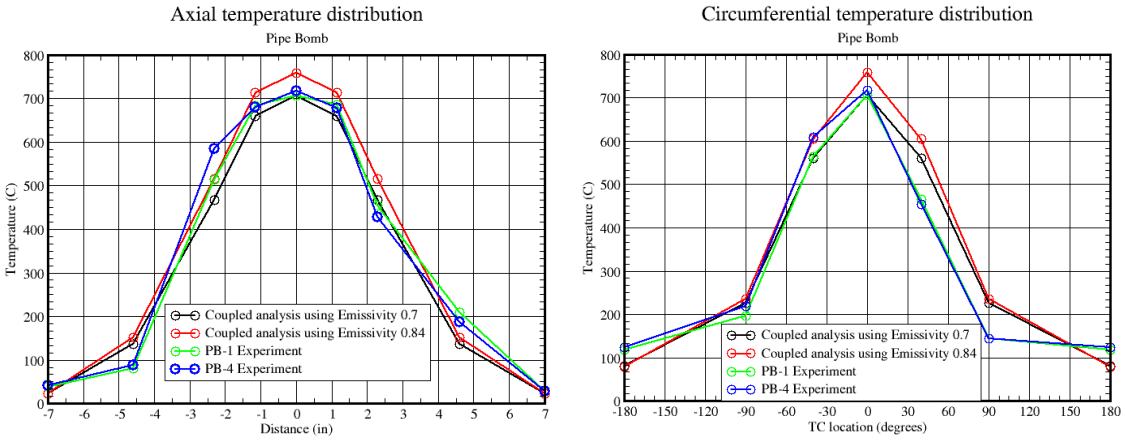


Figure 28. Pipe TC temperatures at time of failure in the 700C-hold experiments PB1 (green curves) and PB4 (red curves) plotted along with synthetic temperature fields (red and black curves) from Figure 24.

5. Validation Experiments, Results, and Processing for Comparison to Model Predictions

In the following, we assess the constitutive model's performance by comparing experimental and predicted failure pressures for the PB tests. We present the tests and simulations; their results and uncertainties; processing of these into a form suitable for the Real Space (RS) model validation method employed; and a description of the comparisons and their interpretation.

As summarized in the Introduction and Appendix D of this chapter, the PB validation problem has a large set of challenging features, but the RS validation and uncertainty quantification (UQ) method appears well equipped to handle these difficulties, compared to other established model validation methodologies. The versatile and practical RS method has also been applied to other complex and challenging calibration and validation problems such as: device thermal response and failure [25, 26]; behavior of irradiation-damaged electronics [27, 28]; combustion in fluids and solids [29, 30]; and foam thermal transport and chemical decomposition causing container pressurization, deformation, and weld failure [43]. Therefore, the methodology demonstrated here can be extended to other difficult model validation applications.

Various fidelity levels of uncertainty treatment can be applied in the RS validation framework. The UQ treatment in this project was driven by severe constraints in the number of simulations that could be completed in the time available. Only five simulations of the model were required in the inexpensive but sophisticated UQ approach applied. Each of the simulations took about a month on 800 processors—the limit of what was feasible under the resource constraints. Nevertheless, we judge that if considerably more simulations and higher-order UQ procedures were conducted, the main conclusions would not be significantly different than those arrived at here.

We emphasize up front that the temperature-dependent constitutive model is the subject of the validation assessment. The pipe-vessel model and the solid mechanics code are the particular vehicles by which the constitutive model's validity is assessed. But it is the constitutive model and not the pipe model that is intended to be used for further predictions subsequent to this model validation activity (and perhaps with other solid mechanics codes). Terminology instantiated [6, 7, 8, 26, 33] to convey this distinction is that the pipe-response model is the "experiment model" in this validation activity and the material constitutive model is the "traveling model" to be validated. Only the constitutive model and uncertainties associated with it "travel" to subsequent uses beyond the validation activity. (See Chapter 10 for further discussion and illustration.) This distinction dictates how the uncertainties associated with the experiments, models, and simulations are categorized and treated in the RS validation approach—which recognizes the different significance and consequences to post-validation predictions that traveling and non-traveling uncertainties present, see the work by Romero [6-8].

The constitutive model being validated carries with it aleatory uncertainty that is an intrinsic element of the constitutive model. This "traveling" aleatory uncertainty reflects material strength variability characterized in the rod tension tests per Section 2 and parameterized into the model per Section 2.5.1, and then propagated to predictions as explained in Section 5.2. In contrast, no traveling *epistemic* uncertainties are parameterized into the constitutive model. In the RS validation approach, when a model being validated *does* have traveling epistemic uncertainties, a certain protocol exists for their propagation to model predictions (see e.g., References 28-30 and 43), and *non*-traveling epistemic uncertainties, such as those associated with experimental boundary conditions, are propagated to experimental results. But the PB problem has no traveling epistemic uncertainties, so the associated UQ protocol can be relaxed for convenience here (as a special case of the general RS method) without loss of validation accuracy or appropriate interpretation. Accordingly, some non-traveling epistemic uncertainties associated with experimental boundary conditions are propagated to the simulation results in Section 5.2, and others are accounted for in Section 5.3, with processing of experimental results from other uncertainty sources, as well.[4]

Experimental and simulation results and their uncertainties are compared and discussed in Section 5.4.

5.1. Experimental Inputs, Outputs, and Uncertainty in Failure Pressure Tests at 700°C

We now present details of hardware geometry and experimental conditions and results that are pertinent to subsequent model validation procedures and comparisons. We consider the 700C-hold set of pipe pressurization experiments PB1 and PB4. The tests were planned replicates of each other. Figure 29 shows the transient internal absolute pressures in the pipes and the transient responses of temperature-control TCs presumed to lie at the peaks of the hotspots in the two tests (TC#10 location in Figure 20). Later, we attempt to account for uncertainties regarding TC accuracy and whether the TCs actually lie at the peak temperatures on the pipes. The peak hotspot

[4] In Section 5.2, one of the non-traveling experimental uncertainties propagated to simulation result uncertainty concerns thermocouple temperature measurement errors. In the RS approach, these boundary condition measurement uncertainties would normally be propagated to uncertainty on experimental results (instead of simulation results). This alternative is illustrated in Appendix A, where temperature measurement uncertainties are propagated to uncertainty of experimental results.

temperatures (as indicated by the control TCs) were ramped up at a rate of approximately 31°C per minute.

When the control TCs reached nominally 700°C, this temperature was maintained while the pipes were pressurized, as shown in Figure 29. Controlled pressurization of the pipes was accomplished via pressure supply tanks, as described by Antoun [2]. PB4 pressurization started about 1.5 minutes after 700°C was reached, while PB1 pressurization started about 4 minutes after 700°C was reached. Minor differences also exist about the nominal pressure ramp rate of 1.3 psi/sec. The differences in pressurization were, from the outset of the project, anticipated to affect failure pressure levels negligibly. However, the experience of project participants concerning stainless steel behaviors at high pressures and especially high temperatures is very limited. The PB1 pipe spent a few minutes longer at 700°C before pressurization than PB4 did, and failed at a lower pressure. The lower failure pressure could conceivably be explained at least partially by the pipe soaking at 700°C longer than the PB4 pipe (the temperature is estimated to vary by <1K through the wall thickness). But the PB1 lower failure pressure appears to be at least partially explained by PB1 vs. PB4 wall-thickness differences and temperature distribution differences over the pipes, as will be established later. Therefore, negligible effects of pressurization timing differences between the PB1 and PB4 tests are assumed in what follows. This assumption may be called into question as further experiences and model development contribute to better understanding of material response/failure dependencies in this regime.

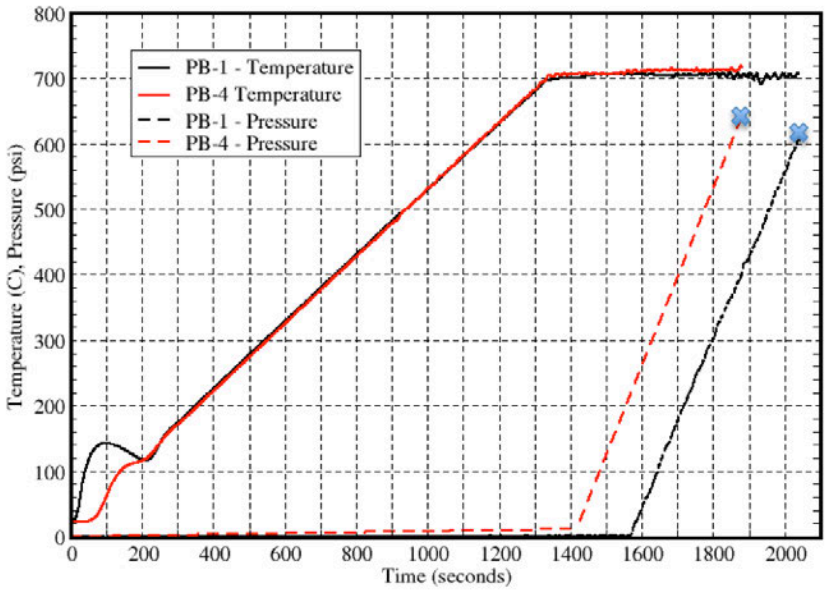


Figure 29. Tests PB1 and PB4 transient internal pressures (absolute) and control thermocouple temperatures at the hotspots. Pipes fail at the pressures indicated and listed in Table 13.

Table 13. Failure pressures measured in 700C-hold experiments. Pressure measurement uncertainties are explained in Section 5.1.1.

Experiment	Measured Pressure at Failure	Uncertainty in Pressure Measurement
PB1	606 psi	± 10 psi
PB4	655 psi	± 10 psi

5.1.1. Experimental Pressure Loading and End Fixturing BCs

Figure 30 replots the pressure loadings and shows the pipe axial loading profiles in the two experiments. The precipitous drops in the pressure readings indicate the rupture depressurization in the tests. The manufacturer-specified uncertainty on the measured pressures is +/- 0.2% of the full-scale rating of the pressure gages used, which were 5,000 psi max. gages. Therefore, the uncertainty is within +/- 10 psi of the measured pressures shown in Figure 30. This amounts to about double the line thickness in the plots. The uncertainty on the axial loading measurements is ±0.1% of reading, or < ±3 lb. for the highest axial load in the tests. This is far less than the line thickness in the plots. Investigations with the PB model indicated that this magnitude of axial loading uncertainty has negligible effect on failure pressures.

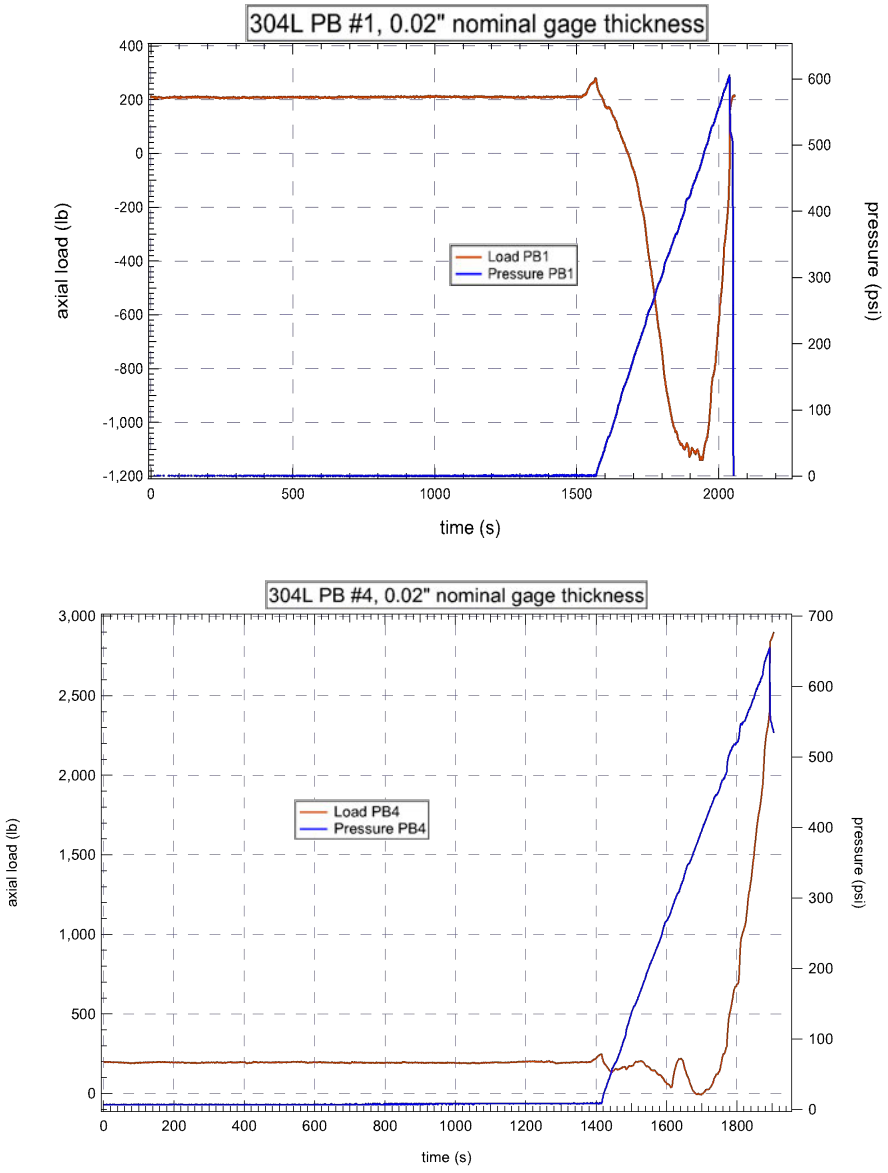


Figure 30. Measured pressure (referenced to scale on right axis), and axial fixture loading (referenced to scale on left axis) in 700C-hold pressurization experiments. Note similar pressure load scales in the two plots, but very different axial loading scales in the two plots.

5.1.2. Pipe Surface Temperature Measurements (TC Mapping/Interpolation Data)

Figures 31 and 32 plot the TC temperature histories in tests PB1 and PB4. These TC data are supplied to the TC mapping/interpolation procedure described in Section 4.2 for reconstructing the temperature fields on the pipe surface in the experiments. These reconstructed fields are the temperature BCs in simulations of the experiments (see next section). The precipitous drops in the temperature data coincide with the pipe bursts.

Table 14 lists linear regression values of the measured temperatures at the pipe front-center location, nominally at the pipe hotspot temperature peaks (TC#4 in figures 21, 31, and 32). The temperatures in Table 14 are obtained by linear regression of TC temperatures over the 60 seconds preceding failure in each experiment. During the last several minutes in each test, electrical noise creates visible oscillations in TC temperatures in figures 31 and 32. The oscillations over the last 60 seconds are on the order of ±10C, so citing a single temperature record at the time of failure could be in error by up to ±10C. Therefore, for each TC we use a linear regression mean-temperature fit over the last 60 seconds. The simulations must be carried out beyond the failure times in the tests because the simulations predict failures at higher pressures and longer times than the actual failures occurred at. No temperature data exists beyond the failure times in the tests, so at all TC locations we evaluate the linear regression mean curves at the times of failure and then hold these temperatures constant for the remainder of the simulations because the temperatures have effectively plateaued at this time, as intended in the tests.

References 31 and 32 discuss various sources of TC temperature measurement uncertainty that are applicable here. The PB experiments used intrinsic TCs of 0.005-in. diameter, which are the most accurate TCs widely available. But small measurement uncertainties exist due to random and systematic sources of error described in Section 5.3. The largest uncertainties are indicated in Table 14, corresponding to the hottest TCs on the pipes. These uncertainties are less than the line thicknesses in figures 31 and 32.

Table 14. Pipe Front-Center Control TC4 Temperature at Failure in 700C-hold Experiments.

Exper.	Measured Temperature at Failure (After Regression)	% Uncertainty in Temperature Measurement	Uncertainty in Temperature Measurement
PB 1	707°C	[-0.25%, +0.5%] of reading in degrees C	[-1.8, +3.5] C
PB 4	711°C	[-0.25%, +0.5%] of reading in degrees C	[-1.8, +3.6] C

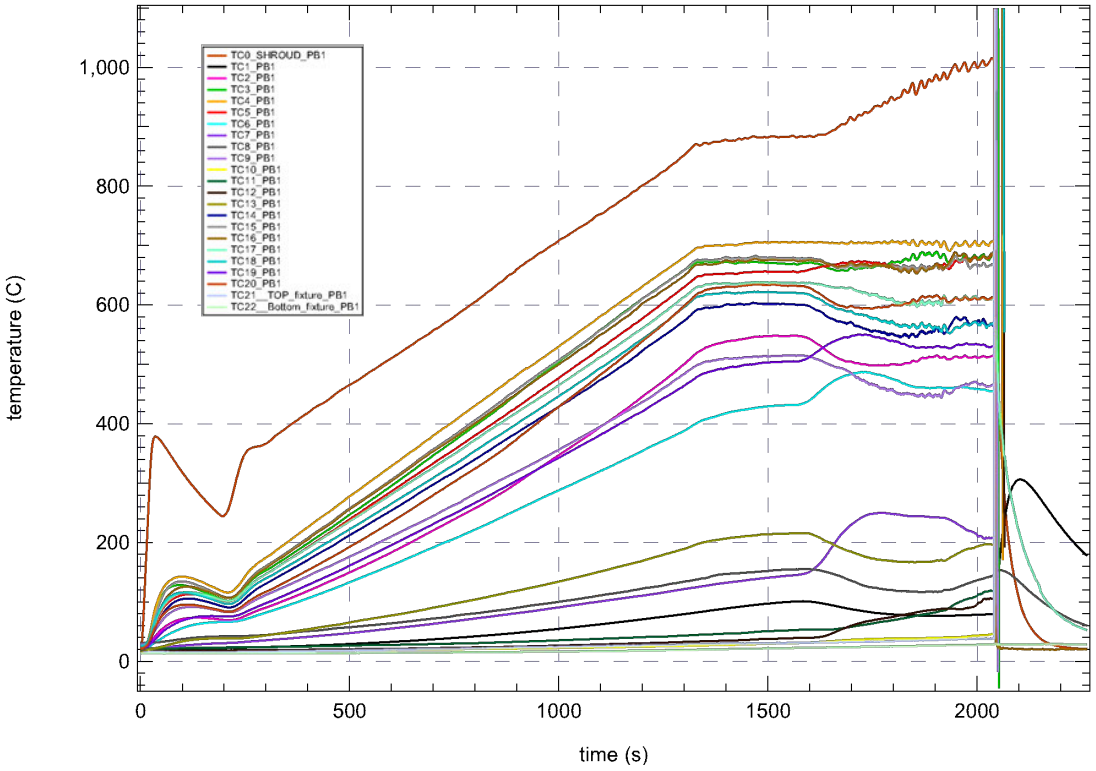


Figure 31. Thermocouple (TC) temperature measurements in experiment PB1. TC numbering in this figure corresponds to numbered locations in Figure 21.

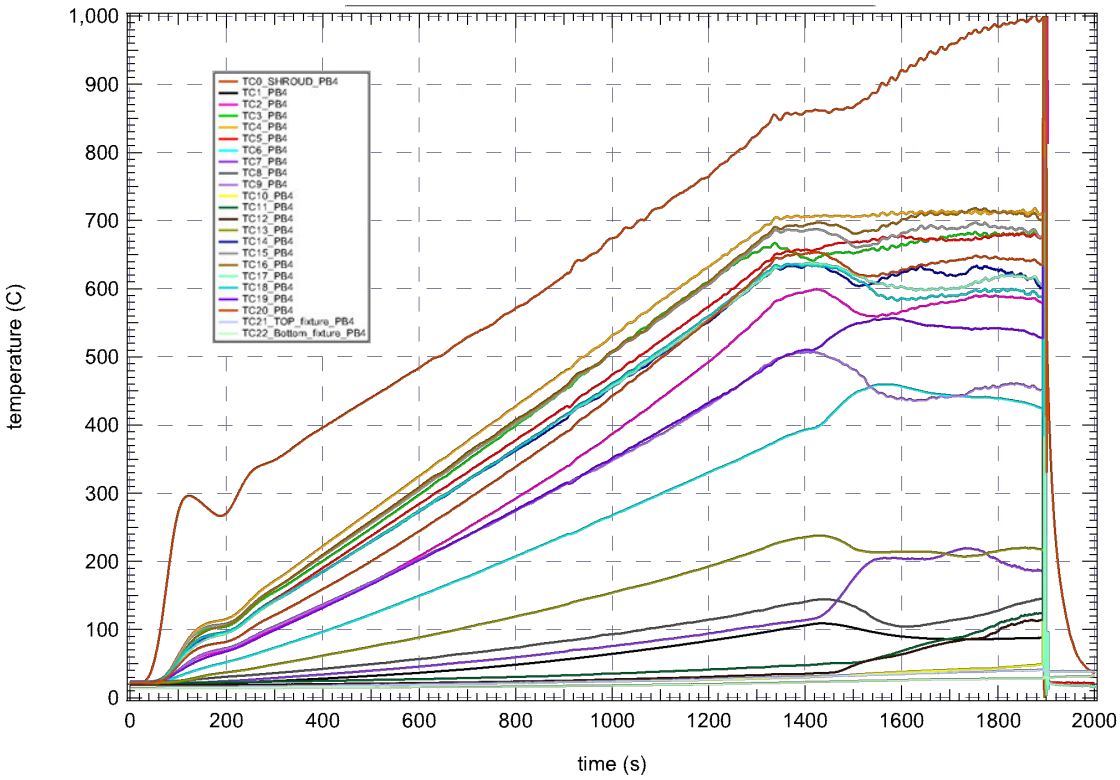


Figure 32. Thermocouple (TC) temperature measurements in experiment PB4. TC numbering in this figure corresponds to numbered locations in Figure 21.

5.1.3. Pipe Wall Thickness Variation

Table 15, above, shows measured minimum and maximum pipe-wall thicknesses for various PB tests. The following measurement procedure was used for each pipe unit. Two V-blocks were set up to support a mandrel. A dial indicator was then used and zeroed on the mandrel. This was the reference for the pipe inside diameter, ID. A pipe test unit was then placed on the mandrel and the dial indicator was placed on the outside diameter (OD) of the reduced section. The dial indicator was then moved along the pipe and the readings were taken. The specimen was then rotated on the mandrel and another set of measurements was taken. This was repeated 15–20 times and the minimum and maximum wall thicknesses were recorded in Table 15.

The machined units held fairly tightly to the nominal 0.02-in. wall thickness specified for the reduced middle length of pipe. All tested pipes were within the allowable tolerances specified on the drawings. PB4 seems slightly skewed to thicker walls than the other units. This might explain at least some of the reason why PB4 had a notably higher failure pressure than PB1. These issues are investigated in detail in Section 5.3.

Table 15. Measured minimum and maximum pipe-wall thicknesses for various PB tests/parts specified. Nominal machining specification thickness was 0.02-in.

Part No.	Wall Thickness
1	.019/.022"
2	.020/.022"
4	.020/.024"
5	.020/.022"
7	.019/.022"
8	.020/.021"
9	.019/.0225"
10	.019/.021"

5.2. Simulations of Validation Experiments and Uncertainty Processing and Roll-Up for Comparisons to Experimental Results

Here we use the PB1 experiment as the reference case for model validation assessment at the 700C-hold conditions. The validation comparisons of experimental results vs. simulation results are built around this reference case. Alternatively, we could use PB4 as the reference; or we could use the averaged experimental conditions and inputs from the PB1 and PB4 replicate experiments, but such averaging is difficult here and in most model validation applications.

The time-varying PB1 temperature, end loading, and pressure boundary conditions shown in figures 30 (top plot) and 31 are applied to a full-geometry PB model. The specification (spec) wall thickness of 0.02 in. for the thin-wall section of the pipe is used, along with dimensions of the thicker sections, as described in Section 3. The model was run with both high strength s-s curves, and low strength variants, as described below in explanatory notes 1 and 3.

A coarse 1tt mesh was used because of the extreme computational demands of this problem. Corrections for mesh size (with associated uncertainty) are described later in this section, working from the mesh refinement study results in Section 3.5.2. The full model was first tried with a 4tt mesh and then a 2tt mesh, but even with 1,200 processors, the progress rates of these simulations

indicated months of runtime would be necessary to get through the multiple simulations required for uncertainty quantification. This was too long for the project schedule. The 1tt-mesh runs took about four weeks (involving four or five restarts) on several hundred processors, which was just within workable limits. The solver settings in Table 7, above, were used. These settings were also used in the mesh refinement study and in the TC temperature mapping/interpolation error characterization discussed in Section 4.4.

Using the full geometry model was necessary due to non-negligible effects of using the non-symmetric temperature distribution in PB1 (see Figure 28, above) vs. the ¼ symmetric mirrored PB1 temperature BCs used in the mesh study. Figure 17 shows the ¼ geometry section used for the ¼ model. The ¼ model and the full model have the same temperature distribution over their common quarter-section region, but the ¼ model assumes symmetry of the temperature distribution about the ¼ model's bottom and side boundaries. So, its solution emulates a full model with ¼ symmetric temperature BCs. The failure pressure for the symmetric temperature distribution (¼ model, 1tt mesh) is 55 psi or ~ 5% higher than the result from the 1tt-mesh full model with non-symmetric PB1 temperature distribution. (The high-strength s-s curves were used in this investigation.)

The results and uncertainties associated with the PB1 reference prediction are summarized in Figure 33. Explanatory notes:

1. The predicted failure pressure at the top-left corner in Figure 33 is from a simulation with the HS s-s curves and the 1tt mesh.

2. A correction to the prediction with the 1tt mesh is made using the results in Section 3.5.2. From the last sentence in that section, corrections of -228 and -336 to the failure pressure calculated on the 1tt-mesh yield upper and lower estimated bounds for the asymptotic grid-converged failure pressure. These correction limits are applied here because the modeled PB1 conditions in Section 3.5.2 only differ from those here by the slight asymmetry of the temperature field, as explained earlier. Although the calculated failure pressures differ non-negligibly for the symmetric vs. non-symmetric temperature fields, it is not anticipated that convergence rates with mesh refinement differ appreciably, but we could not check this assumption.

   Thus, we estimate that the asymptotic grid-converged failure pressure lies within the range [856, 748] psi = 1084 psi (the 1tt-mesh result) + [-228, -336] psi as upper and lower bound corrections. This construction defines the left-most uncertainty bar in Figure 33. This uncertainty will be aggregated with the other uncertainties in Figure 33 in a later step. In the present step, only a nominal reference value within the range [856, 748] psi is selected. A value midway between the upper and lower extremes is an obvious choice, but we select 839 psi and demonstrate that this works just as well. We then parameterize the corrected range [856, 748] psi in terms of the nominal corrected value 839 psi plus an uncertainty range [+17, -91] psi about the nominal value. These numbers are written to the left of the left-most uncertainty bar in Figure 33. Another way of expressing what we have done is to correct the 1tt-mesh result (1084 psi) by a nominal correction of -245 psi (yielding a nominal corrected value of 839 psi), with an uncertainty of [+17, -91] psi about the nominal correction of -245 psi and about the nominal corrected value of 839 psi.

3. Running the model with low-strength s-s curves instead of high-strength curves reduces the calculated failure pressure by 43 psi, from 1,084 psi to 1,041 psi. Applying the nominal correction (from step 2) of -245 psi to the LS result yields a nominal mesh-

corrected value of 1041 psi - 245 psi = 796 psi. This correction assumes that the mesh convergence behavior is similar whether the model has low- or high-strength s-s curves. The nominal corrected LS value of 796 psi coincides with the bottom of the 2nd uncertainty bar from left in Figure 33. This bar, spanning the LS and HS results, has a length of 43 psi, as labeled in the figure. The LS and HS results parameterize the aleatory variability of predicted failure pressures due to stochastic variations in material strength as characterized from the rod tension tests in Section 2.

4. Before the aleatory variability of failure pressure can be predicted, the LS and HS failure pressures must be corrected for errors caused by reconstruction (interpolation) of the pipe temperature field BCs from the spatially sparse TC temperature data in test PB1. A global element of temperature reconstruction error is addressed here. Local elements of temperature reconstruction error are addressed in Section 5.3.

   The global effect of temperature reconstruction errors on computed failure pressure is characterized in Section 4.4.2 for the synthetic temperature distributions in figures 24 and 25. These distributions are similar to test PB1's temperature distribution, as Figure 28 shows. The figure shows that, at the times of failure in the experiments and simulations, the temperature BCs for simulations of test PB1 are bounded between the $\varepsilon = 0.7$ and $\varepsilon = 0.84$ synthetic temperature fields in the critical hotspot region where failure occurs. From Table 12, the temperature interpolation-induced errors for the $\varepsilon = 0.7$ and 0.84 bounding cases are +12.5 psi and +48 psi, respectively, if the simulations are performed with the high-strength s-s curves. Associated bounding corrections to the HS result of 839 psi from note 2, above, are:

   if $\varepsilon = 0.7$, the temperature-interpolation-corrected value is 839 psi - 12.5 psi = 826.5 psi;

   if $\varepsilon = 0.84$, the temperature-interpolation-corrected value is 839 psi - 48 psi = 791 psi.

   The bounding correction for $\varepsilon = 0.7$ is depicted in Figure 33 by the green line segment labeled "TC interp. correction" with the ending value rounded to 827 psi, as labeled. The bounding correction for $\varepsilon = 0.84$ is depicted by the brown line segment that starts at 839 psi and ends at 791 psi (this value is not labeled in the figure).

   For analogous simulations performed with the *low-strength* s-s curves, bounding errors listed in Table 12 are corrected as follows for the LS result of 796 psi from note 3:

   if $\varepsilon = 0.7$, the temperature-interpolation-corrected value is 796 psi - 2.8 psi = 793.2 psi;

   if $\varepsilon = 0.84$, the temperature-interpolation-corrected value is 796 psi - 37.4 psi = 758.6 psi.

   The bounding correction for the low-strength s-s curves and $\varepsilon = 0.7$ is depicted in Figure 33 by the green line segment with ending value 793.2 psi (not labeled in the figure). The bounding correction for $\varepsilon = 0.84$ is depicted by the brown line segment with ending value 758.6 psi, as labeled in the figure.

5. Because the actual PB1 temperature field is effectively bounded by the $\varepsilon = 0.7$ and 0.84 temperature fields (Figure 28), it is reasonable to assume that use of the TC data from the actual PB1 temperature field will yield an interpolation error within the error extremes given by the $\varepsilon = 0.7$ and 0.84 cases—for any s-s curve used in the model. Correspondingly, we assume that an effective emissivity value ε_PB1 for the actual PB1 temperature field lies between these bounding cases such that $0.7 < \varepsilon\_PB1 < 0.84$.

For the limiting case ε_PB1 = 0.7, corrections to the HS and LS simulation results in Figure 33 are indicated by the green line segments introduced in note 4, above. The corrected pressure predictions are 826.5 psi and 793.2 psi, respectively. Figure 34 shows the corresponding tolerance interval scaled from the 700°C experimental failure pressure data in Figure 8. The ends of the scaled TI in Figure 34 are used to define an equivalent Normal PDF of failure pressure variability due to material strength variability characterized in Section 2. The Normal PDF is also shown in purple at center in Figure 33.

For the limiting case ε_PB1 = 0.84, corrections to the HS and LS simulation results in Figure 33 are indicated by the brown line segments introduced in note 4. The corrected pressure predictions are 791 psi and 758.6 psi, respectively. Figure 35 shows the corresponding tolerance interval and normal PDF scaled from Figure 8. The Normal PDF is also shown in red at center in Figure 33.

A normal PDF somewhere between the upper and lower (purple and red) bounding PDFs in Figure 33 would correspond to an actual effective emissivity ε_PB1 that lies between 0.7 and 0.84, which is arguably the case here.

6. Last, the uncertainty of the correction for mesh effects is incorporated. In Figure 33, the mesh correction uncertainty is shown translated to the right-most uncertainty bar in the figure. To aggregate this source of uncertainty with the uncertainty represented by the upper and lower PDFs defined in note 5, linearity and independence are assumed. Essentially, we estimated that the upper and lower PDFs, which are built on the starting reference value of 839 psi (denoted by the dot on the left-most uncertainty bar) would translate up or down as a fixed-shape pair if the reference value (dot) lies at some other position on the uncertainty bar. Thus, the mean separation and the standard deviations of the two PDFs would not change, but the PDF pair would simply translate up and down with the reference value. Note that while we believe this a good approximation, we could not afford to check it.

   Given the assumed linearity and independence, if the reference value (dot) was at the top of the mesh correction uncertainty bar, the PDF pair would be translated upward by 17 psi, per the indicated range at left of the left-most uncertainty bar in Figure 33. Then the upper PDF in the upward translated PDF pair would have a mean increased by 17 psi, making its mean (805psi + 17 psi) = 822 psi. This is shown at upper right in Figure 33.

   At the other extreme, the existing reference value of 839 psi could be translated downward by 91 psi to reach the bottom of the uncertainty bar. Then the lower PDF in the downward translated PDF pair would have a mean decreased by 91 psi, making its mean (768 psi - 91 psi) = 677 psi. This is shown at bottom right in Figure 33. The new extreme upper and lower PDFs comprise a “Level 1 Probability Box” [44]. Their separation signifies lack-of-knowledge epistemic uncertainty regarding the centering of the predicted population of stochastic results (failure pressures).

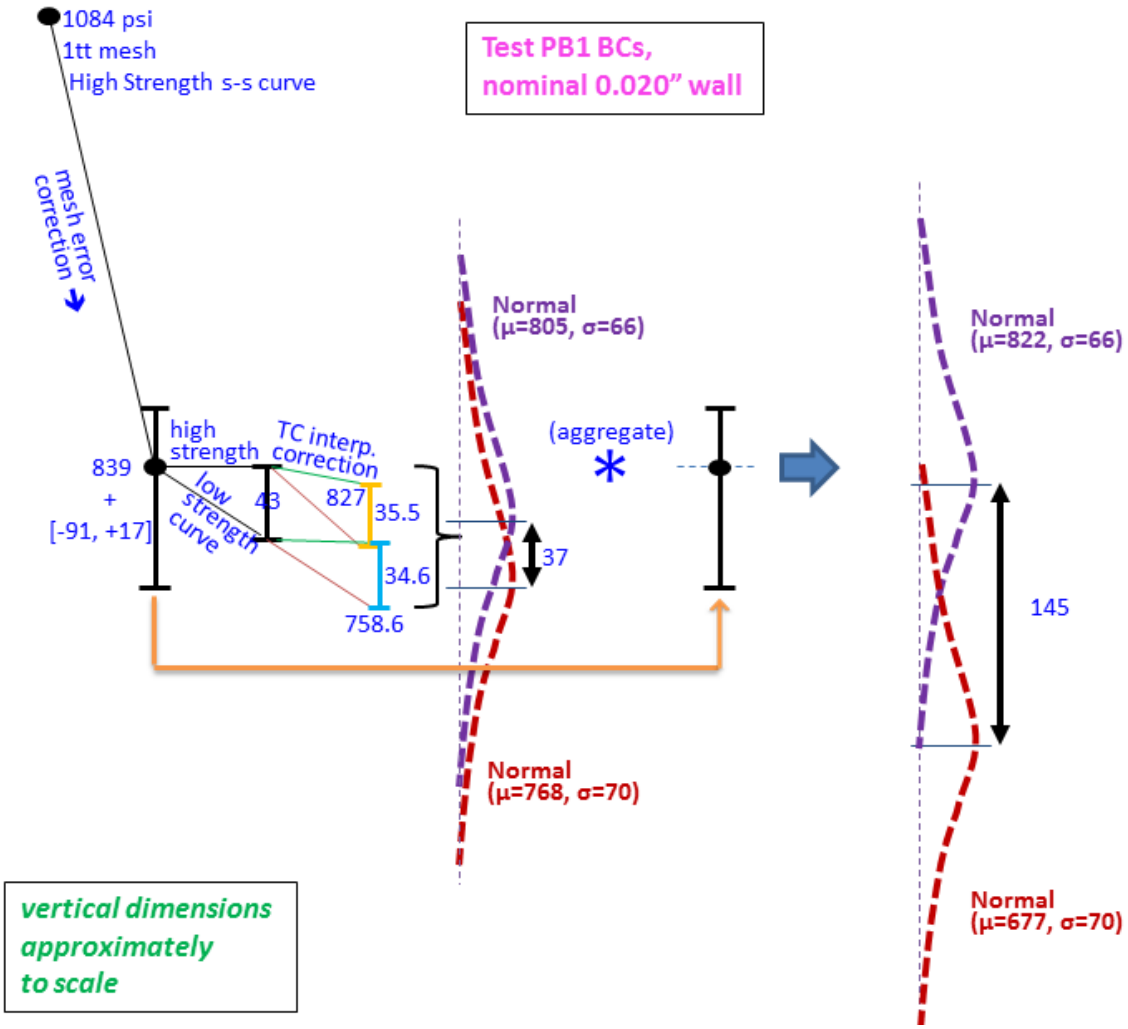


Figure 33. Uncertainty roll-up for simulation results processed for Real Space validation comparisons.

5.3. Processing and Roll Up of Experimental Uncertainties for Model Validation Comparisons

We now discuss the specific manner in which the experimental data is processed for Real Space (RS) validation comparison to the simulation results in Figure 33. An "apples-to-apples" comparison basis must first be established between *predicted* PDF percentiles of failure pressure and *experimentally* derived percentiles of failure pressure. Consequently, here we account for non-traveling experiment-to-experiment random variability and systematic uncertainty that was not accounted for in Section 5.2. In doing so we normalize the experimental data to the following reference conditions input to the simulations:

- 0.02" pipe wall thickness
- PB1 nominal pressure, temperature, and mechanical end-loading boundary conditions.

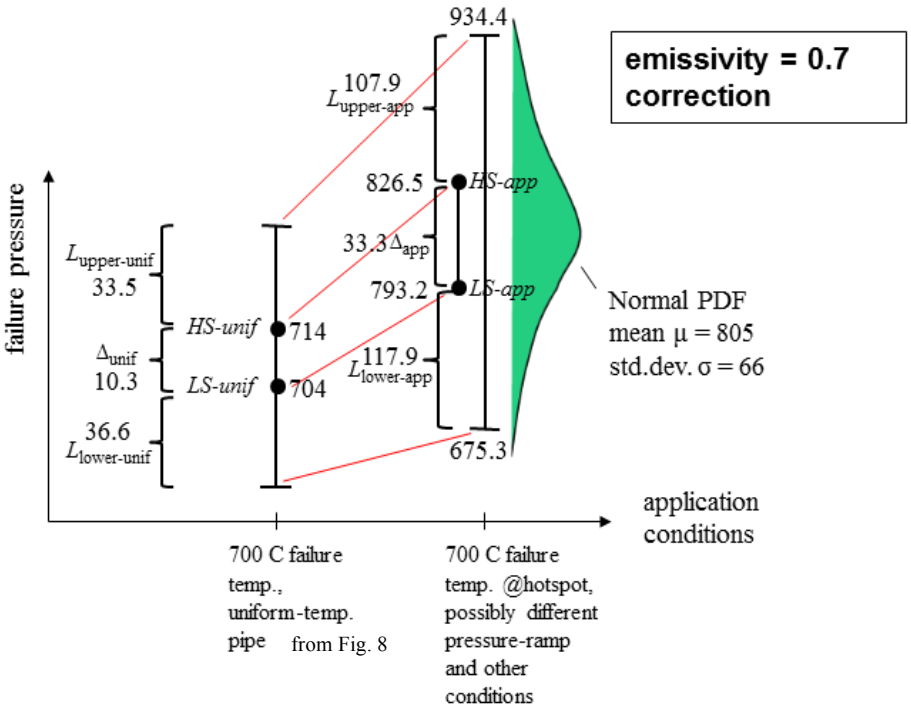


Figure 34. Scaling of Figure 11 failure pressure TI to 95/90 TI and associated Normal PDF for PB1 test conditions and upper-bound (ε = 0.7) correction of temperature field reconstruction error.

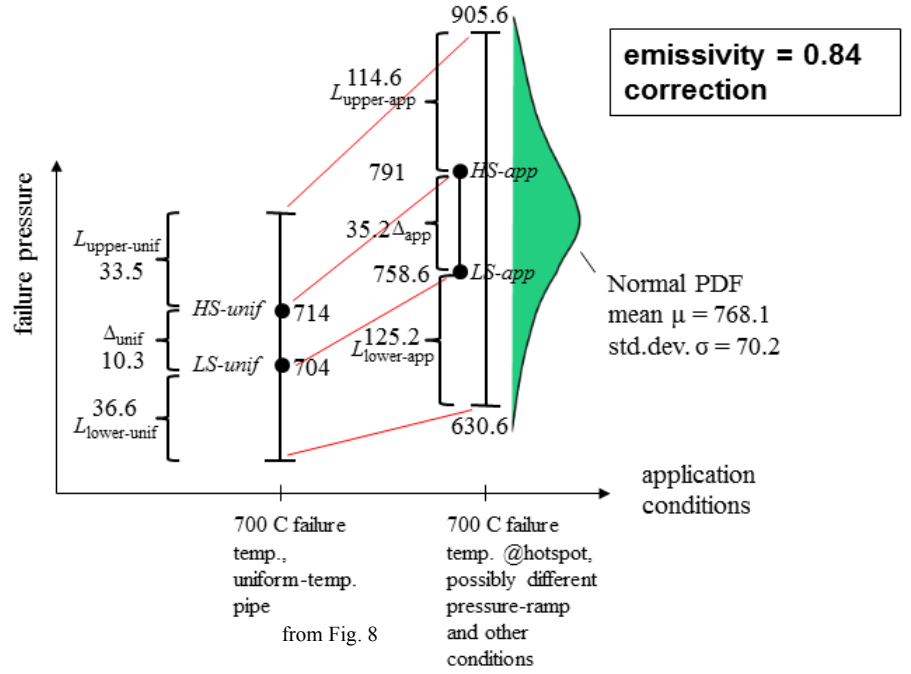


Figure 35. Scaling of Figure 11 failure pressure TI to 95/90 TI and associated Normal PDF for PB1 test conditions and upper-bound (ε = 0.84) correction of temperature field reconstruction error.

We also account for inference uncertainty on estimated percentiles of response from small numbers of experiments. To reduce UQ method complexity and cost, we employ a linearized data normalization process. Examples of higher-order UQ treatment are available in Refs. 28 and 43.

Let w_{act} be the actual pipe wall thickness in test PB1. From Table 13, the failure pressure is $Pfail_{exper,w\_act}$ = 606 psi. (We will account for pressure measurement uncertainty later.) The input-output correspondence (w_{act}, $Pfail_{PB1exper,w\_act}$) can be thought of as the particular point of a function $Pfail_{exper}(w)$ that describes experimental failure pressure as a function of wall thickness:

$$Pfail_{exper,w\_act} = Pfail_{exper}(w=w_{act}). \tag{5.1}$$

A Taylor Series is used to normalize the data to the reference wall thickness w_{ref} = 0.02" used in the failure pressure calculations in section 5.2:

$$\begin{aligned} Pfail_{exper,w\_ref} &= Pfail_{exper}(w_{ref}) = Pfail_{exper}(w_{act}+[w_{ref}-w_{act}]) \\ &= Pfail_{exper}(w_{act}) + \frac{d(Pfail_{exper})}{d(w)}\bullet(w_{ref}-w_{act}) + \text{HOT} \end{aligned} \tag{5.2}$$

Here, HOT stands for "higher order terms." The actual wall thickness w_{act} is a small perturbation from the machining spec wall thickness of w_{ref} = 0.02", so the HOT contribution is relatively small and we retain just the 1st-order term:

$$Pfail_{exper}(w_{ref}) \approx Pfail_{exper}(w_{act}) + \frac{d(Pfail_{exper})}{d(w)}\bullet(w_{ref}-w_{act}) \tag{5.3}$$

The above equation is used to approximately normalize the PB1 failure pressure to a reference wall thickness w_{ref} that differs from the actual wall thickness w_{act} at the point where failure onset occurs in the experiment. The derivative term $\frac{d(Pfail_{exper})}{d(w)}$ can be estimated either with the model or by using available test data from purposefully different wall-thickness pipes exposed to temperature and pressure conditions similar to PB1. This is further discussed below.

If we normalize all replicate tests (e.g., PB1 and PB4) to the same reference wall thickness as in the failure pressure predictions, then we are on the same apples-to-apples basis of wall thickness to compare the predicted and experimental PDFs of failure pressure. We must similarly normalize the experimental failure pressure results to the same basis of pressure loading and temperature boundary conditions before apples-to-apples validation comparisons can be made. Generalizing the first-order Taylor Series approximation (Eq. 5.3) to multiple experimental inputs x_i, the normalization adjustments with respect to each of the individual inputs superpose (add linearly):

$$Pfail_{exper}(\vec{x}_{ref}) \approx Pfail_{exper}(\vec{x}_{act}) + \sum \frac{\partial(Pfail_{exper})}{\partial(x_i)}\bullet(x_{i\_ref}-x_{i\_act}). \tag{5.4}$$

Any significant measurement and processing uncertainties associated with the experimental results of interest (here, $Pfail_{exper}(\vec{x}_{act})$) are combined with any significant uncertainties in the partial derivatives and in the actual input values x_{i_act} on the right-hand side (RHS) of Eq. 5.4. All these uncertainties are combined appropriately to yield an uncertainty on the normalized failure pressure on the left-hand side of the equation. Any correlation between the uncertainties on the RHS of the equation must be accounted for. A simple spreadsheet-based approach is

demonstrated later. This approach would be a first-order or linear UQ method if only incorporating uncertainties of the inputs x_{i_act} and of measured outputs (first term on RHS) in Eq. 5.4. But, including uncertainty of the partial derivatives makes this a nonlinear UQ method; it will be referred to as a "1st-order+" or "linear+" UQ method.

Equation 5.4 with linear UQ handles non-traveling systematic uncertainty of experimental inputs x_i in the same way as two very differently derived linearization approaches presented in References 6 and 33. This is reassuring. Moreover, for these conditions and a further condition of only probabilistic uncertainty involved, equivalence to the linear UQ version of the ASME VV20 approach [22] can be shown.

Extensions of Eq. 5.4 can be written with higher-order approximations of the RHS, as demonstrated in Refs. 28 and 43. Equation 5.4 and higher-order extensions are applicable to random and/or systematic non-traveling uncertainties in experiments supporting model calibration or validation. When only systematic uncertainties are significant in the replicate experiments, the manner of uncertainty treatment and presentation of results in the RS approach can often be simplified, see Appendix C. The simplification allows a streamlined approach (e.g., Romero et al [28]) vs. the spreadsheet methodology presented below.

5.3.1. Normalization of Reference Test Results

The next step is to write Eq. 5.4 for the reference test that we are normalizing all the test results to. The reference test is PB1, and we normalize it to the PB1 *nominal* input conditions used in the simulations in Section 5.2. We will later write normalization equations for the other replicate experiments.

$$Pfail_{\text{PB1}}(\vec{x}_{\text{nomPB1}}) \approx Pfail_{\text{PB1}}(\vec{x}_{\text{actPB1}}) + \sum \frac{\partial(Pfail_{\text{PB1}})}{\partial(x_i)} \bullet (x_{\text{i\_nomPB1}} - x_{\text{i\_actPB1}}) \qquad (5.5)$$

We only know the failure pressure in PB1 to within the uncertainty of the pressure measurement in the test (see Table 13). Therefore, $Pfail_{\text{PB1}}(\vec{x}_{\text{actPB1}})$ on the RHS of Eq. 5.5 has an uncertainty $U[Pfail_{\text{PB1}}(\vec{x}_{\text{actPB1}})]$ = 606 psi ± 10 psi = [596, 616] psi.

Using expert judgment to arrive at an affordable number of uncertainties to include in the analysis, the two most substantial terms—by far—in the summation in Eq. 5.5 involve differences between nominal and actual wall thickness, and nominal and actual temperature, at the location of failure initiation in PB1. Thus, Eq. 5.5 is rewritten as:

$$Pfail_{\text{PB1}}(\vec{x}_{\text{nomPB1}}) \approx Pfail_{\text{PB1}}(\vec{x}_{\text{actPB1}}) + \frac{\partial(Pfail_{\text{PB1}})}{\partial(w)} \bullet (w_{\text{nomPB1}} - w_{\text{actPB1}}) + \frac{\partial(Pfail_{\text{PB1}})}{\partial(T@fail\_point)} \bullet (T_{\text{nomTC4-PB1}} - T_{\text{act@fail\_point-PB1}} \qquad (5.6)$$

Recall that the PB1 TC data provides the nominal temperature distribution on the pipe for the PB1 simulations, and the effect of any errors/uncertainties in reconstructing the temperature BCs from the *nominal* TC data have already been accounted for in Section 5.2. However, the nominal TC data led to temperature BC errors in two other respects:

- Temperature measurement errors of the TCs still need to be accounted for.
- The actual temperature distribution may have a peak temperature that is hotter than at the TC4 location (the hottest TC location in all tests).

It is presumed that failure initiates at the location (point) where the combination of local temperature, wall thickness, and material strength determines the weakest point on the pipe wall; that is, the lowest resistance to pressure loading. Yielding initiates at that point and proceeds until failure occurs there. Indeed, the model simulations, which have uniform wall thickness in the reduced-thickness 0.02-in. wall section, predict that initial yielding and then failure occurs at the hottest point on the pipe (see Figure 27, above). We assume that in the tests the failures initiate within a close vicinity of the hottest TC, TC4. An investigation by Antoun and Connelly [16] suggests this. For the monitored specimen, failure initiates at a location indistinguishable from the TC4 location, where the pipe wall first splits upward and downward, and then circumferentially along the top and bottom thickness-transition shoulders as the "butterfly wings" are created (see Figure 1, above). The approximate symmetry of the final geometries in Figure 1 also implies that failure initiation is at the heated front center of the pipe. So we use TC4 as a reference temperature, above which we propose 15°C as a reasonable maximum possible perturbation to address item 2) in the paragraph immediately above.

TC measurement errors/uncertainties in PB1and PB4 are much more important for the hotspot TC4 than for the other TCs. Therefore, we consider only TC4 in the following. We note that here propagation of all the TCs' uncertainties (if significant) to evaluate their impact involves a spatially coupled problem in the spatially distributed TCs, so evaluating each individual TC error effect via a separate term in the summation in Eq. 5.5 would not be adequate or cost effective. See Appendix A for an example evaluation of the effect of spatially correlated errors of the TC measurements and how that effect would be folded into the larger validation analysis.

The wall thickness uncertainty in PB1 is modeled as the interval range given in Table 15:

$$\mathrm{U}[w_{\mathrm{actPB1}}] = [0.019, 0.022] \text{ inch.} \tag{5.7}$$

The term $\frac{\partial(Pfail_{\mathrm{PB1}})}{\partial(w)}$ in Eq. 5.6 is approximated by considering the experimental and model simulation results in Figure 36. The figure shows predicted failure pressures (listed in Table 16) for test PB1 nominal boundary conditions and the three pipe wall thicknesses listed in Table 15. The model used was the full-geometry 1tt-mesh model used for the predictions in Figure 33, except for changes to wall thickness. Thus, the value in Figure 36 and Table 4 for the nominal 0.02-in. wall is the same as in Figure 33. Failure pressures at the three thicknesses were all corrected downward to approximate mesh-converged results by the same nominal correction of -245 psi shown in Figure 33. The regression line through the simulation results in Figure 36 has a slope of 42,286 psi/inch-wall-thickness and a y-intercept of -16 psi when extrapolated to zero wall thickness. This extrapolated value is not unreasonable given that an exact value of zero pressure would be required for failure when wall thickness goes to zero.

Table 16. Predicted failure pressures for nominal PB1 conditions and pipe wall thicknesses listed.

Pipe Wall Thickness (inches)	Predicted Failure Pressure (Approximately Corrected for Mesh Discretization Effects)
0.019"	780 psi
0.020"	839 psi
0.024"	997 psi

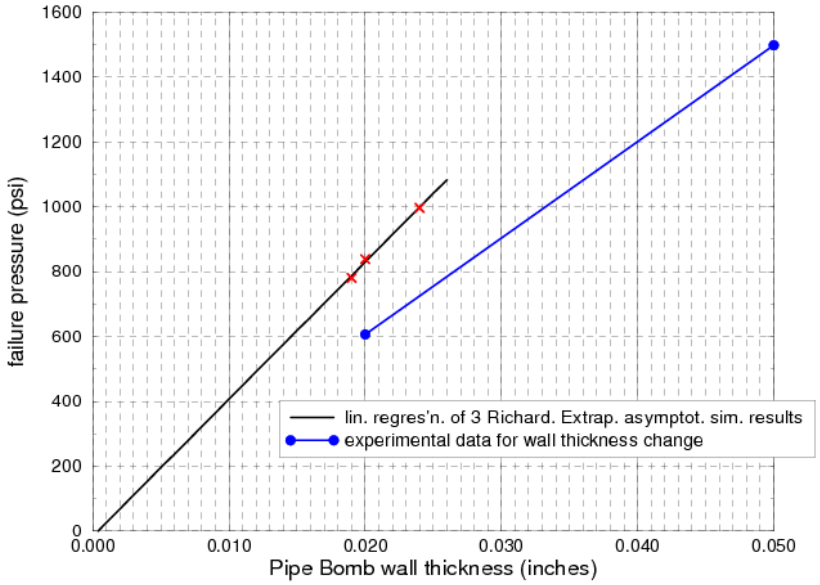


Figure 36. Failure pressures vs. wall thickness from simulations and experiments.

The experimental data line in Figure 36 connects failure pressures in the PB1 and PB3 tests (Table 17). The experimental line has a slope = 29,733 psi/inch wall-thickness and a y-intercept of 11.3 psi when extrapolated to zero wall thickness. Test PB3 [16] had a 0.05-inch thick wall and a hotspot TC4 temperature of ~700°C at failure, similar to PB1. However, the pressure and temperature ramping conditions were somewhat different in the PB1 and PB3 tests. Furthermore, the experimental slope of failure pressure vs. wall thickness is subject to much uncertainty because it is based on only one test with a 0.05-in. wall and one 0.02-in. wall result. (Table 13, above, gives a glimpse of the magnitude of variation that exists among failure pressures even for nominal replicate tests and wall thickness.) It would be best to average over many tests at different wall thicknesses to obtain an experimental slope of failure pressure vs. wall thickness, but PB3 is the only available test with a wall thicknesses other than 0.02-in. Test PB1 had the closest temperature conditions at failure to test PB3, so is the best one to plot with PB3 in Figure 36. Test PB4, at failure, also has close temperature conditions to PB3, but PB4 results from Table 15 appear to have a wall thickness not as close to the 0.02-in. value used in Figure 36 and Table 17.

Table 17. Experimental failure pressures for nominal PB1 conditions and pipe wall thicknesses listed.

Pipe Wall Thickness	Measured Failure Pressure
0.02”	606 psi
0.05”	1498 psi

Given the potential errors in both the simulated and experimental slopes, the following route is taken. The differences in failure pressures in the PB1 and PB4 replicate tests is 49 psi, from Table 13. This is used as a scale of experimental variability to work with. Perturbing the experimental failure pressures in Table 17 by ±49 psi gives four combinations, from which the lowest and highest slopes are 26,467 and 33,000 psi/inch wall-thickness. If we liberally double the perturbations to ±100 psi at both wall thicknesses, then the lowest and highest slopes are 23,067

and 36,400. The upper value is about 15% less than the slope in Figure 36 from the model simulations. We choose to use the higher value of 42,286 psi/inch from the simulations as a liberal upper value in our UQ analysis. We use the experimentally based liberal lower value 23,067 psi/inch as a lower limit.

$$U[\frac{\partial(Pfail_{PB1})}{\partial(w)}] = [23{,}067,\ 42{,}286] \text{ psi/inch-wall-thickness.} \quad (5.8)$$

The term ($T_{\text{nomTC4-PB1}}$ - $T_{\text{act@fail\_point-PB1}}$) in Eq. 5.6 captures the difference between the nominal hotspot temperature (reading from TC4 in Figure 21), and the actual temperature at the location of failure initiation in the PB1 test. With this term and the partial derivative $\frac{\partial(Pfail_{PB1})}{\partial(T@fail\_point)}$ we normalize between the nominal temperature conditions modeled, and the actual temperature conditions in the test. For reasons mentioned earlier, we normalize only for local temperature differences at the failure initiation point (close vicinity of TC4) and ignore measurement uncertainties of the other TCs. The difference ($T_{\text{nomTC4-PB1}}$ - $T_{\text{act@fail\_point-PB1}}$) is separated into two components:

A: TC4 measurement error—the difference between the TC4 reading value $T_{\text{nomTC4-PB1}}$ and the actual pipe temperature at that location, $T_{\text{actTC4-PB1}}$;

B: TC4 location related error—difference between the TC4 temperature after being corrected for any measurement error, $T_{\text{actTC4-PB1}}$, and the temperature where failure occurs, $T_{\text{act@fail\_point-PB1}}$.

The following identity is written in terms of components A and B:

$$T_{\text{nomTC4-PB1}} - T_{\text{act@fail\_point-PB1}} = (T_{\text{nomTC4-PB1}} - T_{\text{act@TC4-PB1}}) + (T_{\text{act@TC4-PB1}} - T_{\text{act@fail\_point-PB1}})$$

$$= \text{TC4 measurement error} + \text{TC4 location related error} \quad (5.9)$$

Component A of failure-temperature error itself has several components:

$$\text{TC4 measurement error} = T_{\text{nomTC4-PB1}} - T_{\text{act@TC4-PB1}} = (T_{\text{nomTC4-PB1}} - T_{\text{ofTC4-PB1}}) + (T_{\text{ofTC4-PB1}} - T_{\text{act@TC4-PB1}}). \quad (5.10)$$

Here, $\{T_{\text{nomTC4-PB1}} - T_{\text{ofTC4-PB1}}\} = \Delta T_{\text{meas-TC/DAQ}}$ is the difference or error between the TC's actual temperature, $T_{\text{ofTC4-PB1}}$, and the temperature $T_{\text{nomTC4-PB1}}$ reported by the data acquisition system (DAQ). This error is caused by inaccuracies associated with the TC transducer itself, the calibration standard and procedure used to calibrate the TCs if calibration is performed, and the DAQ. The combined TC/calibration/DAQ errors are typically very small. Uncertainty on the combined error is estimated from information from Nakos [31] as:

$$U[\Delta T_{\text{meas-TC/DAQ}}] = U[T_{\text{nomTC4-PB1}} - T_{\text{ofTC4-PB1}}] = [-0.25\%,\ 0.25\%] \text{ of TC4 reading in °C.} \quad (5.11)$$

$T_{\text{ofTC4-PB1}} - T_{\text{act@TC4-PB1}} = \Delta T_{\text{meas-contact}}$ is the difference or error between the TC's actual temperature, $T_{\text{ofTC4-PB1}}$, and the temperature $T_{\text{act@TC4-PB1}}$ of the pipe surface at the TC4 location.

This error is caused by contact resistance between the surface and the attached TC, and by convective and radiative conditions affecting heat losses from the TC bead or attached wire tips if using an intrinsic TC. The PB tests used intrinsic TCs with very small wire tip diameters of 0.005-in. Uncertainty on the temperature difference between the TC and the surface point it is attached to is estimated from information from Nakos et al. [32]:

$$U[\Delta T_{\text{meas-contact}}] = U[T_{\text{ofTC4-PB1}} - T_{\text{act@TC4-PB1}}] = [0., 0.25\%] \text{ of TC4 reading in °C.} \quad (5.12)$$

The uncertainty ranges from zero to + 0.25% because the TC4 leads are between the surface being measured and the heating plate. Therefore, at the attachment location the leads will be hotter than the surface, so the error will be positive. Errors ΔT_{meas} in equations 5.11 and 5.12 are defined to have positive values that, when taken alone, cause an overestimate of the true surface temperature at the TC4 location.

Component B of failure-temperature error in Eq. 5.9 is due to TC4's location being potentially different from where failure occurs:

$$T_{\text{act@TC4-PB1}} - T_{\text{act@fail\_point-PB1}} = \Delta T_{\text{TC4location}} \quad . \quad (5.13)$$

We surmised earlier that the actual peak temperature on the pipe could be a small distance away from TC4, and as much as 15°C above the true pipe surface temperature at TC4. Then, Eq. 5.13 yields $\Delta T_{\text{TC4location}}$ = -15°C as the largest negative value we estimate can occur.

But in the real experiment, failure will not necessarily occur at the hottest point on the pipe, due to wall thickness and material strength spatial variations. Instead, it occurs at the location of the weakest combination of thickness, temperature, and material strength. Therefore we must also consider possible failure temperatures lower than the temperature $T_{\text{act@TC4-PB1}}$ at TC4. It is more difficult to propose a reasonable lower bound for the possible true failure temperature $T_{\text{act@fail\_point-PB1}}$. We simply guess that failure could occur at a point that is as much as 15°C below the pipe temperature at the TC4 location. Then, Eq. 5.13 yields a maximum value of 15°C for $\Delta T_{\text{TC4location}}$. So:

$$U[\Delta T_{\text{TC4location}}] = U[T_{\text{act@TC4-PB1}} - T_{\text{act@fail\_point-PB1}}] = [-15, +15]°C. \quad (5.14)$$

Considering both components A and B of failure-temperature error in Eq. 5.9, we have:

$$\begin{aligned} T_{\text{nomTC4-PB1}} - T_{\text{act@fail\_point-PB1}} &= (T_{\text{nomTC4-PB1}} - T_{\text{act@TC4-PB1}}) + (T_{\text{act@TC4-PB1}} - T_{\text{act@fail\_point-PB1}}) \\ &= \text{TC4 measurement error} + \text{TC4 location related error} \\ &= (\Delta T_{\text{measTC/DAQ}} + \Delta T_{\text{meas-contact}}) + \Delta T_{\text{TC4location}}. \quad (5.15) \end{aligned}$$

Failure pressure difference due to different nominal and actual wall thicknesses in the PB1test is approximated in Eq. 5.6 by $\frac{\partial(Pfail_{\text{PB1}})}{\partial(w)} \bullet (w_{\text{nomPB1}} - w_{\text{actPB1}})$, whereas failure pressure difference due to different local material temperatures is approximated by (from equations 5.6 and 5.15):

$$\frac{\partial(Pfail_{\text{PB1}})}{\partial(T@fail\_point)} (\Delta T_{\text{meas-TC/DAQ}} + \Delta T_{\text{meas-contact}} + \Delta T_{\text{TC4location}}). \quad (5.16)$$

The partial derivative in this equation is approximated by the slope of the curve shown in Figure 8, evaluated at the temperature of the failure location. The slopes are calculated from the means

of the data at the various temperature levels in Table 3. The slope in Figure 8 between the data means at 600°C and 700°C is -1.71 psi/°C. The slope between the data means at 700°C and 800°C is -2.61 psi/°C. Averaging these two slopes gives -2.16 psi/°C, which is the same value that would come from central-differencing for a second-order approximation to the slope at 700°C. The temperature perturbations in equations 5.6 and 5.16 are relative to the PB1 nominal failure temperature of 706°C, and applicable perturbations (to be established later) are within about [-30, +80] of 700°C. Given all this information, the previously mentioned values of -1.71 psi/°C and -2.61 psi/°C are used as uncertainty extremes about the nominal 700°C value of -2.16 psi/°C.

So:

$$\mathrm{U}\left[\frac{\partial(Pfail_{\mathrm{PB1}})}{\partial(T@fail\_point)}\right] = -[1.71, 2.61] \text{ psi/°C.} \tag{5.17}$$

With the uncertainty descriptions of the applicable terms on the RHS of Eq. 5.6, we are now ready to sample the uncertainties to estimate the uncertainty on the LHS of Equation 5.6. This uncertainty, $\mathrm{U}[Pfail_{\mathrm{PB1}}(\vec{x}_{\mathrm{nomPB1}})]$, is the uncertainty of the PB1 failure pressure when normalized to the nominal experimental conditions input to the model simulations in Section 5.2. We obtain samples j=1,J of the uncertainty $\mathrm{U}[Pfail_{\mathrm{PB1}}(\vec{x}_{\mathrm{nomPB1}})]$ from realizations of the uncertainties on the RHS terms in Eq. 5.6, as follows.

Realization j:

$$\begin{aligned} Pfail_{\mathrm{PB1}}(\vec{x}_{\mathrm{nomPB1}})_{\,\mathrm{j}} \approx\ & Pfail_{\mathrm{PB1}}(\vec{x}_{\mathrm{actPB1}})_{\,\mathrm{j}} + \left\{\frac{\partial(Pfail_{\mathrm{PB1}})}{\partial(w)} \bullet \Delta w_{\mathrm{PB1}}\right\}_{\mathrm{j}} \\ & + \left\{\frac{\partial(Pfail_{\mathrm{PB1}})}{\partial(T@fail\_point)}\left(\Delta T_{\text{meas-TC/DAQ}} + \Delta T_{\text{meas-contact}} + \Delta T_{\text{TC4location}}\right)\right\}_{\mathrm{j}} \end{aligned} \tag{5.18}$$

where the Δ terms have already been defined, except for:

$$\Delta w_{\mathrm{PB1}} = w_{\mathrm{nomPB1}} - w_{\mathrm{actPB1}}. \tag{5.19}$$

Spreadsheet Illustration of Generating and Combining Uncertainty Realizations

A spreadsheet is a convenient way to do the sampling and processing of the realizations. From Eq. 5.18 we populate a spreadsheet as illustrated in tables 18 and 19. In Table 18, the quantities in the yellow highlighted columns 2, 5, 8, and 10 designate correlated uncertainties with the same-numbered columns for other tests to be described later. All uncertainties designated by interval ranges …] in tables 18 and 19 are sampled from using uniform distributions, but interval interpretations are kept, as discussed next.

The uncertainties in the present application are defined as interval ranges. Both interval and probabilistic (PDF) uncertainties have been treated in other applications [25, 28, 29]. A different treatment of interval uncertainties from Romero, Luketa, and Sherman [29] used interval propagation and aggregation techniques to deal with interval uncertainties over the monotonic uncertainty space in that problem. But that method of aggregation almost certainly gives a grossly exaggerated estimate of the uncertainty when more than a couple of interval uncertainties are present. This is because combinations of the extreme values of the interval uncertainties present an increasingly remote possibility as the number of interval uncertainties rises.

Instead, we use the sensibility that the validation conclusions should not be driven by the remote possibilities of compounded extremes of the interval uncertainties. Rather, we use uniform PDF representations of the interval uncertainties and propagate them probabilistically via the spreadsheet. If sufficient sampling is performed, then the resulting density function (DF) will have endpoints that coincide with the endpoints from an interval UQ treatment. However, the DF will not weight the very remote extreme combinations as prominently as an interval treatment does. But we do not interpret the resultant DF as a probability density; rather, we use the DF to arrive at a moderated magnitude of interval uncertainty for the validation assessment (see Section 5.4).

A similar moderating approach is used by Romero, Rutherford, and Castro [28], in which both interval and probabilistic uncertainties are present and are propagated in a segregated fashion. In Reference 25 the intervals were simply treated as uniform PDFs and comingled with the probabilistic uncertainties in propagation. The treatment in Ref. 28 is now preferred when both interval and probabilistic uncertainties are present.

Care should be exercised with random number generator seeds and sequences in the spreadsheet so that the columns of random samples in Table 19 (and Table 21 and tables B.4 and B.6 in Appendix B) are independent of each other unless correlation is intended and purposely imposed. For the particular comparisons to be made in Section 5.4 we recommend that at least 1000 realizations be taken ($J \geq 1000$ in tables 19, etc.). In the following we use J=1000. In a similar problem [43], a convergence study of column average and standard deviation results vs. the number of samples, J, revealed that 10,000 samples were needed for convergence of these results to within about 1% error. It is recommended that this type of convergence study normally be conducted, although not done here.

Column A12 of Table 19 contains the realizations of uncertainty of PB1 failure pressure, $U[Pfail_{PB1}(\vec{x}_{nomPB1})]$, when normalized to the nominal PB1 experimental conditions input to the model simulations in Section 5.2. These realizations are represented by histograms in Figure 37, along with realizations from similar normalizations of other tests. The other tests and their normalized results are discussed later.

Statistics at the bottom of Table 19 summarize the impacts of various uncertainties in the normalization of PB1 failure pressure. The green box in column A6 shows the average adjustment (-17.8 psi) of PB1 experimental failure pressure normalized for possible differences from the nominal 0.02-in. wall thickness used in the failure pressure predictions (Section 5.2). The test PB1failure pressure adjustment is, on average, negative or downward because the PB1 wall thickness uncertainty from Table 15 yields, on average, larger wall thicknesses than the nominal 0.02-in. Therefore, the average normalization of PB1 failure pressure is a reduction in pressure to coincide with an (on average) thinner 0.02-in. wall that puts the experimental failure pressure on par with the 0.02-in wall in the simulations. Thus, with respect to wall thickness, the predicted and experimental failure pressures are put on an apples-to-apples basis for validation comparisons.

Table 18. Spreadsheet formulas for uncertain experimental quantities in Equation 5.18 for test PB1.

realization j	$\Delta Pfail_{measPB1,j}$ = [-10, 10] psi	$w_{actPB1,j}$ = [0.019, 0.022] inches	$\Delta w_{PB1,j} = w_{nomPB1} - w_{actPB1,j}$ = 0.02" – A3	$\frac{\partial(Pfail_{PB1})}{\partial(w)}_j$ = [23,067 , 42,286] psi/inch	= (A4*A5)$_j$ psi	$\Delta T_{meas\text{-}TC/DAQ,j}$ = [–0.0025*707, +0.0025*707] °C	$\Delta T_{meas\text{-}contact,j}$ = [0., +0.0025*707] °C	$\Delta T_{TC4location,j}$ = [–15, 15] °C	$\frac{\partial(Pfail_{PB1})}{\partial(T@fail\_point)}^j$ = [–2.16, –1.71] psi/°C	= (A7 + A8 + A9)$_j$ *A10$_j$ psi	= 606 psi + (A2 + A6 + A11) psi = $Pfail_{PB1}(\vec{x}_{nomPB1})_j$
A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
1											
2											
…	…	…	…	…	…	…	…	…	…	…	…
J											

Table 19. Spreadsheet realizations for uncertain experimental quantities in test PB1.

	A2=ΔPmeas	A3=PB1wall	A4=Δ0.02"	A5=∂P/∂w	A6 = A4*A5	A7=ΔTC_DAC	A8=ΔTC_cntc	A9=ΔTC_loca	A10=∂P/∂T	A11=A10*(A7	A12=606+A2+
1	7.76	0.02135	-0.00135	30701.34	-41.57	1.00	0.52	-4.65	-2.07	6.49	578.68
2	-5.88	0.01928	0.00072	37837.13	27.38	0.23	1.18	-8.51	-1.87	13.27	640.77
3	-0.69	0.02135	-0.00135	26116.40	-35.35	0.39	1.61	-13.98	-1.83	21.89	591.86
999	-8.55	0.01958	0.00042	37867.63	15.98	1.58	0.10	-2.59	-2.15	1.97	615.41
1000	6.28	0.01943	0.00057	29062.56	16.56	-1.25	0.06	-9.80	-1.99	21.92	650.75
	avg	avg	avg	avg	avg	avg	avg	avg	avg	avg	avg
	0.00	0.02056	-0.00056	32361.71	-17.82	0.06	0.91	0.07	-1.94	-1.99	586.19
	stdev				stdev					stdev	stdev
	5.76				28.13					17.09	32.78
											max
											671.86
											min
											502.78

The green box in column A11 shows the average bias adjustment of -2 psi when test PB1 failure pressure is normalized for possible differences from the nominal temperature field and failure location used in the simulations.

The green box in column A2 shows a zero average bias adjustment when pressure measurement error possibilities are accounted for. The zero average occurs because the measurement uncertainty [-10, +10] psi is symmetric about the nominal value of zero measurement error.

The average adjustment of -17.8 psi for wall thickness, added with the average adjustment of -2 psi for temperature normalization, yields a combined average adjustment of -19.8 psi from the nominal measured failure pressure of 606 psi in Table 13. The resulting average normalized failure pressure is 586.2 psi, as listed in the pink box at the bottom of Table 19. In terms of sensitivity ranking, the average adjustment associated with wall thickness normalization is about 9 times the average adjustment due to temperature normalization.

Possible wall thickness differences from nominal lead to variations in normalized failure pressure characterized by a standard deviation of 28.1 psi, as listed in the grey box in column A6. Possible failure temperature differences from nominal yield a substantial but smaller standard deviation of 17.1 psi in normalized failure pressure per the grey box in column A11. Possible pressure measurement error yields a much smaller standard deviation of 5.8 psi in failure pressure per the grey box in column A2. These three sources of variance combine to yield a variance characterized by the standard deviation in the pink box at the bottom of Table 19. For our linear propagation of the independent sources of uncertainty, the individual variance effects add according to the sum of the squares of their standard deviations. The square root of this sum is 33.4 psi. This closely agrees with the standard deviation 32.8 psi in the pink box, calculated directly from the realizations in column A12. The discrepancy between standard deviation calculated these two ways is expected to approach zero as the number of realizations J approaches infinity.

In terms of sensitivity ranking, the variance contributed by wall thickness normalization is 71% of the total variance. This is a much larger contribution than variance contributed by temperature normalization (26%), and variance contributed by pressure measurement uncertainty (3%). Comparing the temperature correction uncertainty realizations in columns A7, A8, and A9 shows that, by far, the largest component of variance associated with failure temperature normalization is contributed by the [-15, +15] failure temperature uncertainty (column A9), due to uncertainty in the location of failure initiation.

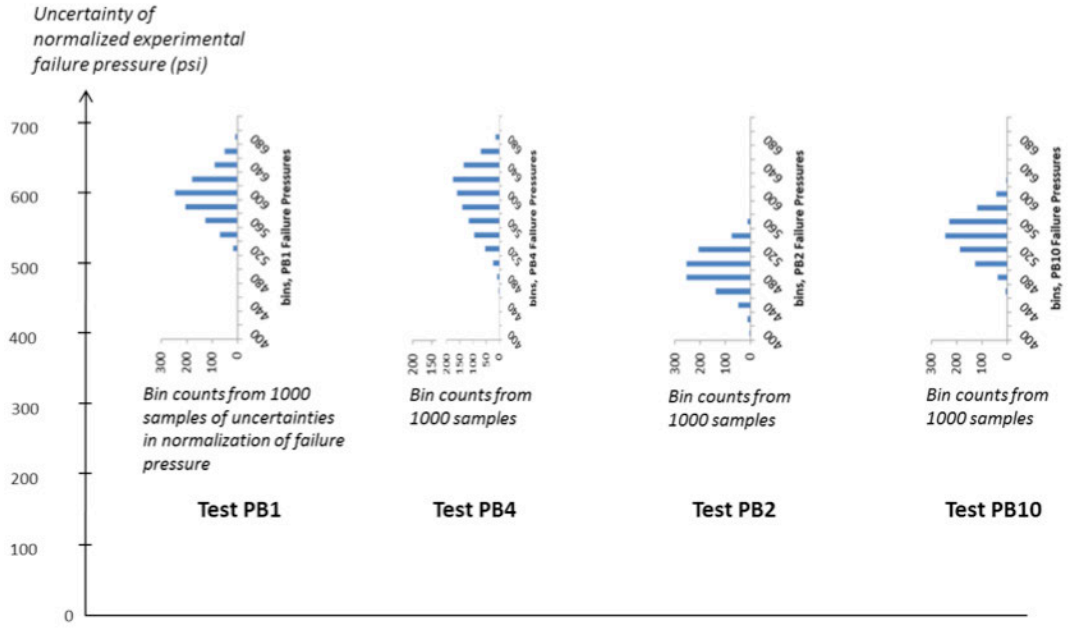


Figure 37. Uncertainty histograms of PB test failure pressures normalized to input conditions of model validation simulations.

5.3.2. Normalization for Replicates of the Reference Test

To write normalization equations for tests other than the one we are normalizing to (here, PB1), we illustrate the procedure for test PB4. We start by writing a characteristic identity for such tests. For PB4:

$$\begin{aligned} Pfail_{\mathrm{PB4}}(\vec{x}_{\mathrm{nomPB1}}) &= Pfail_{\mathrm{PB4}}(\vec{x}_{\mathrm{actPB4}}) \\ &+ [Pfail_{\mathrm{PB4}}(\vec{x}_{\mathrm{nomPB4}}) - Pfail_{\mathrm{PB4}}(\vec{x}_{\mathrm{actPB4}})] \\ &+ [Pfail_{\mathrm{PB4}}(\vec{x}_{\mathrm{nomPB1}}) - Pfail_{\mathrm{PB4}}(\vec{x}_{\mathrm{nomPB4}})]. \end{aligned} \tag{5.20}$$

The second row of this equation pertains to normalization of test results between PB4 actual and nominal experimental conditions. This is accomplished with a PB4 analogue of Eq. 5.5:

$$Pfail_{\mathrm{PB4}}(\vec{x}_{\mathrm{nomPB4}}) - Pfail_{\mathrm{PB4}}(\vec{x}_{\mathrm{actPB4}}) \approx \sum \frac{\partial(PfailPB4)}{\partial(x_i)} \bullet (x_{\mathrm{i\_nomPB4}} - x_{\mathrm{i\_actPB4}}). \tag{5.21}$$

The third row of Eq. 5.20 represents the difference that would exist if the PB4 pipe was subjected to the nominal input conditions from test PB1. We use the model to approximate this difference by simulating to failure with the nominal PB1 inputs and then with the nominal PB4 inputs:

$$Pfail_{\mathrm{PB4}}(\vec{x}_{\mathrm{nomPB1}}) - Pfail_{\mathrm{PB4}}(\vec{x}_{\mathrm{nomPB4}}) \approx Pfail_{\mathrm{model}}(\vec{x}_{\mathrm{nomPB1}}) - Pfail_{\mathrm{model}}(\vec{x}_{\mathrm{nomPB4}}). \tag{5.22}$$

The simulation model is used as the best available mechanism for estimating how behaviors would change in reality under these perturbations to the input conditions. The model does not have to be accurate in an absolute sense. But, over the uncertainty ranges of the uncertain inputs $\vec{x}_{\mathrm{i}}$ it must be sufficiently accurate in a relative sense of providing reasonable trend information (slope if linear UQ is used) such that propagating and accounting for these uncertainties improves the validation analysis vs. simply ignoring them. This is a judgment call by the physics modelers and VVUQ analysts in the project.

Substituting equations 5.21 and 5.22 into Eq. 5.20 yields:

$$\begin{aligned} Pfail_{\mathrm{PB4}}(\vec{x}_{\mathrm{nomPB1}}) &\approx [Pfail_{\mathrm{model}}(\vec{x}_{\mathrm{nomPB1}}) - Pfail_{\mathrm{model}}(\vec{x}_{\mathrm{nomPB4}})] \\ &+ Pfail_{\mathrm{PB4}}(\vec{x}_{\mathrm{actPB4}}) + \sum \frac{\partial(PfailPB4)}{\partial(x_i)} \bullet (x_{\mathrm{i\_nomPB4}} - x_{\mathrm{i\_actPB4}}). \end{aligned} \tag{5.23}$$

The term in row 1 of the RHS of this equation enacts an approximate adjustment for PB1 vs. PB4 nominal input conditions, $\vec{x}_{\mathrm{nomPB1}}$ vs. $\vec{x}_{\mathrm{nomPB4}}$. The term in row 2 of the RHS enacts an approximate adjustment for PB4 nominal vs. actual conditions, $\vec{x}_{\mathrm{nomPB4}}$ vs. $\vec{x}_{\mathrm{actPB4}}$. The treatment of row 2 is analogous to Eq. 5.5 and, therefore, ultimately Eq. 5.18. The new information required for the PB4 evaluation in Table 20 follows.

The wall thickness uncertainty in PB4 is modeled as the interval range given in Table 15, above. So:

$$\mathrm{U}[w_{\mathrm{actPB4}}] = [0.02, 0.024] \text{ inch.} \tag{5.24}$$

The failure pressure in PB4 has an uncertainty $\mathrm{U}[Pfail_{\mathrm{PB4}}(\vec{x}_{\mathrm{actPB4}})] = 655$ psi ± 10 psi (see Table 13). In tables 20 and 21, this ±10 psi uncertainty is treated as being perfectly correlated to the pressure measurement uncertainty ±10 psi in PB1 (tables 18 and 19, above) because the same

pressure gauge was used in the two tests, so *systematic* measurement error exists between the two tests. Assuming that the random-error component is small relative to the systematic component, the gage's majority of error and the associated uncertainty are systematic over the replicate tests. Thus, the realizations in column B2 of Table 21 are assigned to be the same as in column A2 of Table 19. The headings in columns A2 and B2 are highlighted in yellow to signify their commonality.

The quantities in columns B5, B8, and B10 of tables 20 and 21 are also highlighted in yellow for similar reasons. The green-highlighted entries in Table 20 denote changes from PB1's Table 18. Column A12 of Table 18 is shifted rightward to become column B13 in Table 20, with other changes highlighted in green in the column B13 heading. The new column B12 represents row 1 of the RHS of Eq. 5.23. Note that the uncertainties [-15, +15]°C in column B9 and in column A9 for PB1 are of the same magnitude but independent of each other; the realizations in a given row j of columns A9 and B9 are uncorrelated. Different initial seeds for random-number generation are used to sample all column quantities for PB1 and PB4, except for the yellow highlighted columns, as explained earlier.

Column B13 of Table 21 contains the realizations of uncertainty of PB4 failure pressure when normalized to the nominal PB1 reference experimental conditions input to the simulations in Section 5.2. These realizations are depicted by histograms in Figure 37. The realizations are processed with realizations from the other replicate tests, as explained in Section 5.3.3.

The statistics at the bottom of Table 21 summarize the impacts of various uncertainties in the normalization of PB4 failure pressure. The green boxes reveal that the average normalization adjustment in PB4 experimental failure pressure is greatest (-62.8 psi) due to possible differences from the nominal 0.02-in. wall thickness used in the simulations. This is a much larger mean adjustment than the -17.8 psi mean adjustment for PB1 normalization because PB4's [0.02-in., 0.024-in.] range of wall thickness uncertainty is centered significantly further from the nominal 0.02-in. than is PB1's [0.019-in, 0.022-in.] uncertainty range. Next in magnitude of normalization adjustment is the -8 psi in column B12, which comes from simulations at nominal PB1 and PB4 test conditions (Eq. 5.22). Next in magnitude is the average normalization adjustment of -0.96 psi due to failure temperature normalization. The green box in column B2 shows a zero average bias adjustment for failure pressure measurements error. The green boxes yield a combined bias shift of -71.8 psi from the nominal measured failure pressure of 655 psi in Table 13. The resulting average normalized failure pressure is 583.2 psi listed in the pink box at the bottom of Table 21.

Potential wall thickness differences from nominal lead to variations in normalized failure pressure characterized by a standard deviation of 39.1 psi. This is significantly larger than the value of 28.1 psi for PB1. This is because of PB4's larger [0.02-in., 0.024-in.] wall thickness uncertainty compared to PB1's [0.019 in., 0.022 in.] uncertainty. Potential failure temperature perturbations from PB4 nominal yield a standard deviation of 17.0 psi in normalized failure pressure. This very closely reflects the value of 17.1 psi for PB1 failure temperature normalization. Column B2's repeat of column A2, of course, also here yields a standard deviation of 5.8 psi due to possible pressure measurement error.

Table 20. Spreadsheet formulas for uncertain experimental quantities in Equation 5.23 for test PB4.

realization j	$\Delta Pfail_{measPB4,j}$ = A2j, systematic betw. PB1 & PB4	$w_{actPB4,j}$ = [0.02, 0.024] inches	$\Delta w_{PB4,j} = w_{nomPB4} - w_{actPB4,j}$ = 0.02" – B3	$\frac{\partial(Pfail_{PB4})}{\partial(w)}_j$ = A5j, systematic betw. PB1 & PB4	= (B4*B5) $_j$ psi	$\Delta T_{meas\text{-}TC/DAQ,j}$ = [–0.0025*711, +0.0025*711] °C	$\Delta T_{meas\text{-}contact,j}$ = A8j, systematic betw. PB1 & PB4	$\Delta T_{TC4location,j}$ = [–15, 15] °C	$\frac{\partial(Pfail_{PB4})}{\partial(T@fail\_point)}_j$ = A10j, systematic betw. PB1 & PB4	= (B7 + B8 + B9) $_j$ *B10$_j$	$[Pfail_{model}(\vec{x}_{nomPB}) - Pfail_{model}(\vec{x}_{nomPB4})]$	= 655 psi + (B2 + B6 + B11 + B12) $_j$ psi = $Pfail_{PB4}(\vec{x}_{nomPB1})_j$
B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13
1												
2												
…	…	…	…	…	…	…	…	…	…	…	…	…
J												

Table 21. Spreadsheet realizations for uncertain experimental quantities in test PB4.

j	B2=A2, sys	B3=PB4wall	B4=Δ0.02"	B5=A5, sys	B6 = B4*B5	B7=ΔTC_DA	B8=A8, sys.	B9=ΔTC_loca	B10=A10, sy	B11=B10*(B7	B12=PfPB1-	B13=655+B2+
1	7.76	0.02266	-0.00266	30701.34	-81.65	-1.39	0.52	12.84	-2.07	-24.83	-8.00	548.29
2	-5.88	0.02327	-0.00327	37837.13	-123.76	0.91	1.18	-11.81	-1.87	18.17	-8.00	535.53
3	-0.69	0.02195	-0.00195	26116.40	-51.05	0.07	1.61	-6.42	-1.83	8.66	-8.00	603.93
999	-8.55	0.02399	-0.00399	37867.63	-150.97	0.67	0.10	-14.21	-2.15	28.95	-8.00	516.43
1000	6.28	0.02353	-0.00353	29062.56	-102.57	-0.85	0.06	2.18	-1.99	-2.78	-8.00	547.94
	avg	avg	avg	avg	avg	avg	avg	avg	avg	avg	avg	avg
	0.00	0.02195	-0.00195	32361.71	-62.83	-0.05	0.91	-0.39	-1.94	-0.96	-8.00	583.21
	stdev				stdev					stdev		stdev
	5.76				39.06					17.02		42.97
												max
												672.99
												min
												445.61

The square root of the sum of squares of the contributing standard deviations in the grey boxes in Table 21 is 42.99 psi. This closely agrees with the value 42.97 psi in the pink box, calculated directly from the realizations in column B13. This standard deviation is significantly greater than that for PB1 (32.8 psi). This is reflected in the relative widths of the distributions in Figure 37.

The 82% of total variance contributed by wall thickness normalization is here proportionately even greater (compared to the PB1 case) than the variances contributed by failure temperature normalization (16%) and by pressure measurement uncertainty (2%). As in the PB1 case, the largest component of variance due to failure temperature normalization is contributed by the [-15, +15]°C failure temperature uncertainty due to uncertainty in the location of failure initiation.

5.3.3. Combining Possible Values of Normalized Failure Pressures to Yield Estimated Ranges of Failure Population Statistics

Columns A12 of Table 19 and B13 of Table 21 yield, for each row j, a pair of potential failure pressures from two replicate experiments normalized to the same experimental input conditions (those used in the predictions discussed in Section 5.2). So, hypothetically, if the realizations in a given row j happen to correspond to the exact two normalizations for the (different) true input conditions in the two tests, the disparities between the column A12 and B13 values in that row are not attributable to differences in input conditions between tests or to measurement errors on the outputs. The disparities would instead reflect differences between the two tests that were not accounted for in the normalization procedure. In the present case, the only apparent element of major importance that was not explicitly normalized is material strength differences at the failure initiation locations in the tests. Thus we would attribute any remaining differences in the exactly normalized failure pressures in this hypothetical case to material strength variability between tests.

The remaining failure pressure disparities would be a quantitative gauge of the material strength variability in the PB tests. We form associated PDFs of experimental failure pressure variability next, for validation comparisons against the predicted PDFs of failure pressure variability shown in Figure 33, which originate from material variability parameterized by the s-s curves from material characterization tests described in Section 2.

From the hypothetical row j of exactly normalized PB1 and PB4 failure pressures, we can form an n=2-sample 0.95/0.90 tolerance interval (TI) and associated normal PDF to compare against the predicted range of 0.95/0.90 TIs and associated PDFs in Figure 33. (See Table 1 and surrounding text for details on how to construct 2-sample 0.95/0.90 TIs and associated normal PDFs.)

We now go from hypothetical to non-hypothetical. As Figure 38 illustrates, 0.95/0.90 tolerance intervals are formed from PB1 and PB4 normalized failure pressures in columns A12 of Table 19 and B13 of Table 21. If the following conditions apply, then several of the 1,000 constructed TIs will usually be similar to the hypothetical exact TI constructed from perfectly normalized failure pressures simultaneously for PB1 and PB4 tests. This is according to limited numerical experiments performed by the first author.

Conditions

- Parameters explicitly normalized in the spreadsheets contain the actual experimental values within the prescribed uncertainty ranges and a sufficient number J of realizations are taken.
- The modeling of response (here, failure pressure) yields sufficiently accurate relative-trend information over the parameter ranges in the normalization procedure.

We project that the J=1000 realizations and the models used (the physics model, the experimental slope information, the estimated uncertainty ranges, and linear+ Taylor Series model) are sufficient to meet these conditions. We proceed as though this is the case.

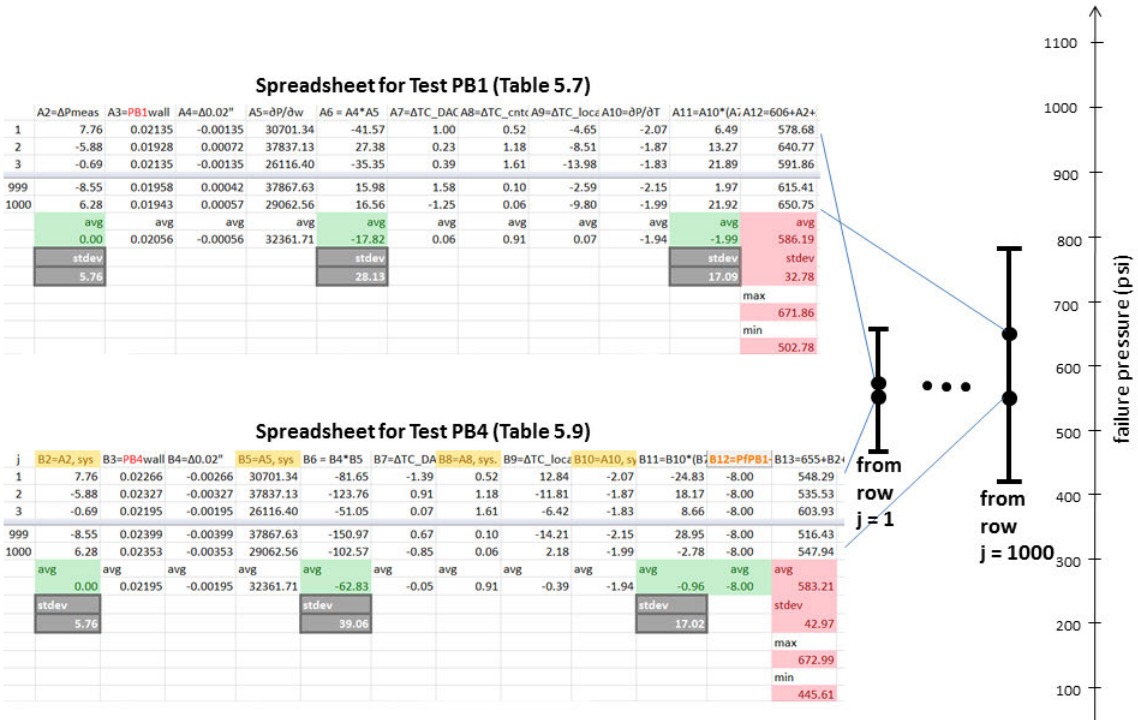


Figure 38. 0.95/0.90 tolerance intervals formed from PB1 and PB4 normalized failure pressures in the indicated spreadsheet columns.

A convenient notional representation of the significance of the 1,000 TIs is portrayed by the associated normal PDFs depicted in Figure 39. The normal PDFs in the figure only serve as a conceptualization aid; they do not need to be constructed for the present analysis. The figure identifies the uncertainty range of the lower end points of the 1,000 TIs (that is, the 1,000 lower-bound estimates for the 0.025 percentile of experimental failure pressures). An uncertainty range is also identified for the upper end points of the 1,000 TIs (that is, upper-bound estimates for the 0.975 percentile of response). We can now compare these uncertainty ranges against those for predicted 0.025 and 0.975 percentiles of response from model simulations (Figure 33) under the common reference experimental conditions. Comparisons are made and interpreted in Section 5.4.

Note that the 0.025 and 0.975 percentiles of response are the only response statistics that can be addressed from the current construction. For example, the uncertainty of the means of the depicted normal PDFs in Figure 39 (which are the same means of the underlying 0.95/0.90 TIs) are not appropriate to compare against the range of means denoted by the upper and lower simulation PDFs in Figure 33. Instead, the 1,000 rows of PB1 and PB4 normalized results would need to be processed to create 1,000 n=2-sample scaled t-distributions. Each such distribution characterizes the uncertainty of a population mean calculated from just two data samples. So, the processing of the PB1 and PB4 normalized failure pressures must be tailored to specific response statistics that are to be compared in the validation assessment.

Experimental and simulated 0.025 and 0.975 percentiles of response were chosen as validation quantities in this work because these percentiles combine the effects of small-sample uncertainties in both the response mean and variance, and are more relevant to the validation assessment of models like the current one that are being evaluated for use in design or safety margin predictions.

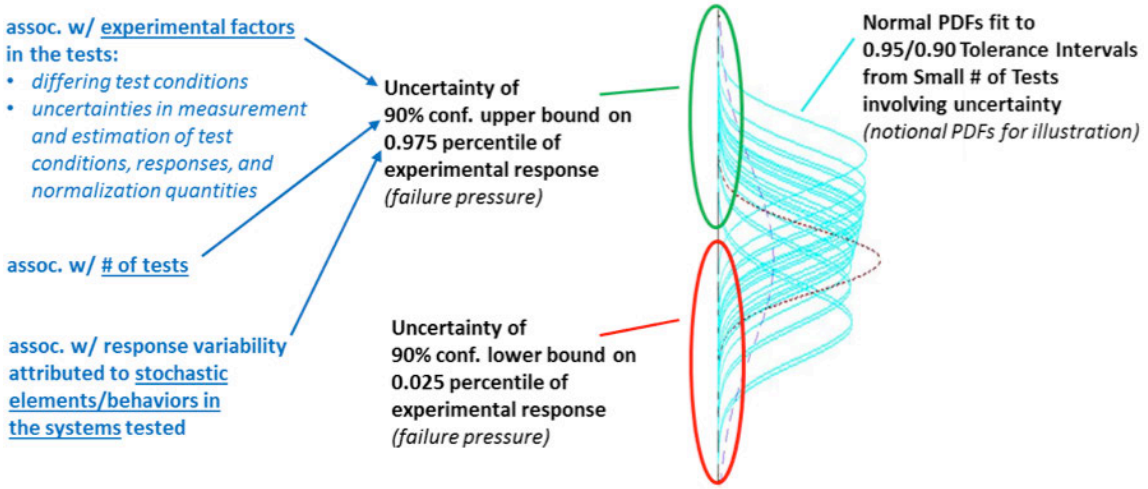


Figure 39. Uncertainty of statistical bounds on 0.025 and 0.975 failure pressure percentiles inferred from a small number of tests normalized to PB1 nominal experimental conditions.

5.3.4. Pooling More PB Tests/Results to Reduce Uncertainty of Population Statistics

The results from using the n=2 samples from the PB1 and PB4 tests yield unreasonably large uncertainty ranges for the sought percentiles. In particular, a substantial portion of the TI uncertainty range for the lower (0.025) percentile of response reached into negative (impossible) values of failure pressure. A substantial contributor to these uncertainty ranges is the large value $f_{0.95/0.90}$ = 18.8, which multiplies the response standard deviation $\tilde{\sigma}$ from two samples to obtain 0.95/0.90 tolerance intervals of half-length 18.8$\tilde{\sigma}$ (see Table 1 and surrounding text).

To obtain more statistical precision we leverage two somewhat similar tests, PB2 and PB10, from the large set of tests run [2]. PB2 and PB10 were performed at 650°C-hold conditions with pressurization profiles similar to those in the PB1 and PB4 tests. The 650°C-hold tests were planned replicates of each other. These tests are described in Appendix B and their results are normalized to the PB1 nominal conditions so they can be pooled with the normalized PB1 and PB4 results to get a population of four samples. This decreases the multiplier value to $f_{0.95/0.90}$ = 4.94 (Table 1), to about one-quarter of the two-sample magnitude of 18.8.

The 650°C-hold tests PB2 and PB10 are similar enough to the 700°C-hold tests PBs 1 and 4 that it is reasonable to expect PB2 and PB10 failure pressures can be normalized to the nominal PB1 test conditions accurately enough that pooling of the four tests' results is justified. Figure 40 shows the axial and circumferential temperature profiles at failure for PBs 1, 2, 4, and 10. The temperature profiles are approximately the same shape but are vertically shifted relative to each other in the hotspot region. The principal effects of the different hotspot temperatures in the tests are approximately normalized-out in Appendix B by assuming that the peak temperature at TC4 determines the failure pressure far more than temperatures away from this peak. Then, the temperature vs. failure pressure relationship in Figure 8 is used to normalize for different peak (TC4) temperatures. Although this approach is deemed sufficient to support the ultimate

conclusions of the validation analysis, it is less accurate than using the physics model, as was done to normalize PB2 failure pressure for its different temperature profile vs. PB1 (see Eq. 5.22 and column 12 in Table 20). It was determined late in the validation project that working with just PB1 and PB4 tests led to unacceptably large tolerance intervals for many of the realizations of normalized failure pressures, but there was insufficient time to use the simulation model for PB 2 and 10 when they were added to the analysis, given the model's month-long run times on 800 processors.

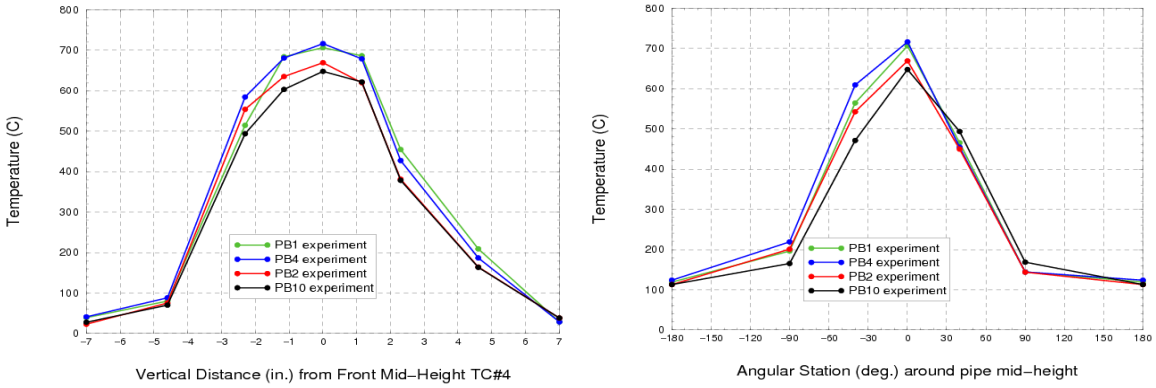


Figure 40. Pipe axial (left plot) and circumferential (right plot) TC temperatures at time of failures in the 650C-hold and 700C-hold experiments.

In Appendix B, tables B.4 and B.6 list sample realizations from the normalization of PBs 2 and 10. Columns 13 contains the realizations of the normalized failure pressures. These realizations are depicted by histograms in Figure 37. In the figure, the normalized 650C-hold PB 2 and PB10 results are, on average, lower by 50 – 100 psi than the normalized PB 1 and 4 results. So there appears to be a systematic difference between the normalized 650°C-hold PB 2 and 10 results vs. the normalized 700°C-hold PB 1 and 4 results. The PB 2 and 10 tests involve pipe temperature fields that are more than a small perturbation away from the reference PB1 conditions. Moreover, normalization for this large difference was not as accurate as it could have been if project resources would have allowed running the physics model at the PB2 and PB10 input conditions. Instead, a less accurate approach considered only the hottest TC temperature (TC4) and normalized using the relationship in Figure 8, as described in Appendix B.

The J=1000 realizations for each of the four PB tests normalized to the reference PB1 experimental conditions are depicted by histograms in Figure 37. From these, J=1000 TIs are constructed in analogy with Figure 38. Again, a convenient notional representation of the significance of the 1,000 TIs and their equivalent normals is portrayed in Figure 39. Uncertainty of the 1,000 lower ends and 1,000 upper ends of the constructed TIs are indicated. Actual values defining the uncertainty for the present problem are binned in Table 22. Corresponding histograms are shown Figure 41. Next, we compare and interpret these uncertainty ranges against those for predicted 0.025 and 0.975 percentiles of response from model simulations (Figure 33).

Table 22. Binned endpoints of 0.95/0.90 TIs constructed from realizations of normalized failure pressures of tests PB 1, 2, 4, 10.

bins, 2.5%ile (μ - 4.94σ, 4-sample 95/90 Tol. Intvl.)	**Frequency**	**Cumulative %**
0	2	0.20%
50	2	0.40%
100	15	1.90%
150	67	8.61%
200	120	20.62%
250	208	41.44%
300	218	63.26%
350	202	83.48%
400	116	95.10%
450	40	99.10%
500	9	100.00%

bins, 97.5%ile (μ + 4.94σ, 4-sample 95/90 Tol. Intvl.)	**Frequency**	**Cumulative %**
570	0	0.00%
630	11	1.10%
690	65	7.61%
750	135	21.12%
810	214	42.54%
870	265	69.07%
930	183	87.39%
990	96	97.00%
1050	26	99.60%
1110	4	100.00%

5.4. Model Validation Comparisons and Observations

In Figure 41, the histograms reflect Table 22's binned realizations of the upper and lower endpoints of 0.95/0.90 tolerance intervals from the four normalized tests. As explained earlier, the interval uncertainties input to the data normalization operations are temporarily represented as uniform PDFs and sampled accordingly. If the interval uncertainties were instead propagated as intervals, the lower and upper interval limits on the 0.975 percentile of response would extend somewhat beyond the range of the red histogram in Figure 41. However, the joint extremes of the uncertainty ranges that would produce these interval limits are considered exceedingly unlikely. Instead, the 0.05 and 0.95 percentiles of the red histogram were chosen to represent a more reasonable range of uncertainty for the 0.975 percentile of response (failure pressure). The corresponding red interval in the figure represents the "moderated" interval uncertainty used in the following comparisons. Similar considerations underlie the rightmost green interval in the figure.

The histograms in Figure 41 were constructed from interval uncertainties input to the data normalization operations, but the methodology is readily extendable to probabilistic uncertainties and combinations of interval and probabilistic uncertainties. Any of the random and systematic measurement/processing/inference uncertainties in the columns of the spreadsheets could alternatively have been PDFs or histograms or probability boxes [44]. The sampling and processing would be similar to what has been presented. Histograms like those shown at right in Figure 41 would result. It is recommended that corresponding intervals like the ones at right in the figure normally be used for comparison to uncertainties of the model-predicted percentiles, but if all data uncertainties are represented as normal PDFs in the columns, the simplified treatment discussed in Appendix C can be employed. The resulting histograms can be interpreted as PDFs of probabilistic uncertainty. Then, either the PDFs or representative spans (for example, the 0.05 to 0.95 spans) can be compared to the uncertainty ranges of the model-predicted percentiles.

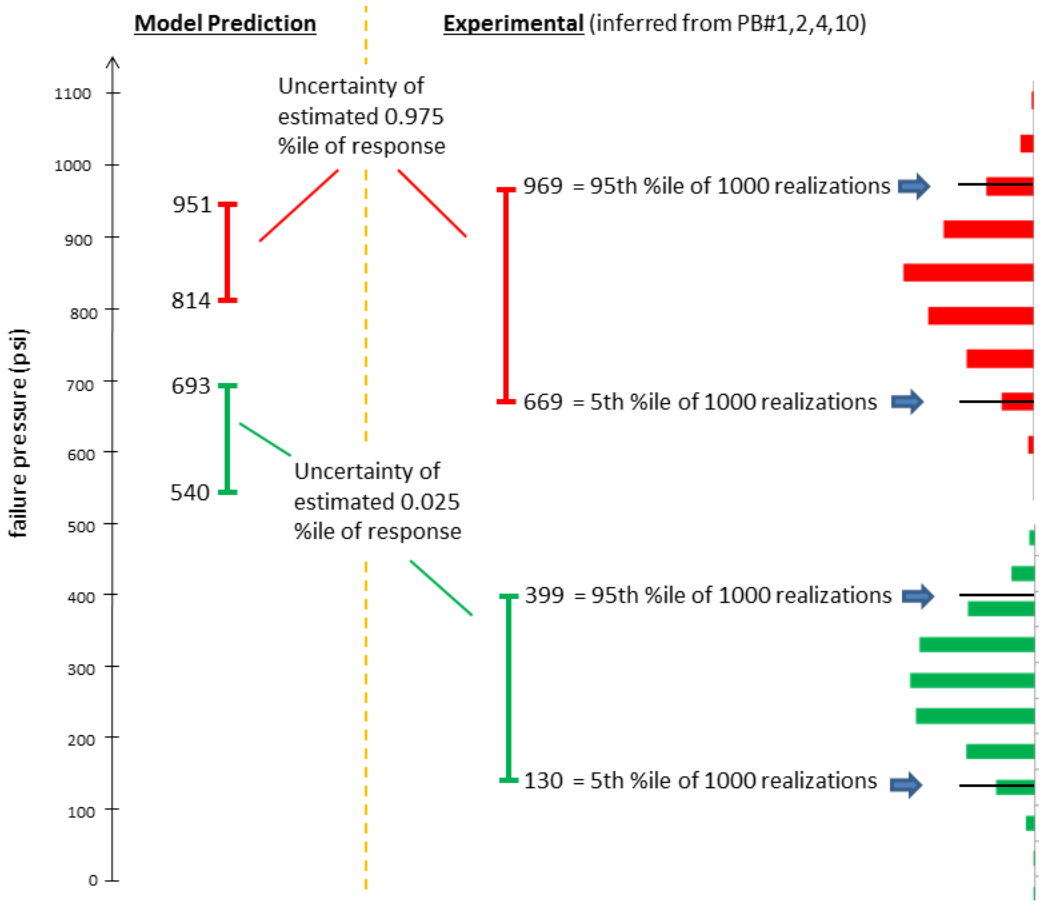


Figure 41. Model validation comparison of uncertainty ranges for simulation percentiles of response and for normalized experimental percentiles of response.

Consider the green intervals depicted in Figure 41 for 0.025 percentiles of predicted and experimental response. Notice that the green intervals do not overlap. Therefore, it is straightforward that, for this lower percentile of response, the model predicts higher failure pressures than inferred from testing. If, for instance, this lower percentile of response is written into a design or safety spec that <2.5% of pipes of this design are to fail under applied pressure and temperature conditions emulated in the tests, then the experiments are indicating a lower failure pressure for 2.5% of pipes than the model is predicting. Therefore, the model gives unconservative predictions for these circumstances.

How these results extrapolate to other applications of the constitutive model (such as different pressure vessel geometries, heating conditions, and wall thicknesses) is a very difficult issue and beyond the scope of model validation. But with some reasonable assumptions and the known variances of the experimental and predicted failure pressure populations, the results in Figure 41 can be extended to cases where the same pipe design and experimental conditions exist but the specification has lower allowable percentages of failure—like 1% or 0.001%.

Now consider the red intervals in Figure 41 for 0.975 percentiles of response. These intervals overlap and the experimental and simulation uncertainties they represent are statistically independent. Therefore, there are numerous possibilities for the experimental 0.975 percentile of response to be higher than the predicted 0.975 percentile, and vice versa. So we cannot conclude,

as was done for the 0.025 percentile, that the predictions are unconservative (or alternatively, that they are conservative). One limiting case for these ranges of uncertainty is that the predicted 0.975 percentile is as high as 951 psi as labeled in the figure, while the experimental percentile is as low as 669 psi, as labeled. In this limiting case, the predicted 0.975 percentile is up to 282 psi higher than the experimental percentile. The opposite limiting possibility is that the predicted 0.975 percentile is as low as 814 psi, while the experimental percentile is as high as 969 psi. Then the predicted 0.975 percentile is as much as 155 psi lower than the experimental percentile.

Thus, the uncertainty of the quantity {predicted 0.975 percentile minus the experimental 0.975 percentile} is [-155, 282] psi. This is the model's range of potential prediction bias for the 0.975 percentile of response (failure pressure). Information in this form accounts for all the experimental and simulation non-traveling uncertainties combined. This combined uncertainty defines the "validation resolution limit" or "validation uncertainty level" below which the model's exact bias in predicting the 0.975 percentile of response cannot be determined. The concept of a validation resolution limit or uncertainty level was first introduced by Coleman and Stern [34] and adopted by the American Society of Mechanical Engineers [22] under a reduced scope of models and uncertainties considered: only deterministic models (no traveling uncertainties intrinsically part of the model) and, therefore, only non-traveling simulation and experimental uncertainties addressed.

Because the non-traveling uncertainties of the simulations and experiments combine to yield a net validation uncertainty, a given source of non-traveling uncertainty can be aggregated either with the simulation non-traveling uncertainties, or with the experimental non-traveling uncertainties. The combined experimental and simulation validation uncertainty will be essentially the same either way—provided that a few restrictions are met, as discussed in Appendix A. Thus, it is often a matter of convenience, constrained by convention and sensibility, where individual non-traveling uncertainties are brought into the Real Space accounting ledger—either within the simulation UQ roll-up or within the experimental UQ roll-up. An example of the interchangeability is presented in Appendix A.

The kinds of information presented and discussed here are available from model validation comparisons to help contextualize model predictiveness. Conclusions regarding model adequacy for a given use purpose often come in terms of judgments about whether the model yields sufficiently conservative (but not overly conservative) predictions as determined to the extent possible in the validation assessment. This is referred to as "zeroth-order" adequacy assessment in Refs. 7 and 8. Most frequently in model validation projects, more precise statements on the adequacy of a model for anticipated uses are not rigorously supported or allowed by the information in the validation activity. See the work by Romero [7, 8] for an extended discussion of the necessary information, problem structure, and implements for proceeding beyond a zeroth-order indication of model adequacy for proposed predictions.

The quantitative validation information is usually weighed with other project information and considerations (both quantitative and qualitative) to make informed—although perhaps still highly subjective—decisions on model acceptability for identified use purposes. Thus, the result of a validation assessment is rarely a black-and-white statement about whether a model is valid for a proposed use. Rather, it provides a crucial source of information in a larger overall assessment procedure regarding model suitability for a specified use. Subject-matter expertise is an important input to these judgments.

It is also beyond the scope of model validation methodology to determine how to best use a model's validation-characterized prediction bias and uncertainty to potentially adjust or bias-

correct the model to mitigate prediction risk beyond the validation conditions. This is a very difficult question and an active area of research. See Reference 45 for more discussion about this.

6. Closing Remarks

In summary, we have presented a pragmatic and novel Real Space (RS) model validation methodology that is geared for:

- Very expensive computational models (minimal number of function evaluations).
- Quantification and economical management of mesh and solver discretization effects.
- Roll-up of various types, sources, and representations of uncertainty.
- Sparse experimental data.
- Multiple replicate experiments.
- Stochastic phenomena and models.

The validation approach and metrics:

- Segregate aleatory and epistemic uncertainties in the validation activity.
- Are relatively straightforward to interpret.
- Are especially suited for assessing models and prediction quantities to be used in the analysis of performance and safety margins.

Among the other established model validation frameworks discussed in Appendix D, the Real Space approach appears to uniquely have the required features to appropriately handle all the attributes of the challenging PB validation problem. See Refs. 7 and 8 and Chapter 10 for further comparisons of the RS approach against other established model validation frameworks.

In the PB validation problem, the largest uncertainty contributors are (in order): (1) the sparseness of repeat experiments at the pipe level; (2) solution uncertainty (discretization effects); and (3) experimental measurement variations and uncertainties in the tests. Substantial simplifications, approximations, and assumptions have been made in order to represent and process these uncertainties in a practical and affordable manner, given the high computational expense of the model and the experimental challenges and constraints.

We judge that the large magnitude of the experimental and modeling/simulation uncertainties themselves—and not the propagation errors from uncertainty linearization and decoupling in processing the uncertainties—dominate the uncertainty results. Given these results, it could not be determined whether the model is biased high or low (relative to the tests) in prediction of the 0.975 percentile of failure pressure. But at the lower end of response, a non-conservative prediction bias exists unequivocally—the tests indicate a significantly lower 0.025 percentile of failure pressure than the model predicts. So the model (without a factor of safety) may be unsuitable for design or safety analysis purposes in that actual failures at the lower end of the failure-pressure spectrum are indicated to occur at lower pressures than the model predicts.

The experimental and simulation results and uncertainties can be easily and inexpensively reprocessed for similar validation comparisons of other statistics of response such as mean, variance, and other percentiles. By analogy with the example presented, readers should be able to

treat a large variety of model validation applications, issues, constraints, and purposes that arise in industrial practice.

Acknowledgments

The authors thank retired Sandia National Laboratories employees Gerald Wellman and Martin Sherman for their contributions to the work reported here. They were integral to several aspects of the project, but are not listed as authors because they were not aware of the final outcomes and analysis methodology and results reported here, and therefore could not be represented as signatories.

The authors also thank the Advanced Simulation and Computing (ASC) program and the Weapons Systems Engineering Assessment Technology (WSEAT) experimental program of the National Nuclear Security Administration (US Dept. of Energy) and the funding managers at Sandia for supporting this work.

References

1. Dempsey, J. F., Antoun, B., Romero, V., Wellman, G., Scherzinger, W., and Grange, S., "Temperature Dependent Ductile Material Failure Constitutive Modeling With Validation Experiments," SEM XII International Congress & Exposition on Experimental and Applied Mechanics, Costa Mesa, CA, June 11-14, 2012.
2. Antoun, B., "Material Characterization and Coupled Thermal-Mechanical Experiments for Pressurized, High Temperature Systems," Sandia National Laboratories C6 L3 Milestone Report, September 11, 2009.
3. Wellman, G. W., "A Simple Approach to Modeling Ductile Failure," Sandia National Laboratories report SAND2012-1343, June 2012.
4. Scherzinger, W., Lester, B., and Newell, P., "Library of Advanced Materials for Engineering (LAMÉ 4.40)," Sandia National Laboratories report SAND2016-2774 O, March 2016.
5. Adagio 4.20 User Guide, Sandia National Laboratories report SAND2011-1825, March 2011.
6. Romero, V. J., "Type X and Y Errors and Data & Model Conditioning for Systematic Uncertainty in Model Calibration, Validation, and Extrapolation," SAE paper 2008-01-1368, Society of Automotive Engineers 2008 World Congress, Detroit, MI, April 14-17, 2008.
7. Romero, V. J., "Comparison of Several Model Validation Conceptions against a 'Real Space' End-to-End Approach," *Society of Automotive Engineers International Journal of Materials and Manufacturing*, June 2011.
8. Romero, V. J., "Elements of a Pragmatic Approach for dealing with Bias and Uncertainty in Experiments through Predictions: Data and Model Conditioning. 'Real Space' Model Validation and Conditioning; Hierarchical Modeling and Extrapolative Prediction," Sandia National Laboratories report SAND2011-7342, Nov. 2011.
9. Romero, V. J., Mullins, J., Swiler, L., and Urbina, A., "A Comparison of Methods for Representing and Aggregating Experimental Uncertainties involving Sparse Data—More Results," *Society of Automotive Engineers International Journal of Materials and Manufacturing*, Vol. 6, No. 3, 2013. doi:10.4271/2013-01-0946.

10. Romero, V., Swiler, L., Urbina, A., and Mullins, J., "A Comparison of Methods for Representing and Aggregating Uncertainties involving Sparsely Sampled Random Variables – Final Results," 15th AIAA Non-Deterministic Approaches Conference, Boston, MA, April 8-11, 2013.

11. Romero, V., Swiler, L., Urbina, A., and Mullins, J., "A Comparison of Methods for Representing Sparsely Sampled Random Quantities," Sandia National Laboratories report SAND2013-4561, September 2013.

12. Kambour, K., Hembree, C., and Keiter, E., "Calibration of Xyce Neutron Models for the QASPR Complex Prototype Circuit," Sandia National Laboratories report SAND2010-7988, December 2010.

13. Montgomery, D. C., and Runger, G. C., *Applied Statistics and Probability for Engineers*, Wiley & Sons, 1994.

14. Pradlwarter, H. J., and Schuëller, G. I., "The use of kernel densities and confidence intervals to cope with insufficient data in validation experiments," *Computer Methods in Applied Mechanics and Engineering*, Vol. 197, Issues 29-32, pp. 2550-2560, May 2008.

15. Romero, V., Rutherford, B., and Newcomer, J., "Some Statistical Procedures to Refine Estimates of Uncertainty when Sparse Data are Available for Model Validation and Calibration," paper AIAA-2011-1709, 13th AIAA Non-Deterministic Approaches Conference, Denver, CO, April 4-7, 2011.

16. Antoun, B., and Connelly, K., "Pipe Bomb Experiments for Abnormal Thermal-Mechanical Project," project review slides, Sandia National Laboratories, Feb. 6, 2013.

17. Scherzinger, W., "Modeling an Internally Pressurized Cylinder with Adagio," Sandia National Laboratories (Internal), July 18, 2013.

18. Romero, V., Dempsey, J. F., and Antoun, B., "UQ and V&V Techniques applied to Experiments and Simulations of Heated Pipes Pressurized to Failure," Sandia National Laboratories report SAND2014-3985, May 2014.

19. Eca, L., and M. Hoekstra, "Error Estimation based on Grid Refinement Studies: A Challenge for Grid Generation," Spanish Association for Numerical Methods in Engineering, 2009 Congress of Numerical Methods in Engineering, Barcelona, Spain, June 29-July 2, 2009.

20. The CUBIT Geometry and Mesh Generation Toolkit, Sandia National Laboratories. http://cubit.sandia.gov/

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

22. American Society of Mechanical Engineers, *V&V 20 – 2009 Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer*, American Society of Mechanical Engineers (ASME) Codes & Standards, 2009.

23. Trucano, T. G., Pilch, M., and Oberkampf, W. L., "General Concepts for Experimental Validation of ASCI Code Applications," Sandia National Laboratories Report SAND2002-0341, March 2002.

24. American Society of Mechanical Engineers, *V&V 10 – 2006, Guide for Verification and Validation in Computational Solid Mechanics*, American Society of Mechanical Engineers (ASME) Codes & Standards, 2006.

25. Romero, V. J., Sherman, M. P., Dempsey, J. F., Johnson, J. D., Edwards, L. R., Chen, K. C., Baron, R. V., and King, C. F., "Development and Validation of a Component Failure Model,"

paper AIAA-2005-2141, 45th AIAA/ASME/ ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Austin, TX, April 18-21, 2005.

26. Romero, V. J., "Validated Model? Not So Fast. The Need for Model 'Conditioning' as an Essential Addendum to Model Validation," paper AIAA-2007-1953, 9th Non-Deterministic Approaches Conference, Honolulu, HI, April 23-26, 2007.
27. Rutherford, R., Romero, V., Castro, J., and Hoekstra, R., "Methods Used in Uncertainty Quantification, Calibration, Validation, and Prediction for the QASPR Silicon Device Prototype Demonstration," Sandia National Laboratories report SAND2011-7940, October 2011.
28. Romero, V., Rutherford, B., and Castro, J., "Model Calibration, Extrapolative Prediction, and Validation with Segregated Aleatory and Epistemic Uncertainties—Demonstration on Prototype Silicon Device," Sandia National Laboratories report (to be published).
29. Romero, V. J., Luketa, A., and Sherman, M., "Application of a Versatile 'Real Space' Validation Methodology to a Fire Model" *AIAA Journal of Thermophysics and Heat Transfer,* Vol. 24, No. 4, pp. 730-744, Oct.-Dec. 2010.
30. Ricks, A., Nicolette, V., Romero, V., and Erickson, W., "Fuego Solid-Propellant Fire Model Verification and Validation," Sandia National Laboratories report (to be published).
31. Nakos, J. T., "Uncertainty Analysis of Thermocouple Measurements Used in Normal and Abnormal Thermal Environments Experiments at the Radiant Heat Facility and the Lurance Canyon Burn Site," Sandia National Laboratories report SAND 2004-1023, April 2004.
32. Nakos, J. T., Suo-Anttila, J. M., and Gill, W., "Shroud Boundary Condition Characterization Experiments at the Radiant Heat Facility," Sandia National Laboratories report SAND2004-5080, Oct. 2004.
33. Romero, V. J., "Data & Model Conditioning for Multivariate Systematic Uncertainty in Model Calibration, Validation, and Extrapolation," paper AIAA-2010-2511, 12th AIAA Non-Deterministic Approaches Conference, Orlando, FL, April 12-15, 2010.
34. Coleman, H. W., and Stern, F., "Uncertainties in CFD Code Validation," *Journal of Fluids Engineering*, Vol. 119, pp. 795-803, Dec. 1997.
35. Chen, W., Xiong, Y., Tsui, K.-L., and Wang, S., "A Design-Driven Validation Approach using Bayesian Prediction Models," *American Society of Mechanical Engineers Journal of Mechanical Design*, Vol. 130, No. 2, 2008.
36. Roy, C. J., and Oberkampf, W. L., "A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing," *Computer Methods in Applied Mechanics and Engineering*, Vol. 200, 2011.
37. Sankararaman, S., and Mahadevan, S., "Comprehensive Framework for Integration of Calibration, Verification, and Validation," 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Honolulu, HA, April 23-26, 2012.
38. American Society of Mechanical Engineers, *Test Uncertainty,* PTC 19.1-2005, American Society of Mechanical Engineers (ASME) Codes & Standards, 2006.
39. Oberkampf, W. L., and Barone, M. F., "Measures of Agreement between Computation and Experiment: Validation Metrics," *Journal of Computational Physics,* Vol. 217, pp. 5-36, 2006.

40. Ferson, S., Oberkampf, W. L., and Ginzburg, L., "Model Validation and Predictive Capability for the Thermal Challenge Problem," *Computational Methods in Applied Mechanics and Engineering*, Vol. 197, pp. 2408 –2430, 2009.

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

42. American Society of Mechanical Engineers, *An Illustration of the Concepts of Verification and Validation in Computational Solid Mechanics*, V&V 10.1-2012, American Society of Mechanical Engineers (ASME) Codes & Standards, April 16, 2012.

43. Romero, V., Black, A., Breivik, N., Orient, G., Suo-Anttila, J., Antoun, B., and Dodd, A., "Advanced UQ and V&V Procedures applied to Thermal-Mechanical Response and Weld Failure in Heated Pressurizing Canisters," Sandia National Laboratories report SAND2015-3005C, presented at the Society of Automotive Engineers 2015 World Congress, Detroit, MI, April 21-23, 2015.

44. Romero, V., "Approximate Probability Boxes and Other Shortcuts in a Broad-before-Deep approach to Balanced UQ," Sandia National Laboratories report SAND2015-3605C, presented at the American Society of Mechanical Engineers 2015 Verification and Validation Symposium, Las Vegas, NV, May 12-15, 2015.

45. Romero, V., "Uncertainty Quantification and Sensitvity Analysis—Some Fundemental Concepts, Terminology, Definitions, and Relationships," *Simulation Credibility—Advances in Verification, Validation, and Uncertainty Quantification*, U. Mehta, ed., D. Eklund, V. Romero, J. Pearce, and N. Keim, NASA/TP-2016-219422 and JANNAF/GL-2016-0001, 2016.

Appendix A. Example of Bias Correction for Spatially Correlated Temperature Measurement Errors

Thermocouple (TC) temperature measurements for steel surfaces at temperatures very different from the effective temperature of the surroundings can have significant errors caused by TC contact resistance and fin effects [32]. This "TC attachment" measurement bias can, for properly calibrated TCs, swamp errors associated with the data acquisition system, the calibration procedure, and TC-to-TC accuracy variations due to manufacturing variability [31]. The primary factors governing attachment bias error are: the TC wire/bead diameter; method of TC bonding or attachment to the surface; temperature differences between the measured surface and the effective radiative and convective temperatures of the surroundings; and the convective and radiative properties and conditions affecting heat transfer between the TC bead and its surroundings.

The experimental and physics modeling investigations and analysis reported by Nakos, Suo-Anttila, and Gill [32] provide a sufficient basis to estimate the attachment bias errors in the "pipe bomb" (PB) project. Early in the PB validation analysis, a miscommunication occurred regarding the diameter of the TCs used in the project. It was initially understood that the TCs had a wire/bead diameter of 0.05 inches. This lies between the 0.04 in. and 0.063 in. diameter data sets from Reference 32. Adjustments to the 0.04-in. and 0.063-in. diameter data from Ref. 32 were made for differences in geometry and the surrounding conditions in the PB experiments vs. those in the experiments in Ref. 32. The adjusted data were then interpolated to get TC measurement biases and corresponding corrections for 0.05-in. diameter TCs as exemplified in Figure A.1. With the environment and geometry parameters fixed for a given PB test, TC measurement bias and associated corrections vary in sign and magnitude according to the TC's temperature and location (which dictates its radiative and convective environment). Therefore, the percentage bias and correction varies in time and space. Figure A.1 shows that the TCs facing the heating plate register hotter temperatures than the surface they are attached to, and therefore require downward (negative) correction. The TCs on the unheated back side of the pipe are cooled by the environment they are exposed to, so they register cooler temperatures than the surface they are attached to, and therefore require upward correction.

A simulation was run with nominal corrections to the TC temperatures. The same solver settings and 1tt mesh full-model were used, as for the calculations in sections 4.2 and 4.3 in the chapter.

The calculated failure pressure was 24 psi greater with TC correction. Lowering the hotspot region temperatures as indicated in Figure A.1 has the effect of strengthening the pipe wall material there. So, a higher pressure is required to initiate failure. Figure A.2 shows a corresponding +24 psi offset arising from TC measurement bias correction. Uncertainty on the nominal corrections was not determined because this would require more model simulations, which could not be afforded.

We estimate that the exact TC bias corrections for each TC are within +/- 50% of the nominal corrections used in the simulation. The methodology for estimating the nominal corrections arose from the first author's analysis and projection of the information from Nakos et al. [32], so a high degree of systematic error may exist in the calculated biases and corrections. Therefore, the correction uncertainties would be treated as highly correlated over the pipe surface. In any case, uncertainties could not be explored because of simulation budget constraints.

Later in the project, we determined that intrinsic TCs with an order of magnitude smaller diameter, 0.005 in. had actually been used. This made the TC-attachment component of measurement error much less important, so a truncated approach that corrects only TC4's temperature (at the hotspot) was used for this component of error in sections 5.3.1 and 5.3.2 and Appendix B.

Note that the approach described in this appendix put the correction (and any uncertainty that would have been affordable to quantify) into the Simulation UQ Roll Up. But the truncated approach in sections 5.3.1 and 5.3.2 and Appendix B adds the correction (and associated uncertainty) to the processing and roll-up of experimental uncertainties. The correction is in the positive (+) direction in the simulation UQ roll-up, whereas it is in the negative (-) direction in the experimental UQ roll-up. This is consistent, and an important flexibility property of the Real Space validation method when traveling epistemic uncertainties do not exist in the validation problem—as is the case here (see the paragraph immediately preceding Section 5.1).

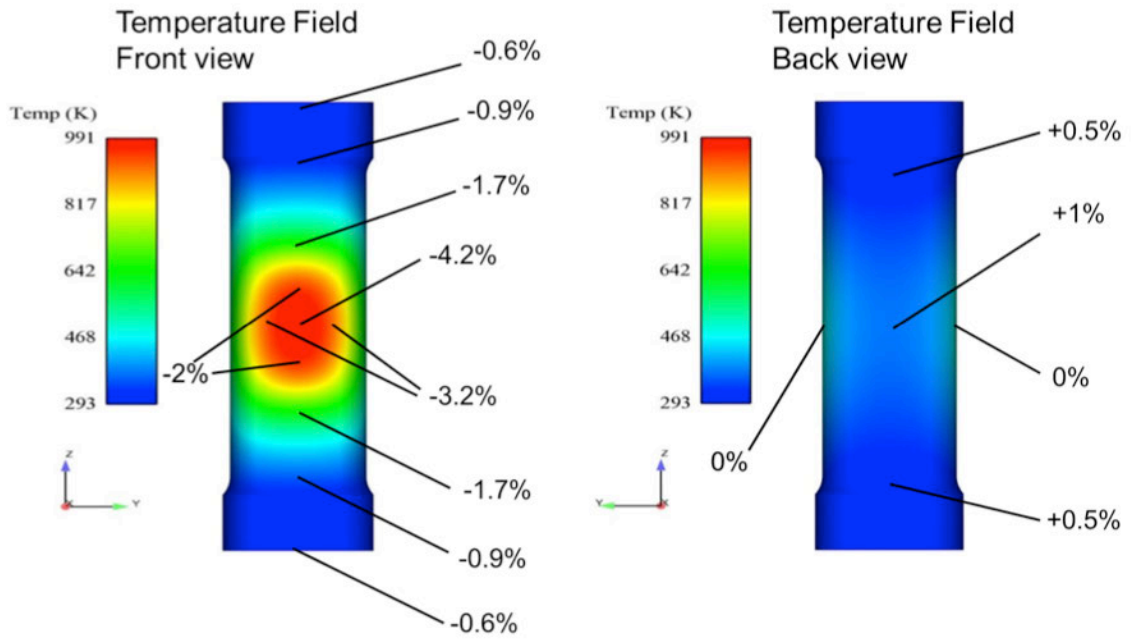


Figure A.1. Percent bias corrections to TC temperatures input to simulation of test PB1, with hypothetical large TC diameters of 0.05 inches. Corrections are for the shown temperature field at the time of failure. Corrections are for temperature measurement errors caused by contact resistance between the TC and the surface it is attached to, and by convective and radiative heat transfer between TC wire/bead and the surroundings.

For example, the indicated correction of +24 psi for 0.05-in diameter TCs can be added to the simulation UQ roll-up, as shown in Figure A.2. This adjusts the simulation result up relative to the experimental failure pressures. The experimental data is not normalized with respect to TC measurement bias because the input TC temperatures in the simulation have been corrected (as close as we can reasonably attain) to be the actual temperatures that occurred in the experiments. Thus, the simulations and experiments are put on the same basis with respect to hotspot temperature.

Alternatively, as in sections 5.3.1, 5.3.2 and Appendix B, the bias correction can be applied to the experimental failure pressures—adjusting them downward by 24 psi. This puts the adjusted experimental result on the same basis of biased temperatures that the simulations were run with (the rolled-up simulation results in Figure 33, above, have no correction for TC measurement bias). That is, the experimental failure pressures correspond to the actual temperatures at the TC locations, not to the biased measured values input to the validation simulations. To normalize the experimental results to the temperature conditions input to the simulations (to put experimental and simulation results on the same basis for comparison), we use the model to estimate how the experimental failure pressures would change if the pipe temperatures were perturbed to the biased temperatures in the simulations. We have already established that the model run with the biased temperatures gives a 24 psi lower

failure pressure than the model run with the corrected temperatures. Thus, we estimate that experimental failure pressures would change commensurately, decreasing by 24 psi and thereby moving downward relative to the simulated failure pressure.

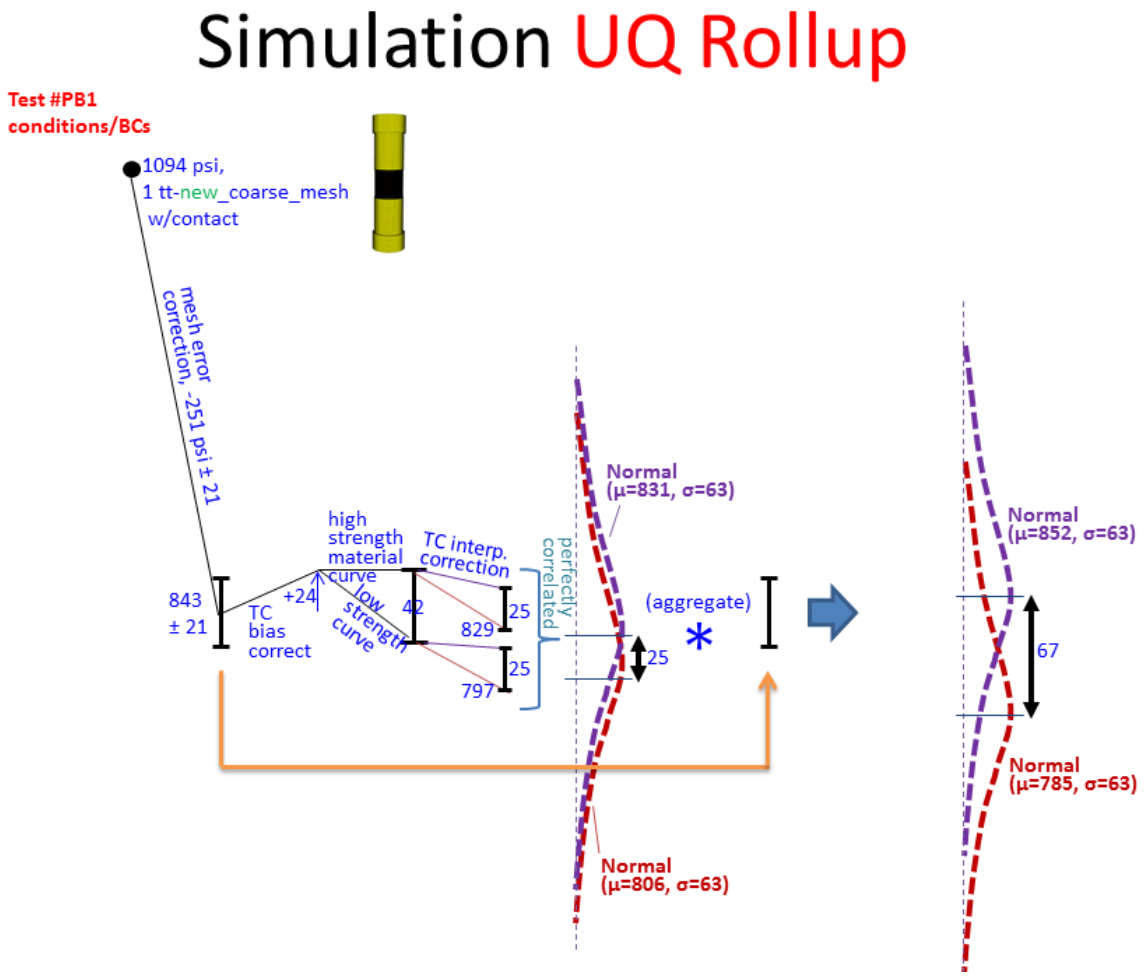


Figure A.2. Early version of Figure 33, above, with +24 psi failure pressure correction labeled "TC bias correct" arising from nominal corrections to TC temperatures input to simulation of test PB1 (for hypothetical large TC diameters of 0.05 inches).

So, either the simulation results are corrected up by 24 psi relative to the experimental results, or the experimental results are corrected down by 24 psi relative to the simulation results. Either case gives essentially the same combined uncertainty range[5] for the model's prediction bias. Thus it is often a matter of convenience, constrained by convention and sensibility, where individual (negligibly correlated or interacting) non-traveling uncertainties are bought into the RS accounting ledger—either within the simulation UQ roll-up or within the experimental UQ roll-up.

[5] This assumes that the same UQ propagation approaches are used for simulation uncertainty roll up and for normalization of the experimental data. Even if the same UQ propagation approaches are used, small differences in the combined uncertainty may exist if interval descriptions exist for some or all non-traveling uncertainties in the simulations and experiments.

Appendix B. Description of PB2 and PB10 Tests and Normalization of their Failure Pressures to PB1 Reference Conditions

This appendix pertains to Section 5.3.4 in the chapter. Figures B.1 and B.2 plot the thermocouple temperature histories in tests PB2 and PB10. The peak hotspot temperatures at the pipe front-center location, as indicated by TCs #4 in each plot, were ramped at a rate of approximately 31°C/min., just as for PB1 and PB4.

Table B.1 lists the linear regression values (over the last 60 seconds before failure) of the TC#4 temperatures. Tests PB 2 and 10 also used intrinsic TCs of 0.005-inch diameter, so very small measurement uncertainties exist from random and systematic sources of error described in connection with Eq. 5.10, above). The largest uncertainties are indicated in Table B.1, corresponding to the hottest TCs on the pipes. These uncertainties are less than the line thickness in the plots in figures B.1 and B.2.

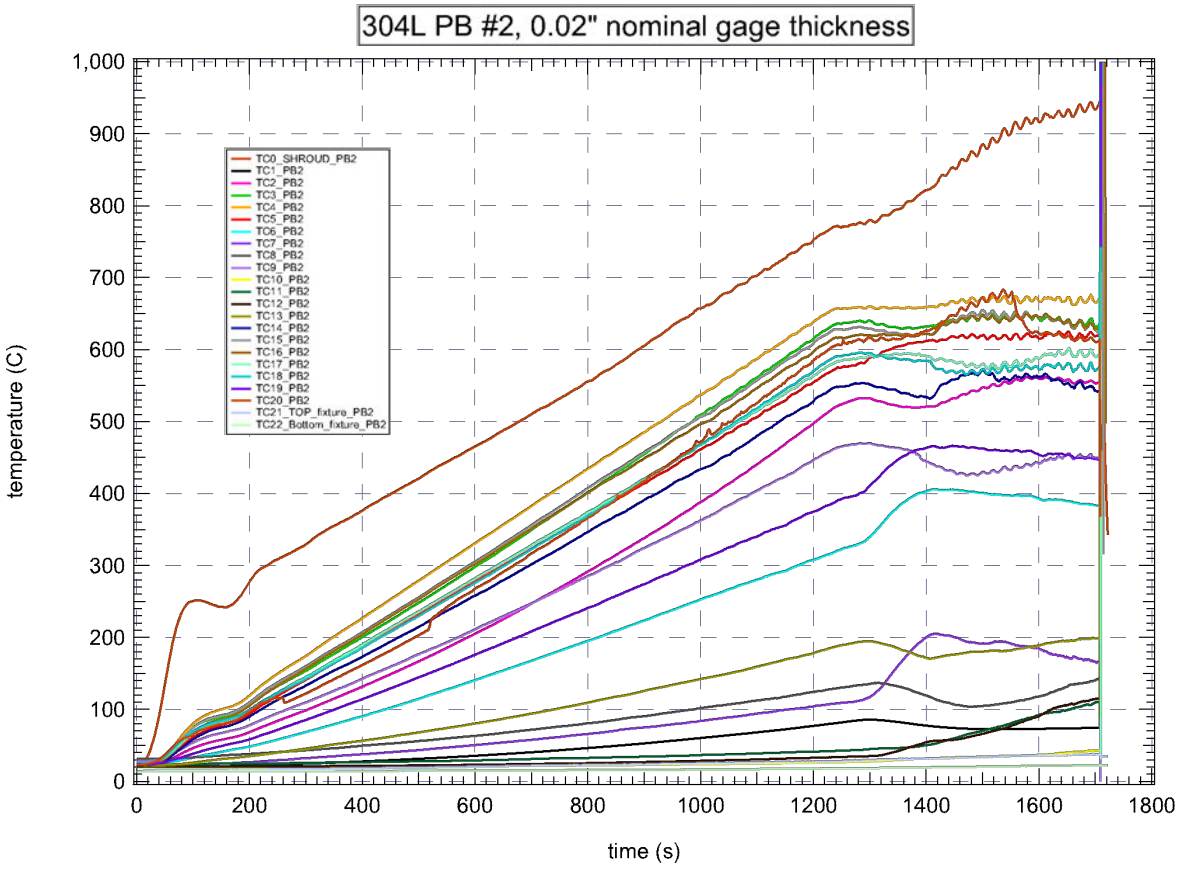


Figure B.1. Thermocouple temperature measurements in experiment PB2. TC numbering in this figure corresponds to numbered locations in Figure 21, above.

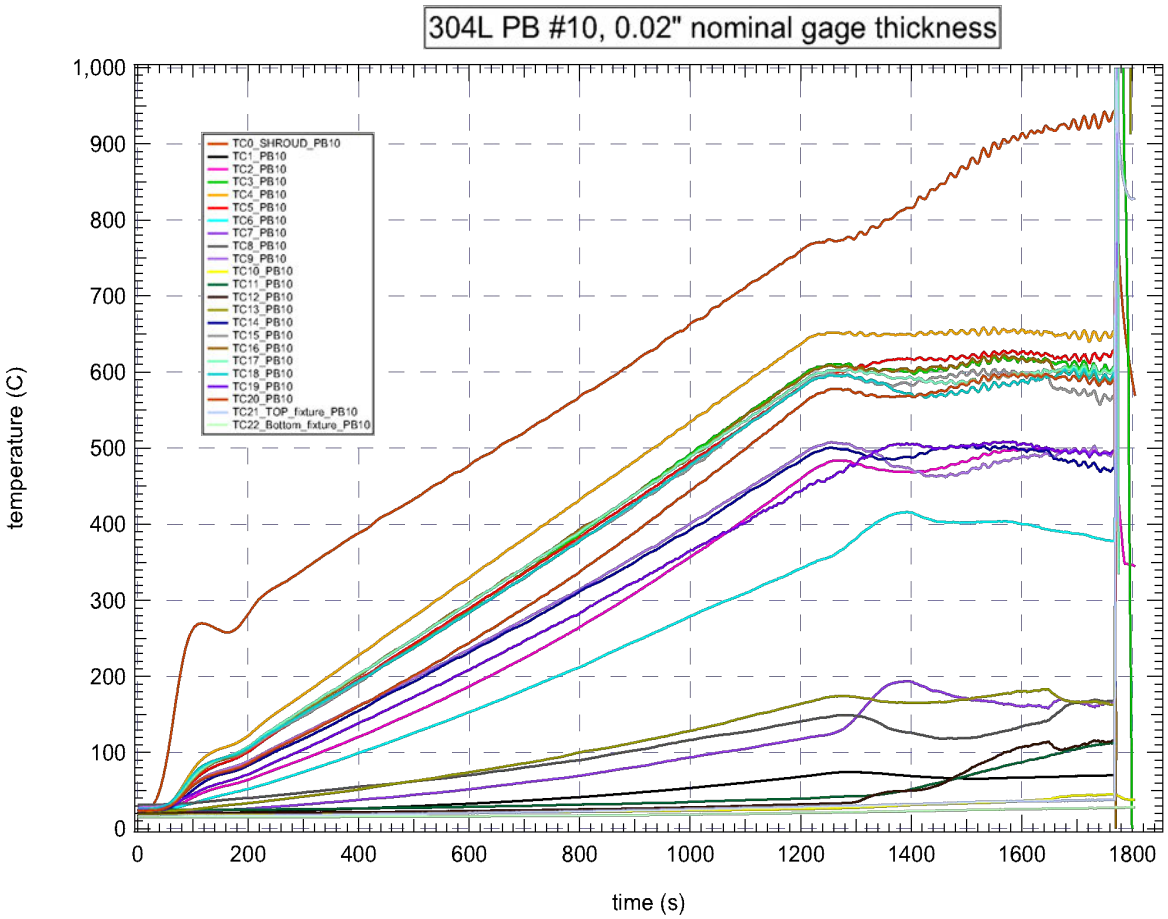


Figure B.2. Thermocouple temperature measurements in experiment PB10. TC numbering in this figure corresponds to numbered locations in Figure 21, above.

Table B.1. Pipe front-center thermocouple temperature at failure in 650C-hold experiments.

Experiment	Measured Temperature at Failure (After Regression)	% Uncertainty in Temperature Measurement	Uncertainty in Temperature Measurement
PB 2	672°C	[-0.25%, +0.5%] of reading in degrees C	[-1.7, +3.4]°C
PB 10	648°C	[-0.25%, +0.5%] of reading in degrees C	[-1.6, +3.2]°C

When the control TC4 in each test reached nominally 650°C, this temperature was maintained while the pipes were pressurized until failure as shown in Figure B.3. Pressures were measured with the same pressure gage as in tests PB1 and 4. Therefore the uncertainty is within +/- 10 psi of the measured pressures in the figure. This amounts to about double the line thickness in the plots. Table B.2 lists the failure pressures and measurement uncertainties for the PB2 and PB10 tests. These measurement uncertainties are perfectly correlated with those in the PB1 and PB4 tests because the same pressure gage was used in all four tests. Pressure measurement uncertainties are explained in Section 5.1.1.

Table B.2. Failure pressures measured in 650C-hold experiments.

Experiment	Measured pressure at failure	Uncertainty in pressure measurement
PB2	587 psi	± 10 psi
PB10	647 psi	± 10 psi

The target pressure ramp rate of 1.3 psi/sec. was met within a few percentage points in both tests, the same as in tests PB1 and PB4. PB2 pressurization started about 40 seconds after 650°C was reached. PB10 pressurization started about 30 seconds after 650°C was reached. In tests PB1 and PB4, pressurization started about 4 minutes and 1.5 minutes, respectively, after the target temperature of 700°C was reached. These test-to-test differences in pressurization delay after target temperatures were reached are not expected to significantly affect failure pressure levels. Certainly, they do not in the model simulations because the modeled physics are agnostic with respect to any such time delay effects.

Figure B.3 also shows the pipe axial loading profiles in the two experiments. The uncertainty on the axial loading measurements is ±0.1% of reading, or < ±3 lb. for the highest axial load in the four PB tests. This is far less than the line thickness in the plots. Investigations with the PB model indicated that this magnitude of axial loading uncertainty has negligible effect on failure pressures.

Table 15 lists minimum and maximum measured pipe-wall thicknesses for tests PB2 and PB10. These min-to-max ranges of thickness are less than the ranges for PBs 1 and 4. Therefore, significantly smaller normalization variance exists for wall-thickness normalizations of PBs 2 and 10 than for PBs 1 and 4.

PB2 Normalization

Tables B.3 and B.4 list test PB2 normalization quantities and sample realizations. Again, the quantities in yellow highlighted columns C2, C5, C8, and C10 designate correlated samples with the analogous columns for tests PB1 and PB4. Different random-number generation seeds are used to sample all column quantities for PBs 1, 2, 4, and 10 except for the yellow-highlighted columns. The green-highlighted entries in Table B.3 denote changes from Table 20. Of particular note are the entries in column C12 and the multipliers 1.07 and 0.95 in columns C5 and C8. These are explained next.

The quantity in column C12 represents the third row of Eq. 5.20, which here becomes the failure pressure adjustment for differences between the nominal input conditions for tests PB2 and PB1. It was determined late in the project to include the PB2 and 10 tests in the validation assessment, so there was insufficient time to use the simulation model to evaluate the PB2 analogue of Eq. 5.22. Therefore, we used the strategy described next.

We observed that the small differences in pressurization delay times and ramp rates in the tests will not yield differences in predicted failure pressures because the modeled physics are agnostic to such differences. But temperature field differences will affect predicted failure pressure. We note from Figure 40, above, that the spatial temperature field is approximately the same shape for PBs 1, 2, 4, and 10, but the fields are vertically shifted relative to each other in the hotspot region. This region, and in particular the mapped peak temperature at TC4 in the simulations, determines the calculated failure pressure far more than any of the other TCs. We employ an approximation that the predicted failure pressure depends fully on the differences of the TC4 peak temperatures at failure. The applicable temperatures are then 707°C @failure for TC4 in PB1 (see Table 14) and 672°C @failure for TC4 in PB2 (see Table B.1). Therefore, the entry in column C12 of Table B.3 serves as an approximate replacement for the term in row 1 of the RHS of Eq. 5.23. The said temperature difference is proposed to affect the predicted failure

pressure in the way that the other approximate temperature-related adjustments in columns C7-C9 do, via the temperature effect factor in column C10, obtained from the trends in Figure 8, above.

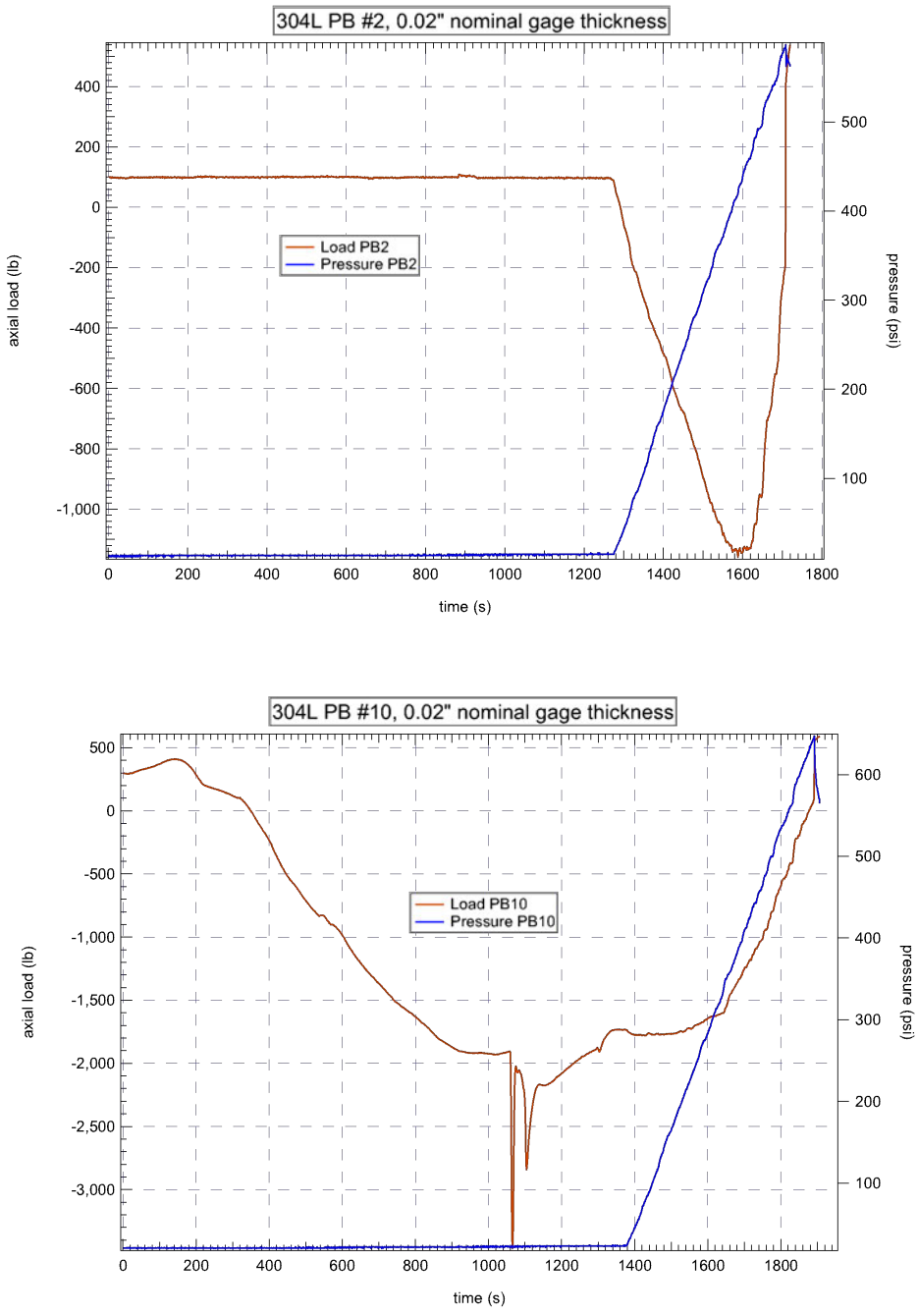


Figure B.3. Measured pressure (referenced to scale on right axis), and axial fixture loading (referenced to scale on left axis) in 650C-hold pressurization experiments. Note that plotted PB10 results here were incorrectly shifted to zero reference time. These results must be shifted earlier by 123 seconds so that the failure time in this plot coincides with that in Figure B.2 at 1769 seconds.

Table B.3. Spreadsheet formulas for uncertain experimental quantities in modified version of Equation 5.23 for test PB2**.**

realization j	$\Delta Pfail_{measPB2,j}$ = A2j, systematic betw. PB1,4,2,10	$w_{actPB2,j}$ = [0.02, 0.022] inches	$\Delta w_{PB2,j} = w_{nomPB4} - w_{actPB4,j}$ = 0.02" – C3	$\frac{\partial(Pfail_{PB2})}{\partial(w)}_j$ = 1.08*A5j, correlated betw. PB1,4,2,10	= (C4*C5) j psi	$\Delta T_{meas\text{-}TC/DAQ,j}$ = [–0.0025*672, +0.0025*672] °C	$\Delta T_{meas\text{-}contact,j}$ = 0.95*A8j, correlated betw. PB1,4,2,10	$\Delta T_{TC4location,j}$ = [–15, 15] C	$\frac{\partial(Pfail_{PB2})}{\partial(T@fil\_point)}^j$ = A10j, systematic betw. PB1,4,2,10	= $(C7 + C8 + C9)_j$ *C10j	= C10*(707C – 672C)	= 587 psi + (C2 + C6 + C11 + C12)j psi = $Pfail_{PB2}(\vec{x}_{nomPB,j})$
C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13
1												
2												
…	…	…	…	…	…	…	…	…	…	…	…	…
J												

Table B.4. Spreadsheet realizations for uncertain experimental quantities in test PB2.

	C2=A2, ΔPm	C3=PB2wall	C4=Δ0.02"	C5=1.08*A5	C6 = C4*C5	C7=ΔTC_DA	C8=0.95*A8	C9=ΔTC_loc	C10=A10, ∂I	C11=C10*(C	C12=C10*(7	C13=587+C
1	7.76	0.02026	-0.00026	32850.44	-8.39	0.71	0.49	-5.44	-2.07	8.79	-72.56	522.61
2	-5.88	0.02163	-0.00163	40485.73	-66.00	-1.57	1.12	13.93	-1.87	-25.18	-65.38	424.57
3	-0.69	0.02003	-0.00003	27944.54	-0.87	1.05	1.53	-14.44	-1.83	21.68	-63.97	543.15
1000	6.28	0.02118	-0.00118	31096.94	-36.67	0.74	0.06	9.48	-1.99	-20.50	-69.80	466.31
	avg	avg	avg	avg	avg	avg	avg	avg	avg	avg	avg	avg
	0.00	0.02099	-0.00099	34627.03	-34.15	-0.01	0.86	-0.10	-1.94	-1.46	-67.76	483.63
	stdev				stdev					stdev	stdev	stdev
	5.76				21.13					17.03	4.47	27.20
												max
												552.20
												min
												392.88

The multiplier 0.95 in column C8 adjusts the uncertainty of TC4 temperature measurement error $\Delta T_{\text{meas-contact}}$ caused by convective and radiative heat losses from the TC and by contact resistance between the pipe surface and the TC. An adjustment to the lower and upper uncertainty bounds in Eq. 5.12 is required because the PB2 TC4 temperature at failure (672°C) is significantly different than the PB1 and PB4 temperatures of 707°C and 711°C, for which the values in Equation 5.12 apply. From [32] the uncertainty bounds in Eq. 5.12 are reduced by 5% for PB2's TC #4 at 672°C. The implementation of this adjustment, shown in column C8, preserves the correlation with realizations in columns A8, B8, and D8 for PBs 1, 4, and 10. This enforced correlation reflects the view that the $\Delta T_{\text{meas-contact}}$ error is uncertain but consistent among TC4s in the PB tests.

The multiplier 1.07 in column C5 adjusts the upper and lower bounding slopes $\frac{\partial(Pfail_{\text{PB1}})}{\partial(w)}$ in Eq. 5.8 to PB2 temperature conditions. First, it is noted that the calculated failure pressure at the nominal 0.02-in. wall thickness in Figure 36 corresponds to a PB1 hotspot TC4 temperature of 707°C. This calculated failure pressure is then adjusted to a PB2 TC4 temperature of 672°C, as follows.

A temperature effect factor of $\frac{\partial(Pfail)}{\partial(T@fail\_point)}$ = -2.16 psi/°C is calculated as the average of the upper and lower bounds in Eq. 5.17. These bounds are derived from a consideration of Figure 8 over the temperature range 600°C – 800°C, applicable here. We have:

$$\begin{aligned} Pfail_{\text{PB2-modeled}}(\vec{x}_{\text{nomPB2}}) \approx\ & Pfail_{\text{PB1-modeled}}(\vec{x}_{\text{nomPB1}}) + \\ & + \frac{\partial(Pfail_{\text{PB1}})}{\partial(T@fail\_point)} (\text{Temp\_TC4-PB2 - Temp\_TC4-PB1}) \\ & = 839 \text{ psi} + (-2.16 \text{ psi/°C})(672°C - 707°C) \\ & = 914 \text{ psi} \end{aligned} \tag{B.1}$$

In Figure 36, above, we plot the temperature-adjusted failure pressure 914 psi at a wall thickness abscissa of 0.02-in. Through this point we create a line that also intersects the origin of the plot (from physical reasoning that zero wall thickness must correspond to failure pressure = 0). This line has a slope of 45,700 psi/in. Thus, the original calculated slope of 42,286 psi/inch for PB1 nominal conditions must be multiplied by an adjustment factor of 1.08 to get the PB2 temperature-adjusted slope of 45,700 psi/in. We use this same multiplier (1.08) to similarly adjust the lower-bound slope in Eq. 5.8. Thus:

$$\frac{\partial(Pfail_{\text{PB2}})}{\partial(T@fail\_point)} = 1.08 \cdot \frac{\partial(Pfail_{\text{PB1}})}{\partial(T@fail\_point)} \tag{B.2}$$

The implementation of this equation in column C5 preserves the correlation with realizations in columns A5, B5, and D5 for PBs 1, 4, and 10. This enforced correlation represents the view that, everything else being equal, failure pressure (adjusted for temperature differences as above) scales with hotspot wall thickness similarly in the tests, but epistemic uncertainty exists regarding the scaling value as reflected in the different experimental and simulation-based estimates in Figure 36.

Column C13 of Table B.4 contains the realizations of uncertainty of PB2 failure pressure when normalized to the nominal PB1 reference experimental conditions. These realizations are presented as histograms in Figure 37.

The statistics at the bottom of Table B.4 summarize the impacts of various uncertainties in the normalization of PB2 failure pressure. The green boxes reveal that the average adjustment in PB2 experimental failure pressure is greatest (-68 psi) due to normalization for the TC4 peak temperature difference between nominal PB1 and nominal PB2 conditions.

The next greatest in magnitude is -34.2 psi for average normalization adjustment for possible differences from the nominal 0.02-in. wall thickness. This is a much larger mean wall adjustment than the -17.8 psi mean wall adjustment for PB1 normalization, but much smaller than the -62.8 psi mean wall adjustment for PB4 normalization.

Next in magnitude is -1.5 psi average adjustment for the other temperature normalizations for columns C7–C9. The green box in column C2 shows a zero average bias adjustment for failure pressure measurement error. The green boxes show a combined bias shift of -103.4 psi from the nominal measured failure pressure of 587 psi I,n Table B.2. The resulting average normalized failure pressure is 483.6 psi, as listed in the pink box at the bottom of Table B.4.

The grey boxes in Table B.4 show the relative contributions to normalization uncertainty. Potential wall thickness differences are the largest uncertainty contributor (standard deviation of 21.1 psi), but this is significantly smaller than the values 28.1 psi for PB1 and 39.1 for PB4.

The square root of the sum of squares of the contributing standard deviations in the grey boxes is 28.1 psi. This closely agrees with the value 27.2 psi in the pink box, calculated directly from the realizations in column C13. The normalization uncertainty for PB2 is smaller than for PBs 1, 4, and 10 as reflected by the relative widths of the distributions in Figure 37, above.

PB10 Normalization

Tables B.5 and B.6 list test PB10 normalization quantities and sample realizations. The comments above concerning the yellow highlighting of columns apply here as well. The green-highlighted entries in Table B.5 denote changes from PB2's Table B.3. Changes that may need explanation follow.

- **Column D8**: The uncertainty bounds of Eq. 5.12 are reduced by 8% for PB10's TC4 failure temperature of 648°C.
- **Column D5**: Use Eq. B.1 and change its PB2 failure temperature of 672°C to a value of 648°C for PB10. A result of 966 psi is obtained. Then, perform the steps in the paragraph following Eq. B.1, where 914 psi in that paragraph is replaced by 966 psi and PB2 is replaced by PB10. The result is the multiplier 1.14 in column D5.

Column D13 of Table B.6 contains the realizations of uncertainty of PB10 failure pressure when normalized to the nominal PB1 reference conditions. These realizations are presented as histograms in Figure 37.

The green boxes at the bottom of Table B.6 reveal that the average adjustment in PB10 experimental failure pressure is greatest (-114.2 psi) due to normalization for the TC4 peak temperature difference between nominal PB1 and nominal PB10 conditions. Next in magnitude is the average normalization adjustment of -2.7 psi due to the other temperature normalizations for columns D7–D9. Next in magnitude is the 0.4 psi mean adjustment due to possible differences from the nominal 0.02-in. wall thickness used in the simulations. This is so small—much smaller than for any of the other tests (PBs 1, 2, and 4) because of the perfect symmetry (about the nominal 0.02-in.) of the uncertainty range listed in column D3 of Table B.5.

The green boxes yield a combined normalization shift of -116.5 for PB10. This large shift, due predominantly to PB10's much lower hotspot temperature, is the largest combined normalization shift compared with any of the other tests (PB 1, 2, 4).

Table B.5. Spreadsheet formulas for uncertain experimental quantities in modified version of Equation 5.23 for test PB10**.**

realization j	$\Delta Pfail_{measPB10,j}$ = A2j, systematic betw. PB1,4,2,10	$w_{actPB10,j}$ = [0.019, 0.021] inches	$\Delta w_{PB10,j} = w_{nomPB4} - w_{actPB4,j}$ = 0.02" – D3	$\frac{\partial(Pfail_{PB10})}{\partial(w)}_j$ = 1.14*A5j, correlated betw. PB1,4,2,10	= (D4*D5) $_j$ psi	$\Delta T_{meas\text{-}TC/DAQ,j}$ =[–0.0025*648, +0.0025*648]°C	$\Delta T_{meas\text{-}contact,j}$ = 0.92*A8j, correlated betw. PB1,4,2,10	$\Delta T_{TC4location,j}$ = [–15, 15]°C	$\frac{\partial(Pfail_{PB10})}{\partial(T@fail\_point)}^j$ = A10j, systematic betw. PB1,4,2,10	= (D7 + D8 + D9) $_j$ *D10 $_j$	= D10*(707C – 648°C)	= 647 psi + (D2 + D6 + D11 + D12)$_j$psi = $Pfail_{PB10}(\vec{x}_{nomPB1})_j$
D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13
1												
2												
…	…	…	…	…	…	…	…	…	…	…	…	…
J												

Table B.6. Spreadsheet realizations for uncertain experimental quantities in test PB10.

j	D2=A2, ΔPf	D3=PB10w	D4=Δ0.02"	D5=1.14*A	D6 = D4*D	D7=ΔTC_D	D8=0.92*A	D9=ΔTC_lc	D10=A10, s	D11=D10*	D12=D10*	D13=647+D2
1	7.76	0.0197	0.0003	34692.52	11.64	-1.26	0.45	14.93	-2.07	-29.28	-122.31	514.81
2	-5.88	0.0208	-0.0008	42755.96	-34.42	0.52	1.03	-13.17	-1.87	21.71	-110.20	518.21
3	-0.69	0.0204	-0.0004	29511.53	-12.22	-0.54	1.41	9.46	-1.83	-18.88	-107.84	507.38
1000	6.28	0.0208	-0.0008	32840.70	-27.41	-0.28	0.05	2.34	-1.99	-4.21	-117.66	503.99
	avg	avg	avg	avg	avg	avg	avg	avg	avg	avg	avg	avg
	0.00	0.0200	0.0000	36568.73	0.39	-0.04	0.79	0.65	-1.94	-2.69	-114.22	530.48
	stdev				stdev					stdev	stdev	stdev
	5.76				20.79					17.22	7.54	29.03
												max
												605.84
												min
												447.43

The grey boxes in Table B.6 show the relative contributions to normalization uncertainty. Potential wall thickness differences are the largest uncertainty contributor (standard deviation of 20.8 psi). This is similar to the value for PB2 but significantly smaller than the values 28.1 psi for PB1 and 39.1 for PB4.

The square root of the sum of squares of the contributing standard deviations in the grey boxes is 28.6 psi. This closely agrees with the value 29.0 psi in the pink box, calculated directly from the realizations in column D13. The normalization uncertainty for PB10 is slightly larger than for PB2 but significantly smaller than the values 33 psi and 43 psi for PBs 1 and 4, which are driven mostly by their larger uncertainties of wall thickness.

Appendix C. Simpler Experimental Data Normalization for Problems Involving Suitably-Random Measurement Errors or Insignificantly Small Random Measurement Errors

Consider a situation where the exact locations of failure initiation in the four pipes were somehow known a-priori. Then the temperature and wall-thickness values at the failure locations could have been measured. If errors in these measurements were the normally distributed random perturbations about the actual values in the four tests, then the experimental data normalization method discussed in Section 5.3 could be simplified and a significant decrease of overall experimental uncertainty in Figure 41 would result. Columns A9, B9, C9, and D9 in tables 18, 20, B.3, and B.5 (above) would be eliminated because the uncertainty associated with failure location is eliminated. But more to the point, a generic simplification is possible given the normality of random measurement errors over the tests. For example, let the (uncorrelated) temperature measurement-error uncertainty ranges in columns A7, B7, C7, D7 be replaced by uncorrelated normal PDFs with ± 3 standard-deviation extents that lie at the ends of the interval ranges. Let analogous PDFs replace the (uncorrelated) wall-thickness uncertainty intervals in columns A3, B3, C3, and D3.

If the normalization protocol in Section 5.3, above, is followed, these normal PDFs would be sampled for potential combinations of random measurement errors in the four tests. This is still legitimate under the new conditions of this section. But when linked with the 0.95/0.90 tolerance interval approach to account for limited numbers of replicate tests (see Section 5.3.3), the processing is unnecessary and likely yields overly conservative results. Numerical experiments by the first author of this chapter show this for generic problems. Conservatism is shown to increase as the number of replicate tests increases, as the number of measurement error sources increases (represented by the wall and temperature measurement sources here), and as the magnitude of potential measurement errors (spread of the error PDFs) increases.

Therefore, if the conditions described in the first paragraph of this appendix did exist, we would recommend altering the normalization protocol from Section 5.3. The random component of measurement error would simply be ignored. This is equivalent to having constant values in columns A3/B3/C3/D3 and A7/B7/C7/D7 instead of randomly sampling from the uncertainty ranges presently prescribed in the columns. The fixed values would be at the midpoints of the said uncertainty ranges, as these correspond to the means of the normal PDFs of measurement errors postulated above. Based on the study by Romero et al. [11], it would not be surprising if this simplified treatment is also applicable for random measurement errors governed by reasonably symmetric central-tending PDF shapes and even uniform PDFs over the stipulated uncertainty ranges. But numerical experiments have not yet been conducted to assess this.

(If using a different approach such as the Pradlwarter-Schueller method [14] to compensate for limited numbers of replicate tests, it is presently *not* recommended that random measurement errors be ignored even if normally distributed. More research is necessary. For example, over the test cases by Romero et al. [11], it was found that the Pradlwarter-Schueller approach is substantially less reliably conservative than the (approximately 90% reliable) 0.95/0.90 TI approach. Sampling for the random measurement error possibilities, per the protocol in Section 5.3, may serve to increase to acceptably high levels the reliability of using the Pradlwarter-Schueller approach. But this has not been investigated or established yet.)

If random uncertainties among replicate tests are not present or are insignificantly small, or do not need to be sampled because they are normal or otherwise suitably distributed and a TI approach is being used, then a simplified approach can be taken (see for example Ref. 28, calibration-data section) to process the *systematic* uncertainties in the problem if the following additional restriction is met: no scaling of the

systematic uncertainties is present; that is, a value of 1.0 exists in place of scale factors 1.08 and 0.95 in columns C5 and C8 (Table B.3) and 1.14 and 0.92 in columns D5 and D8 (Table B.5).

The simplifications discussed in this appendix do not apply to the PB problem. The wall thicknesses and temperatures at the failure locations vary randomly from test to test. But the associated interval uncertainties in columns A3/B3/C3/D3, A7/B7/C7/D7, and A9/B9/C9/D9 are estimated by rather crude techniques—the variations are *not* actually measured (with known approximate symmetric probability distributions for measurement error about the measured values). So, the interval uncertainties reflect two types of uncertainty. A large element of epistemic uncertainty exists in the interval descriptions that attempt to characterize possible random variability in the tests. This brings into question potential similarities to the generic problems previously studied and the scenario outlined at the top of this section, which have no error or epistemic uncertainty in their PDF characterizations of random variability in the replicate tests. So for the PB problem, it is most prudent to use the more conservative interval-respecting UQ treatment in Section 5.3.

Appendix D. Brief Comparison of Real Space Model Validation Capabilities for PB Problem vs. Several Other Established Model Validation Frameworks

Here we make some observations concerning the applicability of several model validation methodologies to the pipe bomb validation problem. The Real Space approach appears to uniquely have the required features to appropriately handle all the attributes of the PB validation problem.

- It appears that the American Society of Mechanical Engineers (ASME) VV20 methodology [22] is not configured to assess models of stochastic phenomena like in the PB validation problem. Another unique feature of the RS method is that it separates uncertainty contributions into traveling uncertainties (here stochastic material behavior variability that is an intrinsic aspect of the traveling constitutive model being validated) and non-traveling uncertainties that exist only in the validation exercise. This separation is incorporated because traveling and non-traveling uncertainties have different implications for model predictivity in post-validation use of the model, see the work by Romero [8].

  Nonetheless, for problems involving non-traveling epistemic uncertainties, Eq. 5.4 with linear UQ ultimately yields the combined "validation uncertainty" in Ref. 22 when its linear UQ version is used and the non-traveling uncertainties from the simulations are taken into account per the different approaches in the two methods—provided the conditions in the examples in the ASME standards [22] are met: A) uncertainty is assumed to be probabilistic and represented and propagated accordingly; B) only non-traveling epistemic uncertainties exist in the models and experiments; C) the system of interest (in the models and experiments) has no significant degree of stochastic behavior/response affecting the quantities of interest. This is a reassuring corroboration of both methods (RS and Ref. 22) for the subset of conditions cited, given that their derivations come from very different conceptual approaches.

- The Oberkampf & Barone approach [39] addresses the uncertainty elements shown at left in Figure 39 for stochastic system behavior/response and confidence levels from the limited number of tests. But the measure of the stochastic behavior is limited to uncertainty of mean response. Furthermore, the work by Oberkampf and Barone [39] does not address most of the uncertainties in the category "experimental factors in the tests" shown in Figure 39. Systematic experimental uncertainties are ignored altogether. For model prediction, a single deterministic model run is made for comparison against the uncertainty PDF of the experimental mean (a very limited basis for assessing accuracy and adequacy of models of stochastic phenomena).

- The validation approaches in References 40 and 41 focus on models of stochastic phenomena or systems. The "area" validation metric compares cumulative density functions (CDFs) of predicted and experimental responses. A numerical value for discrepancy between experimental and predicted CDFs is obtained. A value of zero indicates perfect agreement at all CDF percentiles. However, for non-perfect agreement it is not clear how non-zero values relate to more directly interpretable measures of prediction error such as mean prediction error, error of predicted standard deviation of response, or error of predicted percentiles of response. Furthermore, the methodology does not directly address epistemic uncertainty (including accounting for the bias toward underestimating experimental variability) from the limited number of tests—although the uncertainty indicated by the area metric is generally found to be larger for fewer tests, everything else being equal. The remaining element shown at left in Figure 39 (experimental factors in the tests) is only partially addressed. Random variations of experimental inputs and of measurement errors in multiple replicate experiments are treated, but systematic experimental uncertainties are not. In this regard Refs. 40 and 41 incur somewhat more *Model User's Risk* concerning systematic uncertainties than the RS and ASME VV20 approaches do, see Romero [8]. The latter approaches take a more conservative slant, mitigating *Model User's Risk* by explicitly accounting for systematic measurement uncertainties.

- Two validation approaches are presented in the ASME VV10 methodology guide [42]. One is based on the approaches in Refs. 40 and 41, while the other does not address stochastic elements of behavior in the model predictions, so is very limited.

- Finally, in contrast to the other validation approaches mentioned, the Real Space methodology recognizes that models can have traveling *epistemic* uncertainties that are an intrinsic part of the model. These are not present in the PB validation problem but occur in Refs. 28, 29, and 43 for example, as parametric uncertainties in physics modeling parameters, and in Ref. 29 as multiple plausible discrete submodels for turbulence. The RS method treats these differently from the non-traveling epistemic uncertainties contributed by testing because they have different implications for model predictivity in post-validation use of the model, see Ref. 8.

Hypersonic Air-Breathing Propulsion Systems: Key Physics and the Categorization of Modeling & Simulation Approaches

David W. Riggins

Missouri University of Science and Technology, Rolla, MO 65409

Abstract

The airframe-integrated scramjet, including the dual-mode scramjet, is described. Key physical processes, critical technical issues, and challenges for the modeling, simulation and analysis of scramjet flow fields are reviewed. Example results from a low-order modeling analysis of a scramjet engine/vehicle in flight are shown in terms of typical flow characteristics and engineering performance metrics and parameters. Brief comments on credibility of simulations and some general observations regarding important issues in the modeling and simulation of high-speed air-breathing propulsion systems are provided.

1. Introduction

The physical phenomena that determine hypersonic air-breathing propulsion system performance and operability represent significant challenges for current modeling and analysis capabilities. These challenges also frame the engineering hardware design constructs and methodologies that are required in order to configure, build, and test engines that can achieve adequate thrust production at hypersonic flight velocities. High-fidelity modeling and simulation techniques and approaches for this flight regime are often at the very limit of current capabilities and resources. Some modeling and simulation requirements are arguably still beyond current capabilities—at least, in terms of ensuring a comfortable level of confidence. The phenomena and requirements that are specific to modeling high-speed air-breathing flight systems include (but are not limited to): the generation of intense shock waves and their interactions with vehicle surfaces; the need to handle excessively high heating loads, both overall and local, from the airframe and the engine; the presence of extreme chemical non-equilibrium effects in both the external air and within the propulsion system; hypervelocity fuel-air injection, stabilization, mixing, and combustion issues; and the necessity of an extraordinary degree of airframe-engine integration in order to facilitate both thrust and lift.

The objective of this chapter is to provide a brief overview of both key physics and current modeling categorization associated with hypersonic air-breathing propulsion systems. In the descriptions of key physical processes, the emphasis is on airframe-integrated scramjet and dual-mode scramjet engine performance and operability. Emphasis is also placed on those issues attendant to modeling, simulation, and analysis of scramjet and component flow-fields (i.e., not on issues relevant to structural modeling and simulation).

This chapter is organized as follows: Section 2 provides an overview of the dominant characteristics of high-speed air-breathing propulsion systems, with emphasis on scramjet and

dual-mode scramjet engines.[1] Section 3 itemizes and illustrates critical physical phenomena and modeling challenges encountered in (or characteristic of) scramjet engines, including dual-mode scramjet engines, isolator flow fields, and engine "unstart" issues.[2] Section 4 then illustrates the organizational hierarchy of codes in terms of level of fidelity, parametric ability, and ability to model physical scales. Example results produced by a lower-level engine modeling tool (originally presented in Reference 4) are then given in Section 5 for an airframe integrated scramjet powered vehicle operating at Mach 8 at two different throttling conditions. This information is included primarily to give some perspective to the order of performance and flow metrics expected for operating scramjets within a modeling effort. Section 6 provides concluding remarks on factors affecting credibility of modeling and simulation tools and analyses; other issues, including the impact of the second law and uncertainty quantification in model development and applications, are also briefly discussed.

2. High-Speed (Hypersonic) Air-Breathing Propulsion

A ramjet, like a high-speed turbojet, is characterized by deceleration of the flow to subsonic velocity for combustion and a physical throat in the nozzle/expansion system downstream of the burner. As flight Mach number is increased, ramjet performance drops off due to 1) the need to maintain acceptable material temperatures at the end of the combustor (as well as disassociation problems) and 2) increasing flow losses (irreversibility) associated with the supersonic to subsonic deceleration process in the inlet. Most of the emphasis for hypersonic air-breathing propulsion (flight Mach number greater than 5) has been focused on the development of the supersonic ramjet, or scramjet, which has no physical throats.

Above a flight Mach number of about Mach 7, scramjet engine operation is characterized by predominantly supersonic flow throughout the combustor. Areas of recirculation are generally small and are mainly confined to the aft surfaces of intrusive fuel injectors or associated with very local regions around flush-wall jets or rearward-facing surfaces. There is little upstream interaction in such a flow; limited interaction may occur due to information propagation upstream through relatively thin wall boundary layers, but this limited interaction does not affect the bulk upstream flow in any significant fashion. The critical ratio of heating rate associated with fuel-air combustion in the burner to the total enthalpy rate entering the engine is very small. This fact results in low-pressure rises in the downstream burner (even with modest cross-sectional area increases), with the result that large-scale subsonic regions (or regions of separated flow) are not established. The combustor entrance conditions are then essentially de-coupled from the flow field within the combustor. Furthermore, the degree of diffusion throughout the inlet/combustor system is not sufficient to cause the flow Mach number to approach 1 (even within the combustor); therefore, the issue of choking and engine unstart is generally avoided, at least for the operational performance envelope of the engine. For the high-speed range (flight Mach greater than 7), then, there is little ambiguity—but considerable challenge—for the engineer, at least in terms of understanding the broad design issues, requirements, and dominant flow features for the scramjet in order to predict and analyze performance and operability. Due to increasing

[1] For a more detailed and comprehensive description of various high-speed engine configurations and issues, see Curran [1].

[2] Technical treatment of many of these same issues can be found the work by Heiser and Pratt [2]; in addition, Curran [3] provides an invaluable source reference for scramjet analysts, designers, and modelers.

losses and heat loading, and the resultant decreasing thrust-to-drag ratios—as well as a host of other performance and system issues—it is highly unlikely that scramjets can operate effectively for flight Mach numbers greater than Mach 12 or 13.

Due to the need for adequate inlet and nozzle lengths to sufficiently compress and expand the high-speed airstream processed by a scramjet engine, as well as the requirement to provide aerodynamic lift at hypersonic flight speeds, most of the work in this area has focused on the lifting-body/waverider airframe-integrated scramjet vehicle configuration. This tightly-integrated vehicle-engine configuration has, with justification, been called a flying engine. The airframe-integrated concept has proved highly successful as evidenced by the design and historical flights of the X-43A [5] and the X-51 [6]. In such a vehicle, the propulsion system is encompassed by the entire bottom of the vehicle (from leading edge to exit plane), as sketched in Fig. 1.

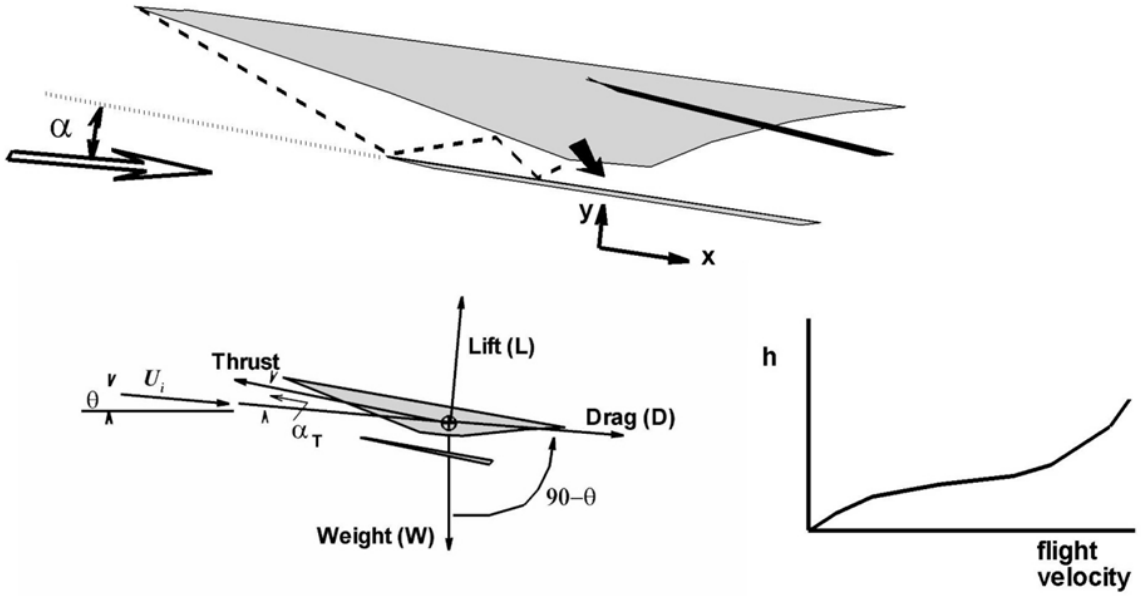


Figure 1. Airframe-integrated scramjet-powered vehicle configuration with air-breathing trajectory sketch.

Specifically, the forebody/inlet serves to decelerate the airflow that is captured between the lower forebody and the cowl leading edge (bottom); this deceleration to a lower supersonic Mach number is realized through a series of oblique shock waves prior to entering the internal inlet/combustor system downstream of the cowl leading edge. Fuel is injected, mixed with the predominately supersonic airstream in the combustor, and burned. Fuel injectors can be intrusive or flush wall. Downstream of the combustor, the nozzle/afterbody then serves as the main expansion (thrust) surface for the vehicle. Generally, however, the combustor also expands due to operability requirements (pressure gradient effects), and therefore also generates thrust; this is augmented by possible momentum and injector base effects associated with fuel injection—the latter effect (rocket effect of injection) becoming increasingly important at high flight Mach numbers. The bottom cowl of the airframe-integrated scramjet is generally truncated well upstream of the end of the (upper body-side) afterbody due to weight and friction issues, causing an expanding exhaust plume to develop beneath the after-body of the vehicle.

Also shown in Figure 1 are the main forces (lift, drag, thrust, and weight) acting on the vehicle; the vehicle climb angle; and a representation of the air-breathing corridor (necessary for air-breathing systems, and generally defined by dynamic pressure limitations) in altitude-velocity space. Note that this corridor exists because it is necessary to keep drag and heat transfer in check (pushing flight requirements up in terms of altitude) while simultaneously maintaining adequate air mass capture into the engine (pushing flight requirements down in terms of altitude).

The design and development of hypersonic air-breathing flight requires an intense focus on integrating and optimizing the entire vehicle system, and therefore requires exceptional focus on multi-disciplinary analysis (MDA) and optimization (MDO). Historically, MDA and MDO have been largely somewhat ad-hoc. For a representative paper on the role of MDA and MDO in the design and development of high-speed air-breathing flight vehicles, see Bowcutt [7].

As the flight Mach number for a scramjet engine is lowered below Mach 7 to Mach 4 or 5, some significant effects drastically change flow character, engine performance, and engine operability. As a result, configuration changes to the engine flow path itself are required in order to allow flight in this regime. The engine designed for operation in such a range is known as a dual-mode scramjet, which is indicative of the ability to have both ramjet/scramjet operational characteristics.

For flight Mach numbers less than about 7, the ratio of heating rate (due to fuel-air combustion) to entering total enthalpy rate becomes large. Significant pressure rise is experienced in the combustor. This pressure rise due to heat release is coupled with the decrease of the Mach number in the combustor such that sonic flow (thermally choked flow) can occur in the combustor, even with an expanding area. Eventually, the resulting flow in the combustor is back-pressured enough that significant interaction occurs upstream of the fuel injection location/combustor entrance. This interaction generally develops as an upstream oblique and/or normal shock train with associated large recirculation (or low velocity) regions adjacent to walls. For a high-speed scramjet operated at such conditions (mid-speed flight Mach numbers of 4 to 7), this interaction will immediately propagate into the inlet. Due to the inherent instability associated with maintaining a shock system in a converging flow path, inlet/engine unstart will result.

Inlet unstart is characterized by the existence of strong shock systems upstream of the engine and significantly reduced mass capture into the engine, with dramatic attendant decreases in performance. For this reason, a component known as an isolator is incorporated between inlet and combustor for dual-mode scramjets. This component, first proposed by Curran and Stull [8], is a constant area or slightly divergent duct, and buffers the inlet from upstream-propagating interaction from the combustor. The oblique shock system and associated pattern of recirculation regions associated with upstream interaction will stabilize in the isolator such that started engine operation can be maintained, even with extensive upstream interaction (provided the isolator is long enough).

Other techniques that facilitate dual-mode operation include: 1) backward-facing steps at the entrance of the combustor proper or at various locations along the combustor, in order to isolate or stabilize recirculation and slow the development of extensive upstream interactions); 2) diverging combustor walls in order to relieve the combustion-generated pressure rise, delay the development of a possible thermal choke, and facilitate reestablishment of supersonic flow subsequent to thermal choking; and 3) axially staged injection, which distributes heat release appropriate to combustor geometry and varying inflow conditions.

The variance of flow features observed for the dual-mode scramjet engine operated at mid-speed conditions is therefore astonishingly large. For example: bulk flow may or may not remain supersonic; thermal choking may or may not occur; upstream recirculation bubbles may extend far downstream into the combustor proper (in fact, some rig-specific experimental data indicates recirculation blanketing the entire combustor); localized oblique and normal shocks can truncate the oblique shock train in the isolator or at the combustor entrance; and the isolator flow can form an effective distortion-induced aerodynamic throat. Such flows are highly complex, strongly coupled, massively elliptical in character, and contain complex turbulence mechanisms and physics; they represent an extreme challenge for analysis, numerical simulation, or experimental work.

3. Critical Physical Phenomena and Modeling Challenges in Scramjet Engines

This section provides a brief summary of some of the critical phenomena and modeling challenges associated with the flow physics encountered in scramjet engines. For this discussion, emphasis is placed on propulsive performance and operability. Performance is defined here as the delivered thrust and efficiency of the engine. Operability is defined as the characteristics of the engine in terms of how it operates across the design space, and how the physical fluid and thermodynamic modes underlying the delivered performance evolve or change as conditions are changed, primarily across the design space of interest. Although the following discussion is, for sake of convenience, presented in terms of component subsystems of the propulsion system itself (forebody/inlet, isolator/combustor for a dual-mode scramjet, combustor, and nozzle/afterbody), the entire engine is best viewed—in terms of understanding realizable and meaningful performance—as a seamless and continuous flow path from tip to tail. This perspective mandates that component subsystems (however and wherever defined) be designed, analyzed, and optimized from the standpoint of the overall engine. In this correct sense, the forebody/inlet system is best viewed as ‘merely’ the upstream section of the combustor and the nozzle/afterbody as ‘merely’ the downstream section of the combustor.

3.1. Forebody/Inlet

The purpose of the forebody/inlet in a scramjet engine is to efficiently induct air into the engine and maximize the mass of air captured, which is in turn directly proportional to delivered thrust. The forebody/inlet prepares the captured airstream for combustion by decelerating it and raising its pressure and temperature as efficiently as possible (i.e., with as little loss—that is, entropy generation—as possible). The deceleration is obtained via some number of oblique shocks generated from the turning of the upper forebody surface; the Mach number entering the internal inlet is then substantially reduced from the flight Mach number. By the nature of the deceleration (diffusion) process occurring in the inlet, the static pressure gradient is adverse, with attendant issues involving possible shock-induced separation of boundary layers along the surfaces of the forebody/inlet. The details of the flow features for shock-boundary layer interactions are generally driven in turn by details of turbulent characteristics of the flow; the effectiveness of modeling such effects and resultant impacts are therefore highly dependent upon turbulence modeling capabilities.

Also, due to the increased skin friction and heat transfer associated with turbulence, as well as the character of turbulent flow in terms of minimizing separation, it is important that the issue of transition from laminar to turbulent flow be understood or addressed as much as possible. Transition in high-speed flows on actual vehicles is a complex and difficult topic; transition prediction methodology is therefore not very effective. Furthermore, for an actual flight vehicle,

high-speed flow transition is highly three-dimensional in terms of inception, growth, development, and location. For small-scale non-reusable systems, the flow can be purposely tripped upstream to fully turbulent in order to minimize the operability questions that arise with laminar or transition flow; however, this is problematic for larger re-usable vehicles due to skin friction and heating. In any event, transition location and characteristics on inlet/forebody surfaces can significantly alter the compression characteristics and skin friction and heating loads experienced by the vehicle. Modeling transition has been and will remain a major limitation/challenge in modeling high-speed air-breathing propulsion systems.

3.2. Dual-Mode Scramjet Isolator/Combustor

The fundamental physical processes that drive high-speed engine starting (and unstart) are, in principle, well-documented and understood. A started engine is one in which the maximum possible nominal mass flow rate is being captured by the engine with any normal shock systems (or contraction, or shock-induced progression to subsonic flow) being fully internal; this condition is supercritical. Conversely, an un-started engine is one that has reduced mass flow rate, with subsonic flow entering the internal flow path; it is generally characterized by external normal shock systems; this condition is termed subcritical. If ΔS_{crit} is defined as the entropy (per unit mass of airflow) transfer and generation from the inflow of an engine to a station within the engine, for which the flow at the internal station is just driven to Mach 1 (i.e., choking), the following relation for quasi-one-dimensional flow can be written as follows:

$$e^{\frac{\Delta S_{crit}}{R}} = \frac{A}{A_{\infty,ref} M_\infty} \left[\frac{\left(1+\frac{\gamma-1}{2} M_\infty{}^2\right)\left(1+\frac{q+w}{C_P T_{t\infty}}\right)}{1.2} \right]^3 \quad (1)$$

Here, q and w are the heat and work added per mass between the inlet and the station of interest; A is the cross-sectional area at the station of interest; and $A_{\infty,ref}$ is the nominal inlet capture area. For any entropy generation (or transference) greater than this critical entropy for the given parameters of entrance Mach number, heat, work, and altitude, the engine will not be able to pass the mass flow rate associated with the nominal capture area—the flow will develop a shock system, which will disgorge from the engine and result in subsonic entering flow with attendant spillage (i.e., reduction of mass flow rate and large total pressure losses through the external shock system). This fact mandates decreasing contraction ratio (free stream capture area to internal inlet throat area) for decreasing flight Mach number, in order to maintain started flow for a conventional high-speed scramjet.

The isolator decelerates the flow upstream of the combustor through contained oblique/normal shock systems. Attendant adjustments occur in terms of overall entropy generation within the isolator/combustor subsystem, allowing the engine to maintain started flow at higher fuel flow rates (i.e., more heat release is allowed than a scramjet without an isolator at the same flight Mach number). This occurs without a physical upstream throat (necessary in a conventional ramjet), even though the flow in the combustor has choked and can be in full subsonic mode. Therefore, the dual-mode scramjet can maintain started flow for combustors operating at either subsonic Mach (ramjet mode) or low supersonic Mach (scramjet mode), as shown in Fig. 2—bridging the critical flight Mach number range from 4 to 7.

Although the shock systems in the isolator generate entropy increases, the lower Mach numbers associated with the downstream combustion can reduce the combustor Rayleigh entropy increases even more; in ramjet mode, the isolator (with its separated flow region and resulting flow

contraction) fluid-dynamically mimics, to some degree, a converging/diverging throated system in a ramjet. As noted, the isolator allows far higher fuel flow rates in the combustor at the mid-speed flight Mach numbers (4 to 7), while maintaining started flow, than would be possible without the isolator; the isolator thus allows the scramjet to operate at flight Mach numbers less than hypersonic—extending the operability range of the scramjet down (perhaps) to the upper limit of turbine engine technology.

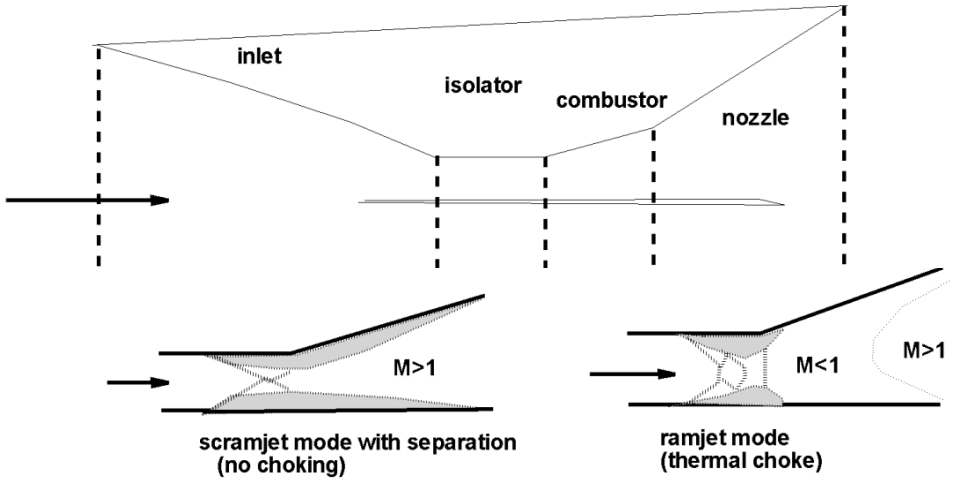


Figure 2. Dual-mode scramjet (with isolator) and flow patterns.

There is an increasing static pressure gradient associated with fluid dynamic contraction and losses in the inlet/isolator system, as well as further back-pressuring of the inlet/isolator due to heat release downstream in the combustor. In addition, there is the inevitable formation of upstream-of-combustor shock wave systems (incident and reflecting) for low enough flight Mach numbers. Therefore, the characteristics of the incoming boundary layer on isolator operation are dominant in terms of isolator design, especially for determining the characteristics of the oblique shock train within the isolator, the length of that train, and the details of the separated flow regions that envelop the train. Oblique shock waves, forming adjacent to and impinging on the boundary layer on walls within the isolator, induce very large adverse pressure gradients in the fluid, which are superimposed on the existing adverse pressure gradient. Heat transfer spikes and separation are the result. Note that laminar flow, although providing lower skin friction and heat transfer, is far more susceptible to separation than turbulent flows. Therefore, in order to minimize the separated flow regions as much as possible within the combustor/isolator due to shock-boundary layer interaction, it is generally desired to have turbulent flow within the isolator.

At this time, the accurate prediction of isolator performance, fueling limitations, and incipient engine unstart for actual flight vehicles operating in the mid-speed range is a dominant and daunting challenge. Some representative work in this area can be found in Refs. 9-12. Turbulence characteristics throughout the flow path (including within the upstream inlet) critically impact boundary layer evolution, shock structures, and the growth and stabilization of separation regions within the isolator and combustor. Furthermore, the downstream fuel-air mixing physics and chemical kinetics drive strong upstream interaction effects through the pressure rise associated with heat release in the combustor and resultant interactions propagating upstream along wall boundaries. Hence, mixing and combustion modeling in the combustor itself—also strongly driven by the evolving turbulent characteristics of the flow entering the combustor—are critical for modeling efforts.

Figure 3 shows some results from a very simple two-dimensional computational fluid dynamics (CFD) example that clearly demonstrates the dual-mode (upstream interaction) effect in an isolator-combustor analog.

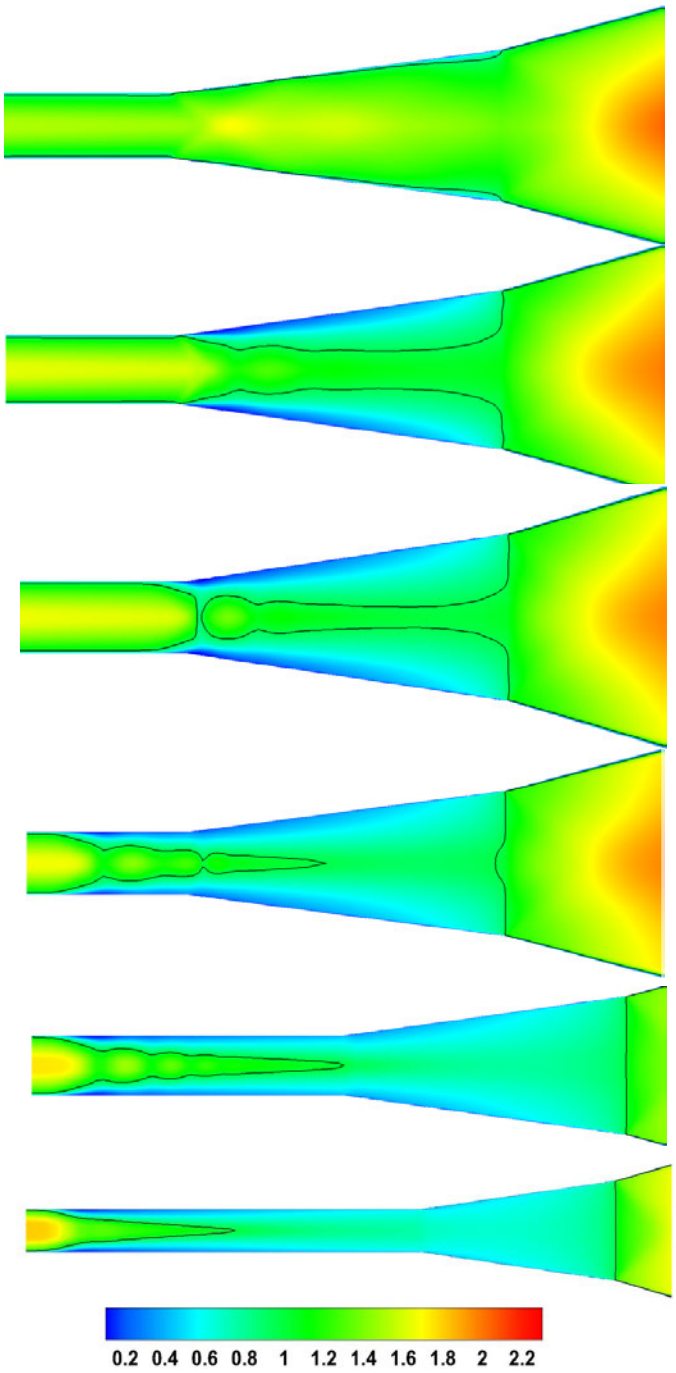


Figure 3. Contours of Mach number (with sonic line shown) for various total heating rates for uniform heating per volume CFD-generated flow fields. The heat addition shown in each subfigure (from top to bottom) is 4, 10, 12, 14, 16, and 17 megawatts.

In this study by Riggins et al. [13], increasing amounts of distributed heat (rather than fuel injection with associated chemistry) are added into an expanding area combustor downstream of a long constant area isolator. Isolator core flow inflow Mach number is 2 with isolator inflow conditions corresponding to flight Mach of 6. The upstream impact of increasing heat release can be plainly seen in this sequence of contours of local Mach number (along with the sonic line showing the boundary between subsonic and supersonic flow regions). For this particular study, the highest heat release (bottom contours) are at an incipient unstart condition. Note that the lower contours in this figure (those characterized by more heat release in the combustor) show more of the upstream isolator geometry than the upper contours (lower heat release); this is due to the increasing amount of upstream interaction and isolator involvement for the high heat release cases.

3.3. Scramjet/Dual-Mode Scramjet Combustor

The purpose of the scramjet combustor is to inject, vaporize (if necessary), adequately mix, and combust fuel with the inlet airstream as rapidly and efficiently as possible, i.e., with the lowest practical irreversible/non-ideal entropy generation (or equivalently, the lowest practical total pressure losses). However, depending on flight Mach number and vehicle heating loads, a secondary but critical purpose of the high-speed scramjet combustor is to inject and entrain an additional increment of fuel associated purely with vehicle thermal management; this increment can drive the overall fuel mass flow rate in the combustor well above stoichiometric. For example, it is a rule of thumb that this situation will occur for flight Mach numbers of 10 and greater for hydrogen-fueled scramjets. A useful review of some topical (and still evolving) issues in modeling and simulation of high-speed reacting flows was provided by Baurle [14].

The velocity of the core flow through a high-speed scramjet combustor is quite close to the actual flight velocity of the vehicle, such that the residence time of a fluid element in the combustor is astoundingly small for practical combustor lengths. This presents the challenge of taking two separate streams—fuel and air—and adequately mixing and burning them within that residence time (or equivalently, across the combustor length). Many injection strategies have been proposed and used to accomplish this, including flush wall injection, ramps, and struts. Generally, techniques that rely on parallel injection (parallel to the air) provide an advantageous thrust component, due to the momentum of the fuel injected. At very high flight Mach numbers this "rocket" effect becomes very significant, in comparison to the heat release associated with fuel/air combustion. In general, however, without the introduction of large-scale three-dimensional flow structures for enhancing macro-mixing, such techniques cannot provide adequate mixing of the fuel with the air for subsequent reaction, in usable combustor lengths. Cross-flow vortex generation associated with ramps, struts, and normal component of fuel injection has therefore been identified as critical to ensure adequate mixing in scramjet combustors. For examples, see References 5, 16, 17 and 18.

For instance, a basic injector ramp or strut protruding from the wall of the combustor, which has fuel injected downstream from its base, can generate strong cross-flow vortices due to compression-induced lateral pressure gradients; a similar effect is experienced by wall jets with a normal component of momentum. The mechanism of downstream mixing is facilitated by the macro-mixing caused by these vortices, which tend to stretch and greatly lengthen fuel-air interfaces; at these interfaces, the necessary turbulent mixing occurs. Turbulent (eddy-driven) micro-mixing is critical due to the insufficiency of purely laminar (micro-scale) mixing between two adjacent layers. Note that turbulent fluctuations and structures span length scales ranging from the microscopic level to spatial dimensions on the order of geometric features of the engine itself. Additionally, time scales of turbulent fluctuations can range across many orders of

magnitudes. Since these fluctuations and structures determine mass, momentum, and heat diffusion in the combustor, turbulence models that adequately describe the impact of turbulence on these diffusive processes are therefore critical in understanding and predicting the totality of the mixing process.

The main flow in a dual-mode scramjet combustor can either be predominately supersonic or subsonic throughout, depending on the engine fueling conditions, physical configuration, and the flight Mach number. For subsonic (ramjet mode) operation, there is a thermal throat downstream in which the heat release-versus-area schedule drives the exiting flow to choke and become supersonic downstream of the throat. The location of the thermal throat is dependent on the flight Mach number, isolator characteristics, fueling schedule, and combustor configuration (geometry). In this case, fuel-air mixing can be facilitated by the lower flow velocities; hence, longer residence times and Rayleigh losses (associated with exothermic heat release at finite Mach number) are significantly less than those incurred in a supersonic flow field.

Chemical reaction between fuel and air is characterized at high-speeds by non-equilibrium kinetics throughout the combustor, for which there is net production (or depletion) of individual species. However, even in high-speed engines there can be certain regions of the flow where the chemical temporal rates are much faster than fluid dynamic transport rates, so that chemical equilibrium (or near- equilibrium) exists. Conversely, in nozzles, flow residence times can be small enough that chemical reactions can sometimes be considered to be frozen, i.e., the time scales associated with the chemical reactions are significantly larger than the time scales associated with spatial transit of the reacting fluid through the nozzle, so the species production and depletion are “frozen” over relevant length scales. It is obvious that for scramjet combustors at high speeds, combustion is generally mixing-limited—the degree of combustion (usually measured in terms of heat release or product specie formation) is limited at any location by the specific degree of fuel-air micro-mixing at that location.

For dual-mode scramjet combustors operating in the ramjet mode, mixing-limited combustion is not an issue and combustion is kinetics-driven. However, the impact of fluctuating turbulence on the chemical kinetics can be very significant, especially in the dual-mode scramjet regime and in base regions at higher flight Mach numbers. Essentially, the relative time scales associated with the chemical reactions and the time scales associated with the turbulent fluctuations determine the details of the physics of—and of modeling approaches for—this interaction.

The number of elemental reactions necessary to meaningfully characterize the chemical kinetics can be very large, particularly for hydrocarbon fuel injection; hydrogen-air reaction is far better understood and far more easily characterized in modeling efforts than hydrocarbon chemistry, although enormous progress in hydrocarbons has been made in recent years. Hydrocarbon fuels may be suitably “cracked,” i.e., have characteristically long-chain complex molecules broken into simpler molecules prior to reaction; this is accomplished through transferring heat to the fuel from hot engine structures prior to the actual injection in the combustor. Note that there are also critical phenomena associated with necessary phase changes within the fuel system itself, or within the near field of injectors; fuel is generally stored in vehicle tanks as liquid and must ultimately be vaporized prior to reaction with the air in the combustor, i.e., the combustion is gas-phase.

In order to conveniently characterize mixing and reaction in scramjet flow fields, it is useful to describe or approximate both the degree of mixing between the fuel and the air and the degree of reaction of the mixed stream in terms of single-valued one-dimensional parameters—for example, parameters that vary solely with axial distance. The following discussion is a distillation of details

and formulations that can be found in other references, such as [4, 19]. Note that a bulk description of fuel-air mixing at an axial location should inherently contain multi-dimensional information, since mixing depends on the spatial distribution of fuel and air within the cross-flow plane.

There have been a wide variety of parameters used in the literature for describing the mixing of fuel and air. Probably the simplest approach has been to measure, or calculate (and plot) the maximum mass fraction of fuel or air present on each cross-flow plane, versus the axial distance downstream of the injection station. This method, while simple, does not include the integrated information necessary for a meaningful assessment of the reaction potential which exists in a flow with a specific cross-flow spatial distribution of fuel and air. A useful mixing efficiency should always represent the maximum potential combustion efficiency for a reacting system at a given state of mixedness, i.e., it should provide the upper limit on the combustion efficiency. In fact, from the standpoint of a one-dimensional computation, the only physically meaningful interpretation of the mixing efficiency is that it is an externally imposed upper limit on the combustion efficiency.

Any effort to define a useful performance measure should begin with the definition or identification of the maximum possible performance criteria. For the mixing efficiency for steady flows, a sufficient state for defining maximum mixedness certainly corresponds to the flow with absolutely uniform composition, i.e., 100% mixing is inevitable if all cells in a cross-flow plane have the same species composition (or, equivalently, if there are no gradients in species in any direction). For the combustion efficiency, the reference state of maximum reaction potential corresponds to complete reaction of the 100% mixed flow as described above. For H2-air systems, this is driven by the one-way global reaction, $H_2 + \frac{1}{2}O_2 \xrightarrow{yields} H_2O$. Although complete reaction overestimates the maximum possible heat of reaction (or the maximum total temperature achieved), it is nevertheless useful as a thermodynamic reference in the definitions of mixing and combustion efficiencies developed subsequently.

There are two viable approaches (with variants) to formally developing definitions of the mixing and combustion efficiencies in scramjet engines. The first approach is based on the mass flux (mass flow rate) of reacted fuel (for the combustion efficiency) or the mass flux of the potentially reacted fuel (for the mixing efficiency). These fluxes are normalized by the overall mass flux of reacted fuel assuming that complete reaction takes place within the uniform (100% mixed) stream. For a flow in which the overall fuel equivalence ratio is not equal to 1, such a reaction-potential-based definition of mixing efficiency yields complete (100%) mixing before the flow actually reaches a state of uniform composition. The second approach to defining mixing and combustion efficiencies is based on the actual heat of reaction and is the most rigorous definition, from the standpoint of thermodynamics, that can be devised for a single-valued engineering representation of the degree of combustion.

Irreversibilities associated with the combustor are caused by shocks, friction (both in-stream and wall), mixing of disparate species, non-equilibrium (finite-rate) combustion, and heat transfer. The vast bulk of entropy generation for a hypersonic air-breathing vehicle occurs within the combustor, where the trade between heat release and losses dominates design, analysis, and optimization. Due to this trade, it is axiomatic that it is not desirable to achieve near 100% mixing and combustion across the length of the combustor. Generally, mixing and combustion efficiencies are asymptotic with combustor length, i.e., mixing and combustion are rapid near the injectors then asymptotically approach a maximum downstream; typically it is not useful to

attempt to use extra length to obtain a mixing/combustion efficiency greater than 80% or so, due to losses accruing with that extra length.

Two major operability issues in scramjet combustors are ignition and flame holding (flame stabilization). In flight systems, ignition is generally achieved by piloting. Due to flame speed issues in high-speed flows, flames must be stabilized by the generation of suitable recirculation zones within the combustor using base regions behind injectors and cavities. Such zones can provide up to 10 times the residence times of that associated with the core flow adjacent to it. Piloting zone and ignition details are driven by turbulence characteristics associated with large-scale separation and recirculation regions, as well as by details of chemical kinetics physics and turbulent-chemical interactions.

3.4. Nozzle/Afterbody

The supersonic flow that exits the combustor is expanded in the nozzle and along the lower surface of the afterbody (downstream of the cowl trailing edge) and hence progressively reduces in static temperature and pressure. This reduction in temperature can allow chemical recombination of species to take place; recombination of dissociated species in the nozzle (and along the afterbody) can provide a significant supplement to the overall exothermic heat release and thereby increase the thrust delivered by the engine. The flow, however, can often chemically freeze in the downstream sections of the nozzle, depending on the chemical and fluid time scales. This scenario is dependent on a number of drivers such as flow speed, reaction rates, and the length of the nozzle/afterbody configuration. In addition, base pressurization on the afterbody surfaces can also significantly impact performance, through additional heat release associated with the upstream injection of fuel.

Because the nozzle/afterbody flow field is in a region of favorable pressure gradient, separation is generally not an issue; boundary layers usually remain attached and stable. The cowl is truncated well before the aft end of the vehicle to address friction, heating and weight; this results in an opening plume of nozzle exhaust below the vehicle. The degree of expansion across the entire subsystem is driven by the fluid conditions at the internal nozzle exit, by the evolution of the nozzle exhaust plume, and by the afterbody geometry. Pressures at the nominal exit of the engine stream (as the engine-processed stream-tube moves into the wake region aft of the vehicle) can range from less than 5 to more than 10 times the ambient pressure for hypersonic scramjet-powered vehicles.

3.5. Summary of Critical Physical Issues/Challenges in Hypersonic Air-breathing Propulsion

The top-level physics drivers on overall hypersonic air-breathing vehicle performance and operability are irreversibility (flow losses), degree of combustion (heat release), mixing of injected propellant mass, and heat transfer and heat load characteristics and limits (both local and overall heat load experienced by the vehicle). These drivers are all primarily associated with the propulsion subsystem which, given the nature of such a vehicle, overwhelmingly dominates the resulting list of physics-based modeling and prediction challenges and issues relevant to overall vehicle design, performance, optimization, and operability.

The propulsion system's primary purpose is to provide the required or mandated generation of axial (flight-direction) force (both accelerative and cruise); however, lift and moment characteristics of the vehicle—also driven largely by the propulsion system and its integration with the vehicle—are also of critical importance. Note that the specifics of the lifting-body/wave-

rider vehicle configuration itself flow from these drivers as well. The propulsion and fuel system also provide the thermal sink for the vehicle. Operability issues also relate back to these drivers; problems with operability (for example, engine unstart) can generally be characterized by sharp degrading gradients in performance, which are in turn linked to sharp transitions in irreversibility and/or combustion as flight condition or engine operating point is changed. In addition, the impact of flow distortion can be of critical importance for both performance and operability, and is of particular relevance for modeling and simulation of complex three-dimensional flows. Some interesting work that relates to distortion techniques and issues can be found in Refs. 19 and 20.

Figure 4 lists some widely recognized and important physics-based challenges and issues relevant to these drivers that exist in the hypersonic regime.

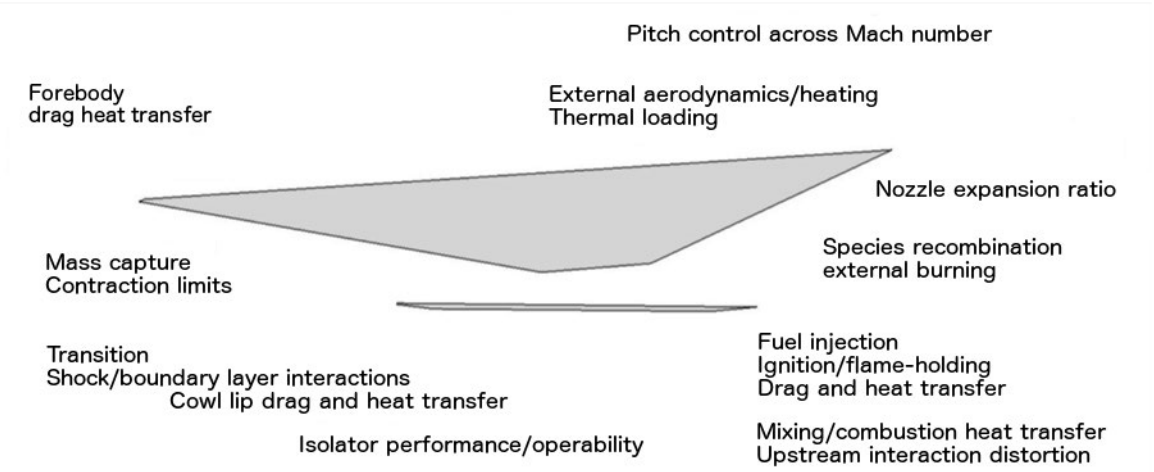


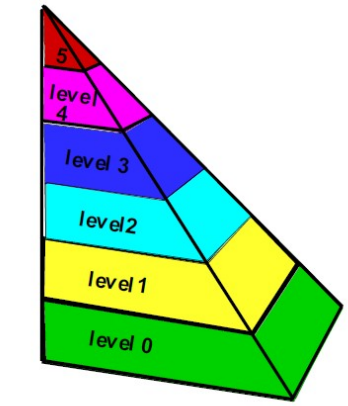
Figure 4. Critical physical issues/challenges in hypersonic air-breathing propulsion.

Two important facts regarding hypersonic flow are observed here as well; first, the hypersonic air-breathing propulsion system can almost without exception be assumed to operate in the continuum regime of fluid dynamics. Secondly, the flow through a high-speed engine can usually (not always) be assumed to be steady or quasi-steady in terms of responses to vehicle dynamics and motion; specifically, a vehicle traveling at Mach 10 has flow velocity through the engine on the order of 2 miles per second (i.e., extremely small residence times). However, there may be large-scale unsteadiness in isolated regions of the engine, such as the flameholders and isolator.

4. Hierarchy of Models Based on Parametric Ability, Physical Scale, and Fidelity

This section summarizes the hierarchy of analytical and computational tools (codes and methodologies) used in scramjet propulsion modeling. The hierarchy is categorized as follows: (1) physics represented via the governing physical laws; (2) complexity including time required for solution and solution strategy (parametric capability); and (3) fidelity reflected by degree of predictive capability. Note that there are three major physical phenomena important in scramjet flow fields that are modeled within these tools. These three modeling areas are: (1) turbulence modeling; (2) chemical kinetics modeling; and (3) turbulence-chemical kinetics interaction modeling. A detailed description and summary of relevant modeling issues categorized in terms of these areas are provided in the work by Riggins [4].

Figure 5 provides a graphical representation of the hierarchy of computational tools available for the analysis of scramjet engines. The code types are segregated loosely into levels 0 to 5, with level 0 corresponding to the simplest (cycle-type) tools and level 5 corresponding to direct numerical simulation (DNS) tools. Three axes are shown: parametric ability, physical scale of problem simulation, and fidelity (defined as true predictive ability). Note that level 0 codes have low fidelity but can easily represent wide physical scales and can be used in massively parametric studies. Conversely, level 5 tools presumably have high fidelity but are unrealistically limited in terms of physical scale and are simply not capable of parametric use of any type. Levels 0 through 3 represent current practical capability.



Level 5: DNS; not currently feasible for design

Level 4: LES, Full N-S with advanced turbulence models (beyond 2 equation); not currently feasible/used for design

Level 3: Full N-S

Level 2-3 PNS (marching NS) with embedded Level 3 N-S solvers in selected regions

Level 2: Marching N-S (parabolized) – small recirculation zones modeled

Level 1: Engineering codes – space marching, often Euler, 3-D effects captured by two-dimensional sweeps, isolator, recirculation modeling

Level 0-1: Cycle-Engineering combination codes/ analysis, includes 3 stream codes

Level 0: Cycle codes – component blocked or differential based, heavy modeling, limited distortion capability

Figure 5. Computational Tools – Overview of Hierarchy.

A separate issue not represented by this figure is that an analysis may use more than one of the various levels of physical modeling of mass, momentum, and energy diffusion (turbulence modeling) and chemical kinetics. For example, a level 3 (full Navier-Stokes) tool can be used with a very low-level turbulence model and global or complete reaction. Conversely, a level 0 (cycle) tool can be used with a very sophisticated chemical kinetics model—and often is, due to expanded resource capabilities when using such a tool. Therefore, actual analysis can be very complicated from the standpoint of physical modeling and chemical kinetics. As a common (almost universal) example: a relatively sophisticated turbulence model (a 2-equation model) is used in a level 2 tool for predicting diffusion of momentum, and is accompanied in the same analysis by the simultaneous prediction of the turbulent diffusion of mass using a spatially and temporally invariant Schmidt number.

5. Example Engine/Vehicle Model and Performance Description

This section provides a brief summary description of a simple modeling exercise for an airframe-integrated, scramjet-powered vehicle (no isolator) with a listing of representative performance obtained at flight Mach 8. Two throttling conditions (throttle up and throttle down) are used to generate these representative results.

Figure 6 shows a sketch of the model hypersonic vehicle considered in this example. The freestream (0.0) conditions are Mach 8 and 30 km (98,425 ft) altitude with a dynamic pressure of approximately 0.52 atm (1,100 psf). (A complete summary of the vehicle configuration, operational parameters, conditions, and assumptions used in the analysis is provided in Section 5.1.)

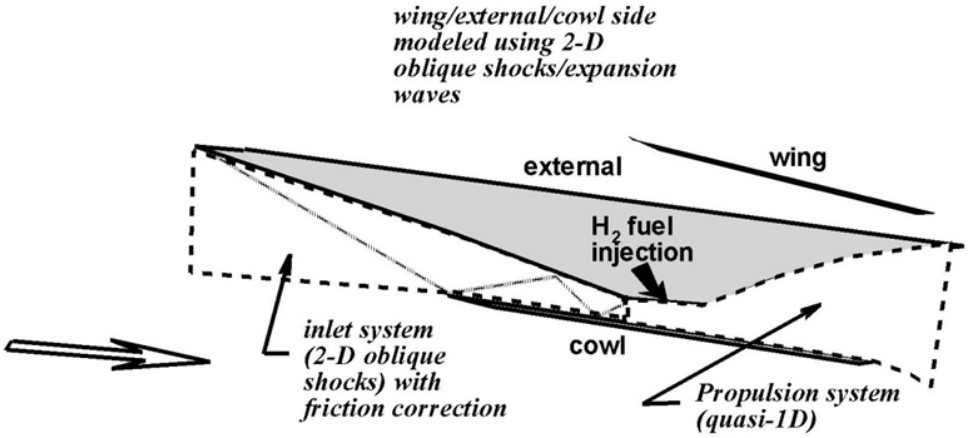


Figure 6. Scramjet Model for Cycle-Performance Example (not to scale).

The inlet/forebody used in these examples are analyzed using two-dimensional oblique shocks (with variable specific heats); the cowl is assumed to be adjustable (in geometry) such that the oblique shock train (4-shock system) is always cancelled at the fuel-injector/combustor entrance. This allows the use of quasi-one-dimensional cycle analysis in the combustor/nozzle/afterbody within the range of simple off-design missions and maneuvers. There is no isolator modeled for these examples.

The fuel-injection region and the scramjet combustor are considered to be constant-area (after an initial step at combustor entrance) for simplicity—in general there would be expanding cross-sectional areas in this region. Fuel injection is handled analytically by conserving mass flux, stream thrust, and total enthalpy. Realistic fuel-air mixing processes are then modeled by an input downstream mixing efficiency, which is simply an allowable upper-limit for the kinetically controlled combustion efficiency). For the examples shown here, hydrogen fuel is injected at 45 degrees downstream from flush-wall injectors. Finite rate reaction is simulated using a 7-species, 7-reaction hydrogen-air kinetics model. Chemistry is then assumed frozen at the nozzle entrance—that is, no further heat release is allowed in the nozzle. There is no plume modeling. Wall temperature from combustor entrance to engine exit is taken as 1000 K (1800 °R). Heat removed from the flow is returned via the injected fuel to maintain thermal balance.

The examples shown are for two different fueling conditions: a throttle down condition corresponding to a fuel equivalence ratio of 0.42 and a throttle up condition corresponding to a fuel equivalence ratio of 0.84. The freestream conditions are held constant except in the mission analysis.

To simulate a complete vehicle using a relatively simple model problem, the hypersonic vehicle has an oblique upper surface that generates an oblique shock wave and a "wing" of given area, which generates both lift and drag using simple shock wave theory. The wing angle of attack can be dynamically adjusted in order to provide the lift force needed (taking into account all other vertical force components). For the vehicle to cruise at given freestream conditions, all forces in both vertical and horizontal directions must each sum to zero; imbalance results in either acceleration or deceleration or change in altitude via the equations of motion. This model represents a simple point-mass approach to the vehicle (no trim moments accounted for) but provides insight into flow losses and engine performance information for both design point and mission studies. It is shown here in order to provide representative values of performance metrics and flow conditions expected for an airframe-integrated hypersonic air-breathing vehicle.

5.1. Summary of Scramjet Vehicle/Flow Parameters for Example Case

The lists in this section provide a summary of the vehicle configuration, operational parameters, conditions, and assumptions used in the example analysis presented above.

5.1.1. Freestream Conditions

- Freestream Mach: 7.93 (altitude 30 km, or 98,425 ft)
- Freestream velocity: 2,438.5 m/s (8,000 ft/s)
- Freestream temperature: 231.66 K (417 °R)
- Freestream pressure: 1,195.8 Pa (24.975 lbf/ft2)
- Dynamic pressure: 1,112 psf (53,250 N/m2)

5.1.2. Engine Geometry and Flow Assumptions

- Engine length: 8.2 m (26.9 ft) (2D engine/vehicle)
- Vehicle mass (initial): 7,850 kg (537.5 slugs)
- Inlet length: 5 m (16.4 ft)
- Combustor length: 0.2 m (0.656 ft)
- Nozzle/after-body length: 3 m (9.84 ft)
- Capture area: 0.584 m2 (6.29 ft2)
- Initial cruise contraction ratio: 10
- Adjustable cowl to maintain shock cancellation at combustor entrance
- Turning angle of upper body surface: 2.1 degrees
- Turning angle of fore-body: 6 degrees
- Nozzle exit area: 0.886 m2 (9.537 ft2)
- Constant area combustor following step at entrance of combustor
- Fuel injection orifice area: 0.0035 m2 (0.0377 ft2)
- Effective wing/lifting area: 36.48 m2 (392.7 ft2); vehicle maintains constant altitude by dynamically adjusting wing angle of attack

- 2D inlet four oblique shock system
- Quasi-one-dimensional flow in combustor and nozzle/afterbody
- Inlet friction coefficient: 0.001 (used in differential analysis; not averaged)
- Combustor friction coefficient: 0.02 (used in differential analysis; not averaged)
- Nozzle/afterbody friction coefficient: 0.001 (used in differential analysis; not averaged)
- Finite rate hydrogen-air reaction (7 species/7 reactions)
- Fixed input mixing schedule (flow is kinetics-limited at both throttle conditions)
- Variable specific heats (temperature-dependent) Gordon-McBride curve fits

5.1.3. Fuel Injection and Cooling

- Sonic flush-wall H2 injection: 45 degrees from horizontal/located at combustor entrance
- Mass flow rate of fuel: 0.31 kg/s (0.6834 lbm/s) (throttled down), 0.62 kg/s (1.37 lbm/s) (throttled up)
- Fuel equivalence ratios: 0.42 (throttled down), 0.84 (throttled up)
- Fuel (tank) total temperature: 120.3 K (216.5 °R)
- Fuel (tank) total absolute enthalpy: -255,1074 J/kg (-1,097 Btu/lbm)
- Fuel total pressure: 215,000 Pa (4,490 lbf/ft2) (throttled down), 363,000 Pa (7,581 lbf/ft2) (throttled up)
- Fuel used as heat sink (injection conditions vary with throttle condition)
- Heat load (rate): -2,458 kW (-2,330 Btu/s) (throttled down), -3,190 kW (-3,023.5 Btu/s) (throttled up)
- Fuel injection conditions (throttled down): 559 K (1,006 °R), 114 kPa (2,381 lbf/ft2)
- Fuel injection conditions (throttled up): 399 K (718.2 °R), 192 kPa (4,010 lbf/ft2)
- Combustor/Nozzle wall temperature: 1,000 K (1,800 °R)

5.2. Analysis Results

Tables 1 and 2, below, present representative cycle results for both throttle-down and throttle-up cases, respectively. Specifically, fundamental fluid dynamic variables are listed at five different engine stations corresponding to freestream (engine entrance), inlet exit, combustor entrance, combustor exit, and engine exit. Note that both the total pressure and the total temperature shown are actual total conditions corresponding to the local flow being slowed isentropically to Mach zero (with variable specific heats). Table 3 shows engine and component performance parameters for these cases; definitions of these parameters can be found in earlier cited references, including Ref. 4.

Table 1. Scramjet Modeling Example Flow Properties - Throttled Down Condition (Φ = 0.42).

Variable	Freestream o (0.0)	Inlet Exit iex	Combustor Entrance	Combustor Exit cex	Engine Exit e
Mach	7.92	4.03	3.63	2.14	4.23
Velocity m/s (ft/s)	2438.5 (8000)	2200 (7218)	2166.6 (7108.3)	1910.2 (6267)	2567.6 (8424)
Pressure N/m2 (lbf/ft2)	1195.8 (25)	43743.0 (913.6)	40400.0 (844)	108885.0 (2274)	3132.5 (65.4)
Temperature K (°R)	231.66 (417)	762.6 (1373)	784.5 (1412)	1982.8 (3569)	878.6 (1581)
Density kg/m3 (lbm/ft3)	0.0179 (0.00112)	0.199 (0.0138)	0.154 (.01066)	0.174 (.012)	0.0113 (.00078)
Area m2 (ft2)	0.584 (6.286)	0.0584 (.6286)	0.07732 (.8323)	0.07732 (.8323)	0.886 (9.54)
Total Temperature K (°R)	2734.0 (4921)	2734.0 (4921)	2463.0 (4433)	3212.0 (5782)	3195.0 (5751)
Total Pressure N/m2 (lbf/ft2)	17,600,000 (367583)	9157342 (191255)	4506358 (94117)	1057220 (22080)	962456 (20101)

Table 2. Scramjet Modeling Example Flow Properties - Throttled Up Condition (Φ = 0.84).

Variable	Freestream o	Inlet Exit iex	Combustor Entrance	Combustor Exit cex	Engine Exit e
Mach	7.92	4.03	3.44	1.55	3.707
Velocity m/s (ft/s)	2438.5 (8000)	2200.0 (7218)	2138.5 (7016)	1671.9 (5485)	2732.5 (8965)
Pressure N/m^2 (lb/ft^2)	1195.8 (24.97)	43743.0 (913.6)	44774.0 (935)	186479.0 (3895)	4500.7 (94)
Temperature K (°R)	231.66 (417)	762.6 (1373)	747.3 (1345)	2707.3 (4873)	1223.64 (2202)
Density kg/m^3 (lb/ft^3)	0.01791 (.00112)	0.199 (.0124)	0.158 (.009864)	0.202 (.01261)	0.0108 (0.000674)
Area m^2 (ft^2)	0.584 (6.286)	0.0584 (0.6286)	0.07732 (.8323)	0.07732 (.83232)	0.886 (9.537)
Total Temperature K (°R)	2734.0 (4921)	2734.0 (4921)	2217.72 (3992)	3525.0 (6345)	3350.2 (6030)
Total Pressure N/m^2 (lb/ft^2)	17,604,532 (367678)	9157342 (191255)	3637674 (75974)	699742 (14615)	655044 (13681)

Table 3. Scramjet Modeling Example Engine Performance Parameters.

Engine Parameter	Throttled Down	Throttled Up
Engine thrust N (lb)	6165.3 (1386)	12478.3 (2805)
Engine thrust (no irreversibility) N (lb)	11236.4 (2526)	20687.6 (4651)
Engine thrust (no irreversibility and complete combustion) N (lb)	13553.0 (3047)	25040.0 (5629)
Mass flow rate of fuel injected kg/s (lb_m/s)	0.31 (0.683)	0.62 (1.367)
Engine specific thrust Ns/kg (lb_fs/lb_m)	241.84 (24.66)	489.35 (49.9)
TSFC mg/Ns (lb_m/lb_fhr)	50.28 (1.775)	49.7 (1.75)
I_{sp} (s)	2027.0	2052.0
Inlet: $\Delta S_m/R$	0.654	0.654
Inlet: total pressure recovery	0.52	0.52
Inlet: kinetic energy efficiency	0.984	0.984
Inlet thrust potential effectiveness	0.96	0.98
Combustion efficiency (energy-based)	0.7914	0.774
Combustion efficiency (mass-flux based): only unburned H_2 counted as unreacted	0.867	0.864
Combustion efficiency (mass-flux based): all H other than in H_2O counted as “unreacted”	0.81	0.80
Mixer/combustor: total pressure recovery	0.1155	0.076
Mixer/combustor thrust potential effectiveness	0.595	0.629
Nozzle: velocity coefficient	0.965	0.955
Nozzle: total pressure recovery	0.91	0.936
Nozzle thrust potential effectiveness	0.96	0.98
Overall engine efficiency (thrust-based) Actual engine to reversible engine	0.55	0.603

6. Credibility of Simulations and Issues in Modeling High-Speed Propulsion Systems

The broad goal of numerical simulation is to provide performance and/or operability information that will facilitate the analysis, design, and optimization of a vehicle or vehicle component. The results obtained from a numerical simulation are credible only in terms of their ability to reproduce to acceptable levels the physical performance of an actual engineering system, model, or experiment. Note that this identification of a simulation effort with a component or a vehicle is

inclusive of extremely detailed unit-problem, high-order numerical simulations that are (for instance) aimed at developing understanding of detailed physical phenomena and may not necessarily define a specific engineering configuration at all. Ultimately, however, the information obtained from such a study should nevertheless in some way facilitate analysis, design, and optimization—that is, it should relate to performance and operability.

The credibility of simulations, as related to the performance parameters of interest for design, analysis, and optimization, therefore depends on what level of code/modeling technique is considered adequate to address a particular engineering problem. The highest level of simulation (level 5, as described in Fig. 5) would imply the use of unsteady DNS for fluids coupled with models for fluid-structural interactions, complex chemistry, and so on. However, that level of detail is generally impossible and is unnecessary for almost all meaningful analysis, design, and optimization efforts in the real world. For instance, level 0-1 codes cannot capture any details of wall heat transfer and moments for a complex 3D configuration, but they can still provide excellent results that are directly useful and acceptable in a MDA/MDO context or within a base parametric study, for example. For very low order methods, it may be acceptable to assess their credibility simply by requiring flow-rate (mass, stream-thrust, and total enthalpy) conservation. Discrepancies in performance results due to other important drivers (such as combustion efficiency distribution, distortion, etc.) that are simply input into such codes cannot be attributed to these methods, which do not have physical models for such phenomena.

It is unfortunate that results for multi-dimensional CFD are often considered credible only if spatial and temporal resolution of some specific result is shown (for example, a grid resolution study is performed showing convergence for a parameter of interest, and the L-2 normal of the residual is mentioned as being sufficiently small). However, there seems to be wide agreement within the community that at minimum, every multi-dimensional CFD solution—whether compared to available experimental data or not—should be accompanied by computations of its flow-rate conservation for mass, stream-thrust, total enthalpy, and entropy; balances of wall friction and boundary heat transfer; entropy increases due to irreversibility and transfer; and so on.

When using modeling and simulation tools to analyze a stand-alone engine, flow-path component, or sub-domain, it is necessary to formally assess the impact of that component or sub-domain on overall vehicle performance in order to ascertain its feasibility or performance. At minimum, this requirement mandates linking high-fidelity modeling of the component to (possibly) lower-fidelity assessment tools or techniques that can mimic the downstream flow-field effects on component performance. While widely acknowledged, however, this requirement is not generally met in modeling and simulation studies, even where it would be most relevant to the results presented.

The force-based performance of a high-speed vehicle can be shown to be solely driven by the generation of entropy, due to irreversibility—with all existing intermediate and component-level metrics and measures, at best, simply providing partial and necessarily incomplete assessments. It would therefore be useful if more explicit attention were given to these second-law issues in both model development and applications. The use of the common currency of entropy generation should also be of particular interest in multi-disciplinary modeling and simulation and in vehicle/mission optimization efforts. Entropy generation is the most fundamental and universal measure of losses, and it applies to every subsystem and process on a vehicle—mechanical, chemical, fluid-dynamic, etc. The fact that entropy generation is directly (explicitly) relatable to classic vehicle performance should make it of paramount (and explicit) importance in all levels of modeling and simulation. For recent developments in this area, see Refs. 21 and 22.

As briefly noted in this chapter, there are many challenging physical performance and operability issues that must be considered in the detailed design of high-speed air-breathing propulsion systems. These include, but are not limited to: transition; fluid-structural interactions; thermal protection; ignition; piloting and flame-holding; isolator (mid-speed) combustor dynamics and interactions including unstart and start; fuel properties and fuel system characteristics; fuel-air mixing and combustion; pitch control; and engine-airframe integration. These issues are directly reflected in the challenges facing the various computational and analytical levels and models used or developed for this class of engine.

However, there is a fundamental limit to developing meaningful modeling capability from an engineering standpoint; this limit should inform the community to some extent as to whether it is useful to build and use ever-better models as prediction tools. Specifically, when the inherent aleatory uncertainty in the details of 1) an actual flight condition and 2) the actual physical vehicle configuration exceeds the modeling uncertainty at a given level of fidelity (such as for a deformed geometry), it is not productive to develop or use higher fidelity models with their attendant expense. It must be noted, however, that regardless of the uncertainty in flight conditions and geometry, modeling can sometimes yield important information on the sensitivity to performance from the uncertainties in different parameters. Nevertheless, it is possible that this limit may already have been reached more frequently than now realized or acknowledged; for instance, the local triggers for transition or inlet unstart may sometimes already be imbedded within the aleatory uncertainties inherent in flying a hypersonic vehicle through the atmosphere. If so, no amount of improved modeling could facilitate truly improved predictions. This, then, mandates increased attention and effort on developing robust uncertainty quantification methodologies and providing uncertainty analyses for high-speed flight systems; the role of uncertainty quantification informs the community where and when to expend effort in developing improved modeling and simulation tools, and what level of tools are necessary or practical for a given study.

References

1. Curran, E. T., *High-Speed Flight Propulsion Systems, Progress in Astronautics and Aeronautics Series,* Volume 137, American Institute of Astronautics and Aeronautics, Inc., Edited by S.N.B Murthy and E.T. Curran, Washington, D.C., 1991
2. Heiser, W. H., and Pratt, D. T., with Daley, D. H., and Mehta, U. B., *Hypersonic Airbreathing Propulsion*, American Institute of Aeronautics and Astronautics, Inc., 1994.
3. Curran, E. T. (Editor), *Scramjet Propulsion, Progress in Astronautics and Aeronautics Series*, Volume 189, American Institute of Aeronautics and Astronautics, Edited by S. N. B. Murthy and E. T. Curran, Washington, D.C., 2001.
4. Riggins, D. W., "The Analysis of Scramjet Engines," *Scramjet Propulsion Testing Standards, Recommended Practices, and Guidelines*, CPIA Publication 710, Vol. 1, 2005.
5. McClinton, C. R., "X-43 - Scramjet Power Breaks the Hypersonic Barrier: Dryden Lectureship in Research for 2006," 44th AIAA Aerospace Sciences Meeting and Exhibit, Aerospace Sciences Meetings, 2006.
6. Warwick, G., "First X-51A Hypersonic Flight Deemed Success," Aviation Week, May 26, 2010.
7. Bowcutt, K., "Multidisciplinary Optimization of Airbreathing Hypersonic Vehicles," *Journal of Propulsion and Power*, Vol. 17, No. 6, pp. 1184-1190, 2001.

8. Curran, E. T., and Stull, F. D., “The Utilization of Supersonic Combustion Ramjet Systems at Low Mach Numbers,” Aero Propulsion Laboratory Report RTD-TDR-63-4097, Jan. 1964.

9. Milligan, R. T., Eklund, D. R., Wolff, J. M., Gruber, M., and Mather, T., “Dual-Mode Scramjet Combustor: Numerical Analysis of Two Flowpaths,” J*ournal of Propulsion and Power*, Vol. 27, No. 6, pp. 1317-1320, 2011.

10. Baurle, R., and Eklund, D., “Analysis of Dual-Mode Hydrocarbon Scramjet Operation at Mach 4-6.5,” *Journal of Propulsion and Power*, Vol. 18, pp. 990-1002, 2002.

11. Riley, L., Hagenmaier, M., Donbar, J., Gaitonde, D., “A Computational Investigation of Unstart in a Dual-Mode Scramjet,” 54th AIAA Aerospace Sciences Meeting, AIAA SciTech, paper AIAA 2016-1901, 2016.

12. Riley, L., Gaitonde, D., Donbar, J., “Preliminary Investigation of Unstart-Related Transients in a Dual- Mode Scramjet,” 51st AIAA/SAE/ASEE Joint Propulsion Conference, Propulsion and Energy Forum, paper AIAA 2015-4206, 2015.

13. Riggins, D., Tackett, R., Taylor, T., and Auslender, A., “Thermodynamic Analysis of Dual-Mode Scramjet Engine Operation and Performance,” 14th AIAA/AHI Space Planes and Hypersonic Systems and Technologies Conference, paper AIAA 2006-8059, 2009.

14. Baurle, R., “Modeling of High-Speed Reacting Flows: Established Practices and Future Challenges,” 42nd AIAA Aerospace Sciences Meeting and Exhibit, paper AIAA 2004-0267, 2004.

15. Drummond, J. P., Carpenter, M. H., and Riggins, D. W., “Mixing and Mixing Enhancement in Supersonic Reacting Flowfields,” *Progress in Astronautics and Aeronautics Series; High-Speed Flight Propulsion Systems*, Vol. 137, pp. 383-455, 1991.

16. Eklund, D. R., Stouffer, S. D., and Northam, G. B., "Study of a Supersonic Combustor Employing Swept Ramp Fuel Injectors," *Journal of Propulsion and Power*, Vol. 13, No. 6, pp. 697-704, 1997.

17. Eklund, D. R., Fletcher, D. G., Hartfield, R. J., McDaniel, J. C., Northam, G. B., Dancey, C. L., and Wang, J. A., "Computational/experimental investigation of staged injection into a Mach 2 flow," *AIAA Journal*, Vol. 32, No. 5, pp. 907-916, 1994.

18. Donohue, J. M., McDaniel, J. C., and Haj-Hariri, H., "Experimental and numerical study of swept ramp injection into a supersonic flowfield," *AIAA Journal*, Vol. 32, No. 9, pp. 1860-1867, 1994.

19. Auslender, A. H., “An Application of Distortion Analysis to Scramjet-Combustor Performance Assessment,” Scramjet Engine Performance Analysis, Evaluation, and Optimization – JANNAF Propulsion and Joint Subcommittee Scramjet Performance Workshop, Dec. 1996.

20. Baurle, R., and Gaffney, R., "Extraction of One-Dimensional Flow Properties from Multidimensional Data Sets," *Journal of Propulsion and Power*, Vol. 24, No. 4, pp. 704-714, 2008.

21. Riggins, D., Taylor, T., and Moorhouse, D., "Methodology for Performance Analysis of Aerospace Vehicles Using the Laws of Thermodynamics," *AIAA Journal of Aircraft*, Vol. 43, No. 4, pp. 953-963, 2006.

22. Riggins, D. W., Camberos, J., Wolff, M., Bowcutt, K., “Mission-Integrated Exergy Analysis for Hypersonic Vehicles: Methodology and Application,” *AIAA Journal of Propulsion and Power*, Vol. 29, No. 3, pp. 610-620, May-June 2013.

Computational Modeling of Liquid Propulsion Systems

Venkateswaran Sankaran[⋆], **Sophonias Teshome**[†1] and **Changjin Yoon**[‡2]

[⋆]*Air Force Research Laboratory, Edwards AFB, CA 93524*
[†]*ERC Inc., Air Force Research Laboratory, Edwards AFB, CA 93524*
[‡]*Purdue University, West Lafayette, IN, 47907*

1. Introduction

High-fidelity physics-based modeling of liquid propulsion systems is becoming increasingly important because of the potential for significantly reducing development and testing costs. We provide a brief survey of liquid propulsion modeling in rockets, gas turbines, and scramjets. Because of the broad scope of these topics, it is not possible to provide a comprehensive summary of all the modeling approaches. Rather, we focus on the dominant physical phenomena in these devices such as injection and mixing, heat transfer, and combustion/ignition dynamics, and discuss the pertinent simulation objectives. We then provide a summary of physics-based modeling approaches, including turbulence, turbulent combustion, combustion kinetics, and multiphase phenomena. Finally, we discuss possible sources of uncertainty in the physical models and the associated numerical procedures.

2. Overview of Liquid Propulsion Physics

Liquid propellant-based combustion systems are used in a number of propulsion devices including rocket-engines, gas turbines, augmentors, and scramjet engines. The underlying physical phenomena in these devices are closely related although their respective operating regimes and combustor designs are very different. Figure 1 (top) shows the schematic of a typical liquid rocket combustor—propellants are injected at the upstream end of the combustor usually in the supercritical state, although multi-phase phenomena may be present under certain operating conditions [1]. The propellants mix and burn and the products of combustion expand in a converging-diverging nozzle to generate thrust. The walls of the combustor may be cooled by using a wall film on the hot-combustion side and/or by circulating cold fuel through regenerative cooling passages. Unlike hydrogen fuels, hydrocarbon fuels can give rise to soot formation and deposition on the combustor or coolant passage walls, which in turn influence the heat transfer rates. Because the combustion occurs within a closed chamber environment, acoustics phenomena can also play a significant role and, under some circumstances, may couple with the combustion processes leading to rough burning or combustion

[1] At Aerospace Corporation, Los Angeles, CA at time of publication
[2] At GE Global Research, Schnectady, NY at time of publication

instabilities. Moreover, combustion dynamics related to ignition and extinction transients are also key phenomena that must be accounted for in these devices.

A schematic of a representative gas turbine combustor is also shown in Figure 1 (bottom). Here, the fuel is introduced in the form of a spray with co-flowing swirling air that promotes atomization and rapid mixing. Additional dilution air is introduced downstream for cooling and for controlling emissions. As in the rocket combustor case, the flowfield is characterized by multi-phase turbulent reacting flows, coupled with dynamics related to acoustic interactions, hydrodynamic stabilities, multiphase effects and particulate (soot) formation [2]. In both of these devices as well as in scramjets, multi-physics and multi-scale coupling can play an important role. Multi-physics refers to the coupling of fluid dynamics with combustion, material properties and structures, while multi-scale refers to the coupling of physical phenomena occurring at the large fluid-dynamic scales with those occurring at the smallest scales of turbulence, combustion, droplet sprays and, perhaps, even at the molecular level of the structural materials.

3. Simulation Objectives

The ultimate objective of high-fidelity modeling of combustion devices is to develop a "virtual" engine, which can be used to carry out parametric studies in support of design and/or analysis prior to full-scale physical testing. However, given the present state-of-the-art in combustion modeling and the limitations of even the largest available computer clusters, most of the existing models are in practice used to address specific aspects of an engine design rather than to simulate all of the details. Thus, we categorize the simulation objectives into three main analysis areas: injection/mixing, thermal management, and combustion dynamics. In each case, we further sub-divide the simulations into two levels: steady or unsteady Reynolds-averaged Navier-Stokes (RANS) models [3] and large eddy simulation (LES) [4] or detached eddy simulation (DES) models [5]. In fact, engineering design analysis often uses lower-fidelity models, but we do not consider these here since they are varied in form and capability and typically require input from experimental or test data or from high-fidelity models. For specificity, the information presented here pertains primarily to liquid rocket simulations, although there is significant overlap with the requirements for gas turbines and scramjets as well. Additional areas of significance for gas turbines include combustor emissions, while for scramjets, ignition, and flame-stability are additional concerns.

Table 1 provides the typical set of simulation objectives for injection and mixing when using RANS and LES models. The broad objective is the design of the injector element in order to permit appropriate mixing and combustion of the propellants. We focus specifically on mixing and combustion here, while attendant phenomena related to injector face-plate heat transfer and combustion dynamics and instabilities are considered later. The fundamental design parameters include the estimation of the intact liquid core length, combustion length, and pressure-drop across the injector element. Note that liquid rocket engines typically operate at supercritical pressures, and the liquid core length may simply refer to the high-density core; on the other hand, in gas turbine engines, multi-phase phenomena such as primary and secondary atomization are critical for determining injector characteristics [6]. Sophisticated RANS models can provide a detailed prediction of these atomization processes as well as distributions of the droplet diameter. Such models are also useful for predicting the effects of inter-element interactions, which are important in practical devices that can have hundreds of these elements packed closely together. It should be pointed out that RANS models can be applied to capture unsteady effects, but their utility is restricted to situations where the unsteady scales are

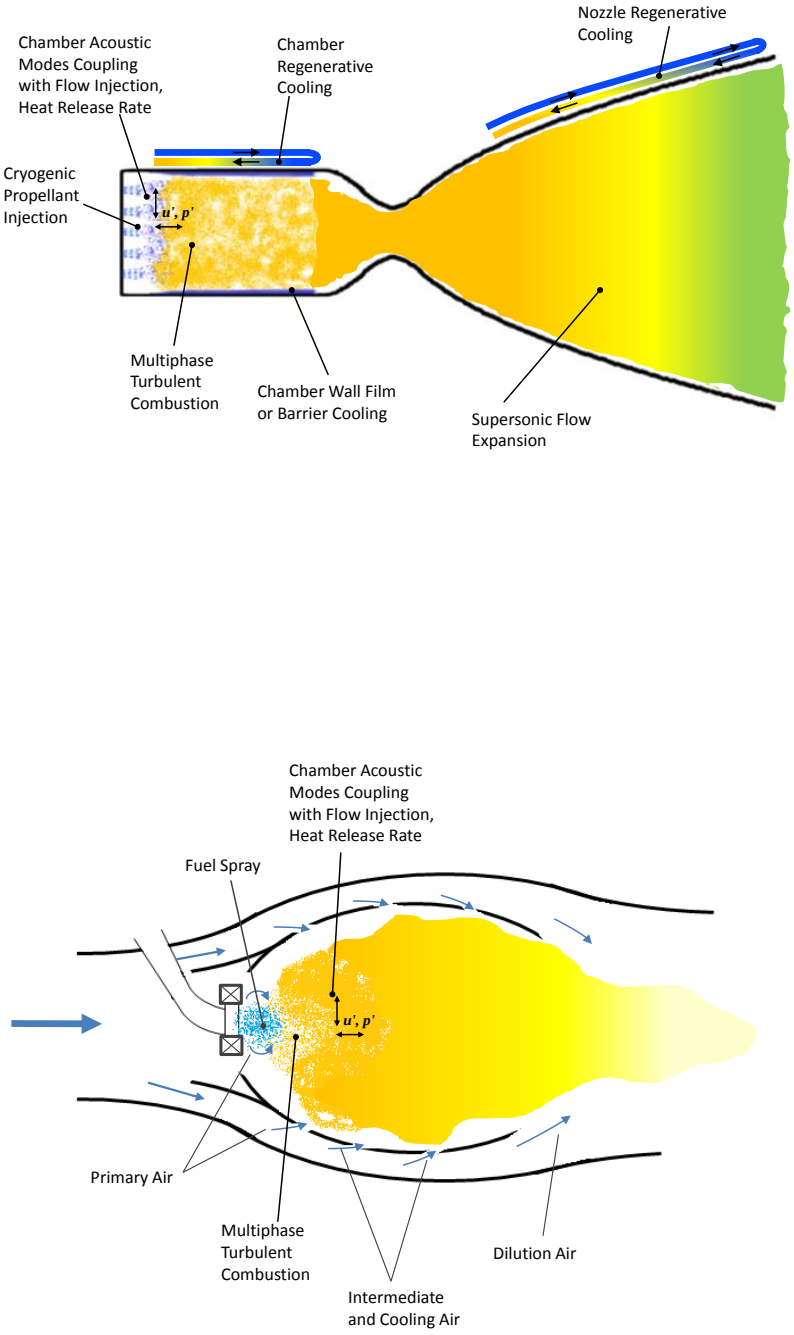


Figure 1. Schematic of a liquid rocket engine (top) and gas turbine engine (bottom).

Table 1. Injection/Mixing Objectives

RANS	LES/DES
Intact liquid length	Intact liquid length
Combustion length	Combustion length
Pressure drop across injector	Pressure drop across injector
Mean droplet distribution	Unsteady droplet distribution
Multi-element interaction	Unsteady multi-element interaction
	Hydrodynamic instabilities
	Injector admittances
	Ignition, extinction, flame blowout

Table 2. Heat Transfer Objectives

RANS	LES/DES
Wall heat transfer	Wall heat transfer
Soot deposition	Soot deposition
Radiation heat transfer	Radiation heat transfer
Wall roughness effects	Wall roughness effects
Conjugate heat transfer	Conjugate heat transfer
	Injector faceplate heat transfer
	Wall streaking and hot spots

much larger than the turbulent time scales. LES and DES models, on the other hand, provide a more detailed representation of the time-dependent aspects of hydrodynamic instability mechanisms, injection dynamics, and ignition/extinction phenomena.

Table 2 lists simulation objectives related to heat transfer in engine combustors. RANS models are typically used to determine wall heat transfer coefficients both on the combustor-side and for the regenerative coolant passage walls. Moreover, RANS calculations can account for finite-rate chemistry effects including soot generation and deposition, radiation heat transfer and the effects of wall roughness and conjugate heat transfer. In addition, LES computations provide dynamical effects that can predict hot spots and streaking on the injector face and combustor walls.

Table 3 lists the simulation objectives for combustion dynamics, which is relevant to the prediction of combustion instabilities as well as ignition phenomena. Combustion instabilities result from the coupling of the combustion heat release with the acoustic modes of the combustion chamber [7]. Engineering models are typically utilized to calculate the acoustic mode frequencies and amplitudes of the unstable modes. The models are based on the acoustic wave equation and require the specification of the mean flow properties and combustion response functions that may be obtained from the more detailed RANS and LES models. Indeed, RANS calculations are generally not suitable for capturing the combustion dynamics, but they can be useful for providing mean properties as inputs to engineering stability codes. LES computations, on the other hand, can in principle represent all of the dynamics interactions between combustion and the acoustics, although these are challenging computations because they require the inclusion of detailed chemical kinetics and turbulent combustion models.

Table 3. Combustion Dynamics Objectives

RANS	LES/DES
Mean temperature & sound speed	Unsteady temperature/velocity fields
Mean combustion heat release	Unsteady heat transfer
	Mode frequency & amplitude
	Spatial & temporal response
	Injector, cavity, nozzle admittances
	Acoustics-heat release phase correlations
	Ignition dynamics

We note that such high-fidelity simulations can be computationally intensive and are usually impractical for use within a design environment. Consequently, variable-fidelity approaches are being proposed. Ideally, the high-fidelity simulations are utilized to train or anchor the lower-fidelity models, while the lower-fidelity or reduced-order models are utilized for rapid turnaround of predictions during the design iterations. The inclusion of uncertainty quantification within such a variable-fidelity design infrastructure is of key importance as discussed later. In the following sections, we restrict our attention to high-fidelity physical modeling approaches of relevance to LES, of which RANS models may be viewed as a subset. We do not consider reduced-order engineering models here since they have considerable variability depending on the application. Moreover, current state-of-the-art engineering models are often empirically derived and do not perform well outside of their specific domain of applicability. However, we foresee an emerging trend toward a new class of reduced models that are mathematically derived from their high-fidelity counterparts. Consequently, such models can automatically provide uncertainty data as part of the model derivation procedure and are therefore likely to play a significant role in design technologies of the future.

4. Physical Modeling Approaches

4.1. Equations of Motion

The governing equations for compressible reacting flows are generally expressed in the Favre-averaged (for RANS) or Favre-filtered (for LES) form. Favre-averaging or filtering refers to density-based averaging (or filtering) of the velocity field which leads to a simpler form of the governing equations that does not introduce new closure terms in the continuity and momentum equations as compared with the incompressible form of the equations. Favre-averaging is defined as:

$$\tilde{u} = \frac{\overline{\rho u}}{\overline{\rho}} \tag{1}$$

where the over-bars refer to conventional Reynolds-averaging or LES-filtering of the density and density-velocity product.

The final form of the Favre-averaged or filtered equations are:

$$\frac{\partial \overline{\rho}}{\partial t} + \frac{\partial}{\partial x_i}(\overline{\rho}\widetilde{u}_i) = 0 \tag{2}$$

$$\frac{\partial}{\partial t}(\overline{\rho}\widetilde{u}_i) + \frac{\partial}{\partial x_j}(\overline{\rho}\widetilde{u}_j\widetilde{u}_i) = -\frac{\partial \overline{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\overline{\tau}_{ji} - \overline{\rho}\left(\widetilde{u_j u_i} - \widetilde{u}_j\widetilde{u}_i\right)\right] \tag{3}$$

$$\frac{\partial}{\partial t}\left(\overline{\rho}\widetilde{h}_0\right) + \frac{\partial}{\partial x_j}\left(\overline{\rho}\widetilde{u}_j\widetilde{h}_0\right) = \frac{\partial \overline{p}}{\partial t} + \frac{\partial}{\partial x_j}\left(\widetilde{u_i\tau_{ij}} - \overline{q}_i - \overline{\rho}\left(\widetilde{u_j h_0} - \widetilde{u}_j\widetilde{h}_0\right) + \left(\widetilde{u_i\tau_{ij}} - \widetilde{u}_i\overline{\tau_{ij}}\right)\right) \tag{4}$$

$$\frac{\partial}{\partial t}(\overline{\rho}\widetilde{Y}_l) + \frac{\partial}{\partial x_j}\left(\overline{\rho}\widetilde{u}_j\widetilde{Y}_l\right) = \overline{\dot{\omega}}_l + \frac{\partial}{\partial x_j}\left[-\overline{\rho}\widetilde{V}_{l,j}\widetilde{Y}_l + \overline{\rho}\left(\widetilde{u_j Y_l} - \widetilde{u}_j\widetilde{Y}_l\right)\right] \tag{5}$$

which are the continuity (2), momentum (3), energy (4) and species (5) transport equations, respectively [8]. The terms requiring closure are the so-called Reynolds stresses, which are the last two terms in the momentum, energy and species transport equations, and the turbulent combustion source term, which is the first term on the right-hand side of the species transport equation. Depending on whether the equations are averaged or filtered, the Reynolds stress terms take distinct forms. In the case of RANS, the stress terms are given by:

$$\overline{\rho}\left(\widetilde{u_i u_i} - \widetilde{u}_j\widetilde{u}_i\right) = \overline{\rho}\widetilde{u_j'' u_i''} \tag{6}$$

On the other-hand, for LES, the stress terms may be expanded as:

$$\overline{\rho}\left(\widetilde{u_i u_i} - \widetilde{u}_j\widetilde{u}_i\right) = \tau_{ij}^{sgs} = \underbrace{\overline{\rho}\left(\widetilde{\widetilde{u}_j\widetilde{u}_i} - \widetilde{u}_j\widetilde{u}_i\right)}_{\text{Leonard stress}} + \underbrace{\overline{\rho}\left(\widetilde{\widetilde{u}_j u_i''} + \widetilde{u_j''\widetilde{u}_i}\right)}_{\text{Cross-term stress}} + \underbrace{\overline{\rho}\widetilde{u_j'' u_i''}}_{\text{Reynolds stress}} \tag{7}$$

Similar expressions hold for the stresses in the energy and species transport equations. Although the stress terms may be written individually, they are typically modeled together as a single entity in most LES models and, for this reason, we refer to the full set as the Reynolds stresses. Specific models for the Reynolds stresses and the turbulent combustion source term are discussed later. We now consider the form of generalized equations of state for closing the above system.

4.2. Generalized Equations of State

The governing equations must be augmented by appropriate equations of state for the constituent species that relate the density and enthalpy to the pressure and temperature. For high-pressure propulsion applications, the ideal gas state relations are typically inadequate, and so appropriate real gas equations of state must be utilized [9]. Specifically, for liquid propulsion applications, the fluid may behave like a liquid in certain regimes and like a gas in other regimes. Different real-gas equations of state may be used, such as Peng-Robinson [10], Soave-Redlich-Kwong [11] and the Bennedict-Webb-Rubin [12] equations of state applicable to a wide range of pressures and temperatures.

For convenience, we express the equations of motion in coupled vector form:

$$\frac{\partial Q}{\partial t} + \nabla \cdot \left(\vec{\mathscr{F}} - \vec{\mathscr{F}}_v\right) = H \tag{8}$$

where

$$Q = \left(\overline{\rho},\ \overline{\rho}\tilde{u}_j,\ \tilde{e},\ \overline{\rho}\tilde{Y}_l\right)^T \qquad \vec{\mathscr{F}} = \sum E_i e_i \qquad \vec{\mathscr{F}}_v = \sum E_{v,i} e_i \qquad H = (0,\ 0,\ 0,\ \overline{\dot{w}_l})^T \tag{9}$$

and where E_i and $E_{v,i}$ are the appropriate inviscid and viscous flux vectors and H is the source term vector. The generalized equations of state may typically be expressed in the following form:

$$\overline{\rho}_l = \overline{\rho}_l(\overline{p},\ \tilde{T}) \qquad \tilde{h}_l = \tilde{h}_l(\overline{p},\ \tilde{T}) \tag{10}$$

The mixture density is obtained by applying Amagat's mixture rule:

$$\overline{\rho} = \left(\sum_1^N \frac{\tilde{Y}_l}{\overline{\rho}_l}\right)^{-1} \qquad \tilde{h} = \sum_1^N \tilde{h}_l \tilde{Y}_l \tag{11}$$

The importance of using Amagat's rule is that it may be applied both for mixtures of gases and for mixtures of gases and liquids (using the homogeneous mixture assumption). At the high pressures representative of rocket conditions, this is an important consideration since the propellant could be at either state in the manifold, coolant channels and the combustion chamber. Other mixing rules are available, but are not considered here.

For the generalized system of equations, the acoustic sound speed is given by:

$$\bar{c}^2 = \frac{1}{(\partial\overline{\rho}/\partial\overline{p})_s} = \frac{\overline{\rho}\overline{h}_T}{\overline{\rho}\ \overline{\rho}_p\overline{h}_T + \overline{\rho}_T(1-\overline{\rho}\overline{h}_p)} \tag{12}$$

Since the inviscid eigenvalues of the system are determined by the fluid velocities and the acoustic speeds, the relative magnitudes of these scales determines the inherent stiffness of the problem. When these scales are very disparate, traditional computational algorithms may degrade in both accuracy and efficiency and it then becomes necessary to include specialized preconditioning algorithms. The interested reader is referred to the literature for details [13, 14].

4.3. Combustion Kinetics

Detailed modeling of combustion requires the specification of chemical kinetics mechanisms that describe the production or destruction of particular species—in other words, it requires the specification of the source term in the species transport equation given above. Before we discuss the turbulence closure of this term, we summarize its basic definition for laminar flows. In fundamental form, the chemical reactions are specified by a set of elementary chemical mechanisms of the following form:

$$\nu'_{j,k}\chi_j \rightleftharpoons \nu''_{j,k}\chi_j \tag{13}$$

where χ_j is the jth species participating in the reaction, and ν'_{jk} and ν''_{jk} are the stoichiometric coefficients for the kth reaction.

The rates of the forward and backward reactions are given by:

$$K_f = A_f T^{\alpha_f} \exp\left(\frac{-E_{af}}{R_u T}\right) \tag{14}$$

$$K_b = A_b T^{\alpha_b} \exp\left(\frac{-E_{ab}}{R_u T}\right) \tag{15}$$

where A_f, A_b, α_f and α_b are constants, T is the temperature, E_{af} and E_{ab} are the forward and backward activation energies, and R_u is the universal gas constant. Additional modifications are used to accommodate third-body reactions and for introducing pressure dependencies (which are important for high-pressure rocket applications) [15, 16].

The net species source term is expressed as:

$$\dot{w}_j = M_j \sum_k (\nu''_{jk} - \nu'_{jk}) \cdot [NRR]_k \tag{16}$$

where $[NRR]_k$ represents the net reaction rate of the kth reaction and is given by:

$$NRR = K_f \Pi_i \chi_i^{\nu'_{jk}} - K_b \Pi_i \chi_i^{\nu''_{jk}} \tag{17}$$

where K_f and K_b are the forward and backward reaction rates that were defined earlier.

4.4. Turbulence Modeling

In this section, we discuss sub-grid models for the Reynolds stress terms defined in Section 4.1. As noted, two approaches have been commonly used, namely, RANS [3] and LES [4]. For specificity, we focus on LES models here, while RANS models are presented later within the class of hybrid RANS/LES or DES models [5]. In LES models, the deviatoric residual stress tensor is given by:

$$\tau_{ij}^{d,sgs} \equiv \tau_{ij}^{sgs} - \frac{1}{3}\tau_{ii}^{sgs}\delta_{ij} \tag{18}$$

and is modeled as:

$$\tau_{ij}^{d,sgs} = -2\nu_t \overline{S}_{ij} \tag{19}$$

where the over-bar signifies application of a filter.

In the constant Smagorinsky model, the turbulent eddy viscosity, ν_t, is modeled as:

$$\nu_t = l_s \left(2\overline{S}_{ij}\overline{S}_{ij}\right)^{1/2} \tag{20}$$

$$= (C_s \Delta_g)^2 \left(2\overline{S}_{ij}\overline{S}_{ij}\right)^{1/2} \tag{21}$$

where l_s, known as the Smagorinsky lengthscale, is proportional to the filter or grid width Δ_g with a proportionality constant C_s, the Smagorinsky coefficient. Its value is dependent on the flow conditions and takes on values in the range $0.10 < C_s < 0.24$ [17].

In the dynamic Smagorinsky model, Germano *et al.* [18] proposed a method whereby C_s is determined as the calculation progresses instead of specifying it *a priori*. The associated eddy viscosity is given by:

$$\nu_t = C(x,t)\Delta_g^2 \left|\overline{\mathscr{S}}\right| \tag{22}$$

where $C(x,t)$ is the dynamic Smagorinsky coefficient and $\overline{\mathscr{S}} = \left(2\overline{S}_{ij}\overline{S}_{ij}\right)^{1/2}$. A procedure for solving for C proposed by Lilly [19] involves the application of a test filter of width Δ_t with a scale larger

than the grid filter width, Δ_g, that is, $\Delta_t = a\Delta_g$ and $a > 1$. Applying a test filter to the Favre-filtered momentum equation generates a residual stress tensor T^{sgs} so that:

$$\text{Test filter}: \quad T_{ij}^{d,sgs} \equiv T_{ij}^{sgs} - \frac{1}{3}T_{ii}^{sgs}\delta_{ij} = -2C\Delta_t^2 \left|\widehat{\overline{\mathscr{S}}}\right| \widehat{\overline{S}}_{ij} \tag{23}$$

$$\text{Grid filter}: \quad \tau_{ij}^{d,sgs} \equiv \tau_{ij}^{sgs} - \frac{1}{3}\tau_{ii}^{sgs}\delta_{ij} = -2C\Delta_g^2 \left|\overline{\mathscr{S}}\right| \overline{S}_{ij} \tag{24}$$

It is possible to solve for a single value of the coefficient C at the two scale ranges in a least squares sense. The dynamic model thus incorporates local instantaneous information about the strain-rate field at the smallest resolved scales to determine a value of C that varies continuously in space and time throughout the simulation.

The isotropic part of the stress tensor is modeled as:

$$\tau_{ij}^{I,sgs} = \frac{1}{3}\tau_{ii}^{sgs}\delta_{ij} = 2C_I\Delta_g^2|\overline{S}|^2 \tag{25}$$

where C_I is a dimensionless coefficient [8]. Similar expressions may be written for the sub-grid-scale energy and species fluxes using the standard gradient diffusion hypothesis.

Other models for the sub-grid stress are possible including one- or two-equation models such as the k and $k-l$ models. In addition, hybrid RANS/LES or DES [5] models based on the $k-\varepsilon$ and $k-\omega$ models have emerged as a popular choice for engineering simulations. Such models solve the RANS equations in the near-wall region while approaching the LES solution in regions away from the walls. The approach is particularly useful for liquid combustion systems since propellant mixing and reactions typically occur in the off-body region and the use of LES in this region improves the accuracy of the associated combustion dynamics. Here, we show only the k or turbulence kinetic energy equation corresponding to the $k-\omega$ model [3] since the relevant modifications to the DES model involve only this equation:

$$\frac{\partial}{\partial t}(\overline{\rho}k) + \frac{\partial}{\partial x_j}(\overline{\rho}\widetilde{u}_j k) = \overline{\rho}\,\overline{\tau}_{ij}\frac{\partial \widetilde{u}_i}{\partial x_j} - \beta^*\overline{\rho}k\omega + \frac{\partial}{\partial x_j}\left[\left(\mu + \sigma^*\frac{\overline{\rho}k}{\omega}\right)\frac{\partial k}{\partial x_j}\right] \tag{26}$$

The DES modification replaces the dissipation term in the above equation, $\beta^*\overline{\rho}k\omega$, as follows [20]:

$$D_k = \frac{\overline{\rho}k^{3/2}}{l_{des}} \tag{27}$$

where

$$l_{des} = \min(l_{k-\omega}, c_{des}\Delta) \tag{28}$$

$l_{k-\omega}$ is the standard turbulent length scale that is given by:

$$l_{k-\omega} = \frac{k^{1/2}}{\beta^*\omega} \tag{29}$$

and $\Delta = \max(\Delta x,\ \Delta y,\ \Delta z)$.

It is evident that in regions where the maximum grid sizing is smaller than the turbulent length scale, the traditional length scale definition is replaced by a length scale specified by the grid-sizing. The net effect of this DES modification in well resolved sections of the flow is to increase the dissipation of kinetic energy based on the cell-size, thereby leading to the destruction of the RANS-portion of the model and the switching on of the LES model.

Table 4. Turbulent Combustion Models

Model	Key Assumptions	Procedure	Validity
Flamelets	1D, steady, laminar Incompressible Constant diffusion coefficent Uses presumed-PDF	Solves for Z, Z" Solves reaction progress variable Tabulate reactive scalar solutions Derive filtered quantities	High Da Low Mach Low Re
G-Equation	For premixed combustion Flamelet assumptions Tracks flamefront	Solves for Z, Z" Uses level-set for G-eqn Requires flamelet tabulations	High Da Low Mach Low Re
LEM	Sub-grid transport model 1D, constant pressure in sub-grid Exact combustion closure exact	Species convection in LES grid 1D Reaction-diffusion in LEM grid LEM solution is explicit	All regimes
PDF-Transport	Scalar mixing model Exact combustion closure Joint-PDF simiplifications	Solve Langevin equation Solution process can be expensive	Low Da Low Mach High Re

4.5. Turbulent Combustion

A number of different approaches have also been proposed for the closure of the turbulent combustion source term, which was defined earlier for the laminar case [21, 22, 23]. Table 4 lists the main classes of models, along with the key assumptions and the solution process in each case. Some models such as the linear eddy model (LEM) solve the full set of transport equations—continuity, momentum, energy and species—and only sub-grid terms are modeled. On the other hand, models such as flamelets and transported probability density function (PDF) or the related filtered mass density function (FMDF) models involve additional assumptions, such as the modeling of the combustion kinetics (in flamelets) or scalar mixing (in PDF or FMDF methods).

The so-called flamelet approximation involves the assumption that non-equilibrium effects are restricted to the immediate vicinity of the flame front, which is justifiable when the timescales of the chemistry mechanisms are smaller than the smallest turbulent scales [21]. Flamelet models involve the use of a conserved scalar equation, which represents the mixing between the fuel and oxidizer streams instead of a large number of independent reactive scalar equations. The Favre-filtered form of the conserved scalar equation is given by:

$$\frac{\partial}{\partial t}(\overline{\rho}\widetilde{Z}) + \frac{\partial}{\partial x_j}\left(\overline{\rho}\widetilde{u}_j\widetilde{Z}\right) = \frac{\partial}{\partial x_j}\left[\overline{\rho D}\frac{\partial}{\partial x_i}\widetilde{Z} + \overline{\rho}\left(\widetilde{u_j Z} - \widetilde{u}_j\widetilde{Z}\right)\right] \tag{30}$$

Note that the conserved scalar equation contains no source terms. In the steady flamelet model, the reactive scalars are parametrized in terms of the conserved scalar function and the scalar dissipation rate, χ, and the associated flamelet equations are written in one-dimensional form as:

$$\frac{\rho}{2}\chi\frac{\partial^2\psi_i}{\partial Z^2} + w_i = 0 \tag{31}$$

where Z is the conserved scalar variable and ψ_i is the reactive scalar variable, which represents the vector $(Y_i, T)^T$. The key approximations in the derivation and solution of the flamelet equations are: (1) the pressure in the reactive scalar system is approximated as a constant (low Mach number assumption), (2) the velocity field is specified from a simplified problem (such as counter-flow diffusion flames) rather than from the actual flow-field solution, and (3) the entire set of species

equations is solved in conjunction with an approximate form of the energy equation. The representative solution is obtained analytically or numerically for a model problem (such as counter-flow diffusion), parameterized by the mixture fraction and the scalar dissipation rate, and tabulated for efficient look-up:

$$\psi_i = \psi_i(Z, \chi_{st}) \tag{32}$$

This assumption of laminar flamelets is justified in the small Karlovitz (Ka) number limit, i.e., when the chemistry scales are much smaller than the Kolmogorov scales. Alternately, this limit may be expressed in terms of the Damkohler (Da) number, which is the ratio of the turbulent time scale to the chemical time scale, as shown in Table 4. We point out that these conditions are more likely to be encountered at relatively low Reynolds numbers (Re).

The Favre-averaged (or filtered) reactive scalars, which are necessary for calculating the mixture density, can be expressed using a joint-presumed probability density function (PDF) for Z and χ_{st}:

$$\tilde{\psi}_i(x,t) = \int_0^1 \int_0^\infty \psi_i(Z,t,\chi_{st}) \tilde{P}(Z,\chi_{st};x,t) d\chi_{st} dZ \tag{33}$$

The presumed PDF is usually expressed by assuming statistical independence of Z and χ_{st} and the PDF of Z is taken to be the beta-PDF [21], which also requires specification of the variance of the conserved scalar function or Z''. The latter quantity is obtained through the solution of an appropriate transport for the variance. Additional details are provided by Peters [21].

Efforts to extend the validity of the model to represent dynamic phenomena such as ignition and extinction include work by Moin and Pierce [24] who originally introduced a reaction progress variable in combination with the steady flamelet equation. Recent progress using the unsteady flamelet equations is due to Pitsch, Ihme and their colleagues [25, 26]. For premixed combustion, other forms of the model have been proposed in the literature, particularly the G-equation model [21], where G is a level-set function that describes the flamefront (see Table 4).

In contrast to flamelets, the LEM approach solves the complete conservation equations [27, 28, 29]. The continuity, momentum and energy equations are solved on a global (LES) scale, while the species transport equations are solved in unfiltered form using a multi-scale approach. The overall solution procedure involves solving the LES equations only for the velocity, density and energy components of the flow. On the other hand, the species transport equations are solved in unfiltered form:

$$\rho \frac{\partial Y_k}{\partial t} + \rho u_j \frac{\partial Y_k}{\partial x_j} = -\frac{\partial}{\partial x_j}\left(\rho V_{j,k} Y_k\right) + \dot{w}_k \tag{34}$$

where V_{jk} is the diffusion velocity and $\dot{w}_k$ is the species production and destruction term. The convective term of the above equation can be rewritten in a slightly different form:

$$\rho \frac{\partial Y_k}{\partial t} + \rho\left(\tilde{u}_j + (u_j')^R + (u_j')^S\right) \frac{\partial Y_k}{\partial x_j} = -\frac{\partial}{\partial x_j}\left(\rho V_{j,k} Y_k\right) + \dot{w}_k \tag{35}$$

where we have substituted the unfiltered velocity in terms of the filtered velocity field ($\tilde{u}_j$), the LES-resolved sub-grid fluctuation ($u_j'^R$) and the unresolved sub-grid fluctuation ($u_j'^S$).

In the LES-LEM model, this equation is further split into two parts: the sub-grid part, comprised of the sub-grid mixing, diffusion and chemical source term; and the large-scale advective part. Thus:

$$Y_k^* - Y_k^n = \int_t^{t+\Delta t_{LES}} -\frac{1}{\rho}\left(\rho (u_j')^S \frac{\partial Y_k^n}{\partial x_j} + \frac{\partial}{\partial x_j}\left(\rho V_{j,k} Y_k\right)^n - \dot{w}_k\right) dt \tag{36}$$

$$Y_k^{n+1} - Y_k^* = -\Delta t_{LES}\left(\tilde{u}_j + (u_j')^R\right)\frac{\partial Y_k^n}{\partial x_j} \tag{37}$$

Note that the sub-grid species solution is accompanied by an approximate form of the thermal energy equation, which is necessary to properly capture the effects of the combustion heat release and the sub-grid thermal field. The sub-grid solution is obtained using a 1D sub-grid that is aligned in a direction along the maximum gradient of the species mass fraction. The sub-grid solution is comprised of three terms. The first term is the advective mixing term that is modeled using stochastic rearrangement events called triplet maps, which model the effects of sub-grid eddies of specified size, location and frequency [27]. The second term represents molecular diffusion and is solved using explicit time-stepping in the 1D LEM grid. The third term is the combustion source term, which is typically integrated using an ordinary differential equation (ODE) solution procedure. Finally, the large-scale advective solution is carried out using a Lagrangian approach and involves splicing and regridding (of the 1D LEM grid) operations to account for mass transfers between LES cells and volumetric expansion due to combustion. In principle, the LEM model is applicable for all flow and combustion regimes because it makes no assumptions of scale separation between combustion and turbulence. Further details on the LEM approach are given by Menon and Kerstein [28] and Patel [29].

Approaches that solve for a transported PDF or FMDF have also been proposed, where the PDFs of one or more variables are used to describe the full range of turbulent fluctuations [30, 31]. The Fokker-Planck equation is used to describe the evolution of the PDF, the solution of which involves the Langevin equation that describes a set of Lagrangian particles that are advected in phase space. As shown in Table 4, transported-PDF models represent the closure of the combustion source term exactly and can accommodate detailed chemical kinetics. The major approximations are in the form of the joint-PDF and in the modeling of the scalar mixing terms. See work by Pope [30, 31] and Givi [32, 33] for further details.

4.6. Multiphase Models

For multi-phase flow situations, a number of different approaches, ranging from Eulerian volume-of-fluid (VOF) methods and mixed Eulerian-Lagrangian methods, have been proposed. The VOF methods are good for resolving spray atomization and droplet formation and transport; however, these methods require extremely fine meshes to properly capture fine droplet sprays. While pure Eulerian methods have been proposed for spray problems, mixed Eulerian-Lagrangian methods are more commonly used in gas turbine analysis codes. These mixed methods solve the gas-phase transport equations using the conventional Eulerian approach, but handle the liquid phase droplets using a Lagrangian approach. While they are relatively efficient, fundamental physical phenomena cannot be naturally represented and have to be modeled. In this section, we review only the mixed methods because of their relative popularity for engineering multi-phase simulations.

Before we consider the liquid phase, we first provide the gas-phase source terms that account for the mass, momentum, energy and species transfer due to droplet vaporization:

$$H = \begin{pmatrix} \sum_N n\dot{m}_v \\ \sum_N n(\dot{m}_v u_i - m_l \dot{u}_i) \\ \sum_N n(\dot{m}_v h_v - m_l \dot{h}_d) \\ \sum_N n\dot{m}_v \delta_{lv} \end{pmatrix} \tag{38}$$

where N represents the number of parcels and n represents the number of drops per parcel.

The Lagrangian formulation for the liquid particle formulation is written as a set of ordinary differential equations for the Lagrangian solution variable Q_L:

$$\frac{dQ_L}{dt} = \frac{d}{dt}\begin{pmatrix} x_i \\ u_i \\ T_l \\ m_l \\ r \end{pmatrix} = \begin{pmatrix} u_i \\ F_D(u_i - v_i) \\ \dot{m}_v[c_p(T_g - T_l) - L_v]/m_l C_l \\ -\dot{m}_v \\ -\dot{m}_v/(4\pi r^2) \end{pmatrix} \tag{39}$$

where x_i, u_i, T_l, m_l and r are the components of the Lagrangian solution vector and these variables represent the position and velocity vector, temperature, mass and radius of the liquid particle respectively. The first and second rows represent the motion of particle droplets, where the drag is considered as an external force. The third row represents the energy equation for the liquid particle, which assumes a uniform temperature inside the droplet and accounts for sensible heating and latent heat of vaporization. The fourth row accounts for mass balance due to vaporization and the last row represents the reduction of the liquid particle radius due to vaporization. The set of Lagrangian phase equations is integrated explicitly and the Lagrangian time step is determined by consideration of the time scales of the particle lifetime, the atomization initiation time and the Eulerian time step. All other physical properties of the liquid particle are updated every time step as well.

The above formulation (Eq. 39) is capable of transporting the Lagrangian droplet particles and accounting for their vaporization into the Eulerian gas-phase. The actual formation of the droplets from the fuel spray still needs to be modeled and is usually done by using appropriate spray formation and atomization models. Table 5 shows models for the different stages along with their regimes of applicability given in terms of the Weber number (We). The first stage involves injection of the liquid jet into the combustor, and the physical model must be capable of resolving the liquid jet interface with the gas-phase—in many cases, both inside the injector and outside. As mentioned earlier, the VOF approach is often used to describe multi-phase problems involving an interface and the model involves tracking a scalar characteristic function that takes the value of zero when a cell contains no liquid, a value of unity when it is fully liquid and values inbetween when the cell contains the interface [34]. In order to properly track the interface, the VOF method may be combined with a level-set approach, which introduces a distance function from the interface. Such a combined method is referred to as the coupled level-set and volume-of-fluid (CLSVOF) approach, where the VOF method is used to properly account for mass conservation between the phases, while the level-set function is used to define the normal and curvature of the interface [35]. VOF models can also be directly used to capture primary and secondary atomization processes as shown by Yang *et al.* [36], but they are extremely expensive because of the grid resolution required to capture the interface between individual droplets. In practical computations, therefore, the VOF methods are best used to capture the overall spray characteristics such as the liquid sheet thickness, spray angle

Table 5. Primary and Secondary Atomization Models

Physical Process	We Number	Model	Results
Injection:		Volume of Fluid (VOF) Coupled Level-Set-VOF	Liquid sheet thickness Spray angle Sheet velocity
Primary Atomization:		Linearized Sheet Instability Atomization (LISA) Model	Ligament Size Droplet Size
Secondary Atomization:			
Vibration mode	$0 < We < 11$	No breakup	
Bag mode	$11 < We < 35$	Taylor Analogy Break-up (TAB)	Droplet size
Multi mode	$35 < We < 80$	TAB/Kelvin-Helmholtz (KH)	”
Sheet Thinning	$80 < We < 350$	KH Model	”
Catastrophic mode	$We > 350$	Rayleigh Taylor (RT)	”

and sheet velocity, and separate physical models are utilized for primary and secondary atomization processes (e.g., see Ashraf and Jog [37]).

Atomization of a liquid sheet involves the breakup of the sheet into ligaments and the subsequent break of the ligaments into droplets. This combination of processes constitutes primary atomization. The linearized instability sheet atomization model (LISA) of Senecal *et al.* [38] is a typical approach for representing these processes. Here, a linearized instability analysis of the liquid sheet is used to determine the ligament and droplet sizes and the associated breakup times. The ligaments and droplets resulting from primary atomization further encounter aerodynamic forces due to interactions with the gas-phase flow, which lead to a variety of secondary atomization processes depending on the Weber number. At low Weber numbers, we encounter strong surface tension effects, which leads to a pressure difference that blows the drop downstream forming a “bag” shaped structure. In time, the bag bursts into fine fragments or small droplets. The Taylor Analogy Break-up (TAB) model proposed by O’Rourke and Amsden is typically used to describe this bag mode [39]. The TAB model is based on the classical analogy between an oscillating and distorting droplet and a spring mass system. The external restoring and damping forces in the spring mass system correspond to the drag, surface tension, and viscous forces in the drop oscillation and distortion process. The model provides a solution for the droplet oscillation and, using an energy balance to describe the break-up process, the droplet diameter can be determined. Intermediate Weber numbers are governed by the so-called sheet-thinning mode, which is best described by the Kelvin-Helmholtz (KH) instability process. Reitz’s break-up model uses a linear stability analysis to determine growth rate and frequency, which are in turn used to define the droplet sizes [40]. At high Weber numbers, acceleration of the drop by aerodynamics leads to unstable surface waves at the leading edge of the droplet. These waves eventually penetrate the droplet and lead to breakup into fine fragments, a process that can be described using Rayleigh-Taylor (RT) stability analysis [41, 42]. Again, the analysis provides the wavelength and frequency of the fastest growing waves, from which the droplet size can then be determined. As discussed earlier, the droplets appear as source terms in the Lagrangian equations.

5. Sources of Uncertainty

Physics-based modeling of liquid propulsion systems is fraught with many of sources of uncertainty, the exact details of which depend on the particular application, i.e., gas turbine, rocket or scramjet.

Broadly, we may characterize the uncertainties as arising from either the physical model or the numerical solution procedure. Physical model errors may be characterized as being due to the limitations of the models for turbulence, combustion, and multi-phase phenomena. In the case of turbulence modeling, the uncertainties are similar to those found in other fluid dynamics applications, although the presence of multi-species transport in reacting flows introduces additional considerations. Importantly, the ability of RANS models to capture unsteady dynamics is questionable unless there is adequate scale separation between the turbulent time-scales and the physical time-scales of interest (hydrodynamics or acoustic or other). For most problems of interest to liquid propulsion, such scale separation does not exist and, generally, unsteady-RANS models are not applicable. Possible exceptions could be cases dealing with externally imposed transients such as throttle operation and vehicle acceleration. In the case of hybrid RANS-LES models, model inconsistencies between the near-wall RANS region and the off-body LES region also present problems. For instance, the definition of the turbulent kinetic energy is different in the two regions: it represents the total kinetic energy of the fluctuating field in the case of RANS models, while it represents the *unresolved* sub-grid kinetic energy in the case of LES models. Methods to reconcile the definitions of the two regions have been proposed [43] but are not widely used yet because research issues remain. A related and, perhaps, even more fundamental issue has to do with the definition and interpretation of the eddy viscosity in LES models in the presence numerical dissipation errors. To address such problems, explicitly filtered LES models have been proposed [44], but because of computational expense, such models have yet to enter the mainstream of engineering analyses. Finally, models for transition of laminar to turbulent flow remain a challenge in all of the turbulence modeling scenarios, i.e., RANS, LES or hybrid RANS-LES. It is fair to note that these models perform best in fully turbulent regimes and, while promising advances have been made in external aerodynamics flows, similar work for internal propulsion flows remain in their infancy.

For combustion problems, model uncertainties may be traced to two areas: combustion kinetics and turbulent combustion closure. In the former case, detailed kinetics of complex hydrocarbons can involve thousands of species and tens of thousands of chemical reactions. For simulations of complex engineering applications, such detailed models are not computationally feasible and global or reduced models are typically employed. In turn, such reduced models have the disadvantage of being valid only over narrow regimes or for specific purposes. In other words, a given reduced mechanism may be suitable for predicting combustion efficiency, but would be inappropriate for predicting ignition because it lacks certain key intermediate species or chemical steps. Turbulent combustion closure has been discussed in greater detail elsewhere in this summary and it is evident that these models introduce a number of physical assumptions that constrain the model validity to specific flow regimes. For instance, flamelets are valid only for high Damkohler numbers because of the underlying assumptions of the flame thickness. Moreover, most turbulent combustion models involve assumptions of incompressible flow to simplify the form of the energy equation, which in turn lead to limitations in their ability to handle compressible phenomena such as shocks and acoustic-combustion interactions.

In the case of multi-phase models, the Lagrangian model equations typically assume spherically-shaped droplets and uniform droplet temperatures. The models for primary and secondary atomization introduce further modeling uncertainties. In the case of the rocket conditions, the operating pressure is typically super-critical and multi-phase phenomena are usually neglected. However, recent research has indicated that multi-phase phenomena may persist under certain conditions involving super-critical mixtures [45]. Moreover, the equations of state of high-pressure fluids are

themselves not well established, introducing further uncertainties. Similar uncertainties are also present in the transport properties of the fluids and the associated mixture rules especially under high-pressure conditions.

The second category of uncertainties arises due to numerical errors. As noted elsewhere in this volume, considerable research has been focused on quantifying the influence of numerical accuracy for engineering simulations. However, we note that the field is still in its relative infancy when it comes to unsteady flows such as LES computations. Importantly, LES requires the resolution of a certain fraction of the energy-carrying scales. The grid resolution required to achieve this requirement can be very different for different algorithms and codes, a fact that is not widely appreciated in the engineering community. In this regard, we point out that a grid-filtered-LES model should eventually converge to the correct direct numerical simulation (DNS) solution, while an explicitly filtered LES model would converge to a grid-independent model-equation solution. In either case, it is important that the influence of grid size on the convergence of the statistical quantities in LES calculations be systematically verified.

A further element of LES computations is the inherent stochasticity of the problem. Boundary conditions and initial conditions are only known in terms of some statistically stationary quantities and the precise time-dependent nature needs to be properly accounted for. Moreover, in the case of multi-phase flows, obtaining grid-converged solutions is extremely challenging because of the need to include very large numbers of droplet parcels. Thus, the analysis of uncertainty quantification methods for time-dependent stochastically varying flows remains an active area of research and development.

A final consideration of liquid propulsion simulations is the process of validation. The presence of complex physics and complex geometries makes validation difficult, in part due to the challenges of obtaining detailed measurement data under such conditions. This is all the more true for time-dependent LES calculations which are needed for capturing multiphase combustion dynamics. Given these issues, there is a strong need for benchmark DNS solutions that can provide the basis for term-by-term validation of the physics models and the numerical methods. Such an approach would also be instrumental for quantifying uncertainties as well as for establishing the basis for improving the physical models themselves.

References

1. V. Yang, M. Habiballah, J. Hulka, and M. Popp, "Liquid Rocket Thrust Chambers: Aspects of Modeling, Analysis, and Design," volume 200, *AIAA Progress in Astronautics and Aeronautics*, 2004.
2. T. I. Shih and V. Yang, Turbine Science and Technology, *AIAA Progress in Astronautics and Aeronautics*, 2012.
3. D. C. Wilcox, *Turbulence modeling for CFD*, DCW Industries, La Canada, CA, 1998.
4. C. Menneveau and J. Katz, "Scale-invariance and turbulence models for large-eddy simulation," *Annual Review of Fluid Mechanics*, 32:1–32, 2000.

5. P. R. Spalart, "Detached-eddy simulation," *Annual Review of Fluid Mechanics*, 41:181–202, 2009.

6. M. Lightfoot, "Fundamental classification of atomization processes," *Atomization and Sprays*, 19:1065–1104, 2009.

7. V. Yang and W. Anderson, "Liquid Rocket Engine Combustion Instability," volume 169, *AIAA Progress in Astronautics and Aeronautics*, 1995.

8. J. C. Oefelein and J. Charles, "Large eddy simulation of turbulent combustion processes in propulsion and power systems," *Progress in Aerospace Sciences*, 42:2–37, 2006.

9. Poling B. E., Prausnitz J. M., and O'Connel J. P., *The Properties of Gases and Liquids*, McGraw Hill, fifth edition, 2001.

10. D. Peng and D. B. Robinson, "A new two-constant equation of state," *Industrial Engineering Chemistry Fundamentals*, 15:59–64, 1976.

11. G. Soave, "Equilibrium constants from a modified redlich-kwong equation of state," *Chemical Engineering Science*, 27:1197–1203, 1972.

12. Benedict, Manson, Webb, George B., and Rubin, Louis C., "An Empirical Equation for Thermodynamic Properties of Light Hydrocarbons and Their Mixtures: I. Methane, Ethane, Propane, and n-Butane," *Journal of Chemical Physics*, 8:334–345, 1946.

13. Weiss, J. M. and Smith, W. A., "Preconditioning applied to variable and constant density flows," *AIAA Journal*, 8:2050–2057, 1995.

14. Merkle, C. L., Sullivan, J. A. Y., Buelow, P. E. O., and Venkateswaran, S., "Preconditioning applied to variable and constant density flows," *AIAA Journal*, 36:515–521, 1998.

15. J. Troe., "Predictive Possibilities of Unimolecular Rate Theory," *Journal of Physical Chemistry*, 83:114–126, 1979.

16. Tsang W. and Herron J. T., "Chemical Kinetic Data Base for Propellant Combustion. I. Reactions Involving NO, NO2, HNO, HNO2, HCN and N2O," *J. Phys. Chem. Ref. Data*, 20:609–663, 1991.

17. U. Piomelli., "Large-eddy simulation: achievements and challenges," *Progress in Aerospace Sciences*, 35:635–362, 1999.

18. M. Germano, U. Piomelli, P. Moin, and W. Cabot, "A dynamic subgrid-scale eddy viscosity model," *Phys. Fluids A*, 3:1760–1765, 1991.

19. D. K. Lilly, "A proposed modification of the Germano subgrid scale closure method," *Phys. Fluids A*, 4:633–635, 1992.

20. M. Strelets, "Detached Eddy Simulation of Massively Separated Flows," *39th AIAA Aerospace Sciences Meeting and Exhibit*, 2001.

21. N. Peters, *Turbulent Combustion*, Cambridge Monographs on Mechanics, 2000.

22. R. O. Fox, *Computational models for turbulent reacting flows*, Cambridge Univ. Press, 2003.

23. T. Echekki, *Turbulent Combustion Modeling*, Springer, 2011.

24. C. D. Pierce, *Progress Variable Approach for Large Eddy Simulation of Turbulent Combustion*, PhD thesis, Stanford University, 2001.

25. Ihme, M., Cha, C., and Pitsch, H., "Prediction of local extinction and re-ignition effects in non-premixed turbulent combustion using a flamelet/progress variable approach," *Proc. of the Combustion Institute*, 30, 2005.

26. Ihme, M. and See, Y. C., "Prediction of autoignition in a lifted methane/air flame using an unsteady/progress variable model," *Combustion and Flame*, 157, 2010.

27. A.R.Kerstein, "Linear-Eddy Model of Turbulent Transport," *Combustion and Flame*, 75:397–413, 1989.

28. S. Menon and A. R.Kerstein, "Stochastic Simulation of the Structure and Propagation Rate of Turbulent Premixed Flames," *Proceedings of the Combustion Institute*, 24:443–450, 1992.

29. S. Menon and N. Patel, "Subgrid Modeling for Simulation of Spray Combustion in Large-Scale Combustors," *AIAA Journal*, 44(4):709–723, 2006.

30. S.B. Pope, "Pdf Methods for Turbulent Reactive Flows," *Progress in Energy Combustion Science*, 11:119–192, 1985.

31. S. B. Pope, "Computations of turbulent combustion: progress and challenges," *Proc. Comb. Inst.*, 23:591–612, 1990.

32. P. Givi, "Filtered density function for subgrid scale modeling of turbulent combustion," *AIAA Journal*, 44:16–23, 2006.

33. Yilmaz, S. L., Nik, M. B., Givi, P., and Strakey, P. A., "Scalar Filtered Density Function for Large Eddy Simulation of a Bunsen Burner," *J. Propulsion and Power*, 26:84–93, 2010.

34. Hirt, C. W. and Nichols, B. D., "Volume of fluid (VOF) method for the dynamics of free boundaries," *Journal of Computational Physics*, 39:201–225, 1981.

35. M. Sussman, M. Arienti, X. Li, M. Soteriou, C. Eckett, and B. Jensen, "Coupled Level-Set/Volume-of-Fluid Method for simulation of injector atomization," *Journal of Propulsion and Power*, 29:147–157, 2013.

36. Xiaodong Chen, Dongjun Ma, and Vigor Yang, "Coupled Level-Set/Volume-of-Fluid Method for simulation of injector atomization," *ILASS Americas, 24th Annual Conference on Liquid Atomization and Spray Systems*, 2012.

37. I. Ashraf and M. A. Jog, "Nonlinear breakup model for a liquid sheet emanating from a pressure-swirl atomizer," *Journal of Engineering for Gas Turbines and Power-Transactions of the ASME*, 129:945–953, 2007.

38. P. K. Senecal, D. P. Schmidt, I. Nouar, C. J. Rutland, R. D. Reitz, and M. L. Corradini, "Modeling high-speed viscous liquid sheet atomization," *International Journal of Multiphase Flow*, 25:1073–1097, 1999.

39. P.J. O'Rourke and A. A. Amsden, "The Tab Method for Numerical Calculation of Spray Droplet Breakup," *SAE Technical Paper*, 872089, 1987.

40. J. C. Beale and R. D. Reitz, "Modeling spray atomization with the Kelvin-Helmholtz/Rayleigh-Taylor hybrid model," *Atomization and Sprays*, 9:623–650, 1999.

41. Z. Y. Han, S. Parrish, P. V. Farrell, and R. D. Reitz, "Modeling atomization processes of pressure-swirl hollow-cone fuel sprays," *Atomization and Sprays*, 7:663–684, 1997.

42. M. A. Patterson and R. D. Reitz, "Modeling the effects of fuel spray characteristics on diesel engine combustion and emission," *SAE Technical Paper*, 980131, 1998.

43. H. Xiao and P. Jenny, "A consistent dual-mesh framework for hybrid LES/RANS modeling," *Journal of Computational Physics*, 231:1848–1865, 2012.

44. S. Bose, P. Moin, and D. You, "Grid-independent large-eddy simulation using explicit filtering," *Annual Research Briefs, Center for Turbulence Research, Stanford University*, pages 173–187, 2008.

45. R. N. Dahms and J. C. Oefelein, "On the transition between two-phase and single-phase interface dynamics in multicomponent fluids at supercritical pressures," *Physics of Fluids*, 25, 2013.

Review of Analytical Approaches for Modeling Solid Propulsion Rocket Motor Environments and Performance

Andrew M. Eaton

Orbital ATK Propulsion Systems, Magna, UT 84044

1. Introduction

An overview of current analytical approaches used to model the major physical mechanisms related to the behavior of solid propulsion rocket motor environments and performance is presented in this work. The discussion of these approaches will focus on performing simulations of solid rocket motor (SRM) chamber and nozzle flow environments as they pertain to predicting performance and providing boundary conditions for designing components to meet reliability requirements. In the context of this discussion, modeling and simulation refer to mathematical and computational techniques for characterizing the processes of interest, as opposed to experimental modeling methods. There is a close relationship between numerical experiments and physical experiments, however, and the best design practices use the two approaches to complement each other. This overview is divided into the following parts:

- Overview of solid propulsion physics
- Modeling and simulation objectives
- Modeling and simulation approaches
- Example of SRM modeling and simulation results
- Sources of modeling and simulation uncertainties

2. Overview of Solid Propulsion Physics

The major geometric configuration features and physical-chemical mechanisms associated with flow environments in an SRM with an aluminized propellant are shown in Figure 1. An excellent reference describing solid rocket propulsion fundamentals in more detail can be found in Sutton and Biblarz [1].

The major geometric features of an SRM are simply an igniter (not shown in Figure 1), a chamber containing the propellant, and a nozzle. Unique flow environments that must be modeled for the motor can be divided into four main areas: (1) transient ignition phenomena, (2) chamber phenomena, (3) nozzle phenomena, and (4) plume phenomena. Extensive modeling efforts have been focused on plume flow and are not considered here.

2.1. Ignition Environments

An igniter, often a small version of an SRM, provides the heat source that initiates propellant burning. As illustrated in Figure 2, the ignition event occurs in a few tenths of a second, during which time the flame initiated by the igniter spreads across the propellant surface and the motor reaches operating conditions of hundreds of psi in pressure and 3,000 to 4,000 K in temperature. Ignition of a solid propellant is a complex phenomenon governed by the interaction of many physicochemical processes described in references such as Sutton and Bublarz [1], Kuo [2], and Kulkarni, Kumar, and Kuo [3].

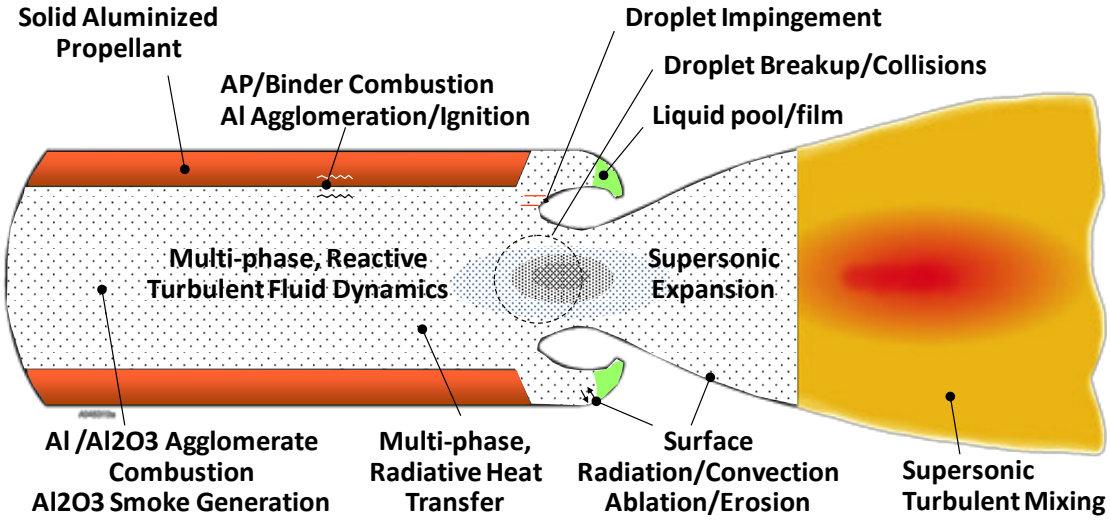


Figure 1. Major physical-chemical mechanisms in an SRM with aluminized propellant.

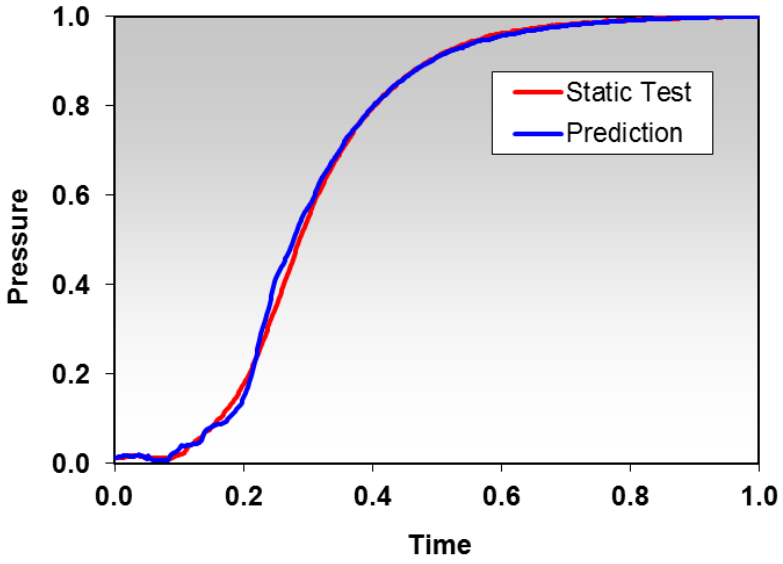


Figure 2. SRM ignition transient.

Major mechanisms include the following:

- Transient heat conduction, convection, and radiant heat flux arriving at the propellant surface from the igniter and from ignition of the propellant.
- In-depth conductive heating of the propellant. This discussion focuses on aluminized solid propellants. Major ingredients for typical solid propellants are aluminum (Al), ammonium perchlorate (AP), and a binder. Among the combustion products that complicate the modeling is the condensed-phase form of aluminum oxide (Al_2O_3) in the form of very small liquid or solid droplets.
- Given sufficient heating, propellant pyrolysis, and propellant surface and subsurface exothermic chemical reactions.
- Diffusion and convection of gaseous fuel, oxidizer, and combustion product species away from the propellant surface.
- Heating, melting, and chemical reactions of Al particles.
- Alumina and aluminum/alumina agglomerate droplets leaving the surface.

Methodologies based on these mechanisms and incorporated in computational fluid dynamics (CFD) codes have been developed that generally give very good predictions, as shown in the comparison between measured and predicted motor pressures in Figure 2.

2.2. Chamber Environments

The chamber is composed of a metal or composite structure case with a liner and insulator that contains the propellant, which is a mixture of oxidizer, fuel, and binder and has the appearance of a hard-rubber or plastic-like material. Once the entire propellant surface has ignited, the propellant combustion becomes self-sustaining and the following major mechanisms become important:

- Heat conduction, convection, and radiant heat flux arriving at the surface provided by the propellant combustion products.
- Reaction zone very close (on the order of 1 or 2 millimeters) to the surface of the propellant where propellant combustion is essentially completed. It begins with the heating and off-gassing of the AP/binder that provides gaseous oxidizing species and some gaseous combustion products. Heating of the Al particles fractures the oxide shell coating the particle and the Al and oxide melt and continue to increase in temperature. Some of the Al reaches its vaporization temperature and the gaseous Al reacts with the propellant oxidizing species to form micron-sized droplets. Remaining particles, which are now a mix of Al and Al_2O_3, melt together in pockets at the surface of the propellant. These pockets were created when the AP/binder, especially large AP particles, were gasified. The droplets forming in the propellant pockets reach relatively large sizes and ultimately leave the propellant surface to enter the flow field. These large-sized droplets, which are a mix of Al_2O_3 (often referred to as an oxide cap) and unburned Al, are referred to as agglomerates. Beckstead [4] has provided a review of data and theories related to agglomerate formation. While some trends related to AP size have been identified, there is no approach available that can predict agglomerate sizes as a function of propellant constituents.

- Transport of the micron-sized alumina (Al_2O_3) droplets and the larger agglomerates in the gaseous flow field. This combination of condensed-phase combustion products is typically bi-modal in nature, having a large fraction (~60 to 70%) of the micron-sized droplets of pure alumina and a smaller fraction (~30 to 40%) of the agglomerates ranging from tens to hundreds of microns in diameter. An example of the size distribution for typical aluminized propellant used in an advanced technology demonstration motor (hereafter referred to as DM) is shown in Figure 3. This size distribution was measured using a quenched particle collection bomb (QPCB) device. Al makes up approximately 19% by weight of the propellant, but alumina (the condensed-phase combustion product) makes up approximately 33% of the mass flow rate in the motor. The volumetric fraction of the alumina is very small in the chamber flow (~1e-04), except in certain regions where the droplets impinge on the surface and form liquid films or pools.
- Reaction of the agglomerates with oxidizing species in the chamber.
- Impingement and accumulation of alumina and agglomerate droplets on interior surfaces, which generate flowing liquid films and pools that are geometry-dependent.
- Convective and conductive heat flux on insulation/liner surfaces.
- Chemical decomposition and ablation of insulation/liner surfaces.
- Mechanical removal of charred insulation/liner due to surface-shear forces and impact from alumina droplets impingement.
- Pyrolysis off-gassing from insulation/liner surfaces.
- Radiant heat flux from the combustion products (primarily from the droplets) to insulation/liner surfaces.

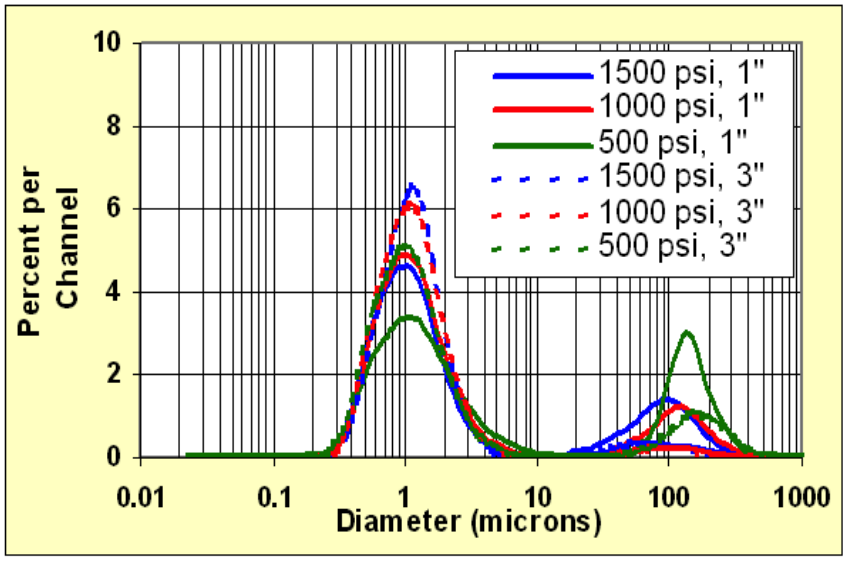


Figure 3. DM propellant condensed-phase combustion product (alumina) droplet size distribution, 1 inch and 3 inch refer to the distance between the burning propellant sample and the quench liquid surface.

2.3. Nozzle Environments

The nozzle is bolted to the case with its nose situated within the chamber (submerged as shown in Figure 1) or in-line with the aft-end of the chamber (de-submerged). It is designed to maximize the propulsive force available from the propellant combustion products. Major mechanisms governing the nozzle flow include:

- A complex combination of alumina and agglomerate breakup/coalescence due to aerodynamic shear and droplet collisions in the converging portion of the nozzle. In the converging portion of the nozzle, the flow accelerates from subsonic (Mach number <1) chamber flow to a Mach number of 1 at the nozzle throat.
- Multi-phase transport (largely convective) of combustion products accelerated to supersonic speeds and cooled as enthalpy of combustion is converted to kinetic energy in the expansion portion of the nozzle. The mass flow and velocity at the exit plane dictate the thrust that is generated by the motor.
- Potential freezing of the alumina droplets as they cool below their melting point.
- Convective and conductive heat flux on insulation/liner surfaces. Radiant heat flux is usually of secondary concern because of the cooling of the combustion products during expansion.
- Chemical decomposition and ablation of nozzle insulation surfaces.
- Mechanical removal of charred nozzle materials due to surface shear forces and impact from alumina droplets impingement.
- Pyrolysis off-gassing from nozzle insulation surfaces.

3. Modeling and Simulation Objectives

The primary environment simulation objectives for SRMs can be divided into two main areas:

1. Designing the chamber propellant configuration and nozzle to meet performance requirements. In its most fundamental form, the primary performance requirement is to deliver thrust as a function of the time needed to deliver a payload to a particular location within a specified amount of time. This should be achieved by minimizing as efficiently as possible the mass flow (and associated motor weight) rate required to generate a specified amount of thrust.
2. Providing boundary conditions (pressure, temperature, droplet impingement magnitude, and surface heat flux and shear forces) that can be used by motor component designers to optimize the design configuration within performance, physical (materials, size, and weight), and cost constraints.

The design process iterates through several cycles, and as the cycles progress, the simulation tools increase in their level of complexity. Simulation tools are also often required to assess the effects of scaling from small or subscale motors, where data are available, to the full-scale component motor designs.

4. Modeling and Simulation Approaches

The discussion of SRM modeling and simulation approaches is divided into two parts:

1. Traditional modeling approaches used to design an SRM.
2. Advanced modeling approaches based on multi-physics computational fluid dynamics (CFD).

4.1. Traditional Modeling Approaches

The traditional modeling and simulation approaches used in the design of an SRM can be divided into two areas:

1. Propellant grain design and ballistics prediction.
2. Nozzle design.

While the design of the motor igniter is also an important aspect of the design process, it is not discussed here. Details of the motor component design are also not discussed.

4.1.1. Propellant Grain Design and Ballistics Prediction

The propellant configuration in a motor is referred to as the propellant grain. Because the thrust is directly proportional to the mass flow rate, and the mass flow rate is directly proportional to the surface area of the propellant, the first step in the motor design process is to design the propellant grain with surface area and recession rate that will meet the mission requirements.

All of the complex physicochemical processes described previously that are involved with the combustion of solid propellants are modeled by an experimentally measured burn rate of the propellant given by the following expression:

$$r = ap^n \tag{1}$$

Where r is the burn rate in units of centimeters or inches per second, a is an empirical constant that is a function of propellant temperature, p is the chamber pressure just upstream from the inlet to the nozzle, and n is the burn rate exponent. Given this relationship between pressure and propellant surface recession and with the addition of one-dimensional isentropic relationships for flow in the nozzle, approaches such as those described in Reference 6 can be used to arrive at a grain design that meets the motor performance requirements. This calculation must be coupled with nozzle flow relationships to predict motor thrust as a function of the propellant mass flow rate.

4.1.2. Nozzle Design

The nozzle flow relationships are summarized in Figure 4 and derivations for these expressions have been described in several references, such as the text by Sutton and Biblarz [1].

$$F = \dot{m}v_e^2 + \left(p_e - p_a\right)A_e$$

$$F = p_0 A_t \left(\eta_{noz} C_F + \frac{p_e - p_a}{p_0}\frac{A_e}{A_t}\right)$$

$$F = \eta_{nozz} C_F \dot{m} \eta_{c^*} c^*$$

$$c^* = \frac{p_0 A_t}{\dot{m}} = \frac{I_{sp}}{C_F}$$

$$C_F = \sqrt{\frac{2k^2}{k-1}\left(\frac{2}{k+1}\right)^{(k+1)/(k-1)}\left[1-\left(\frac{p_e}{p_0}\right)^{(k-1)/k}\right]} + \frac{p_e - p_a}{p_0}\frac{A_e}{A_t}$$

$$C_{F-opt} = \sqrt{\frac{2k^2}{k-1}\left(\frac{2}{k+1}\right)^{(k+1)/(k-1)}\left[1-\left(\frac{p_e}{p_0}\right)^{(k-1)/k}\right]}$$

Figure 4. Rocket motor and nozzle flow relationships.

In these equations, the variables are as follows:

- A_e nozzle exit plane area
- A_t nozzle throat area
- c^* relates the ideal propellant combustion-to-thrust conversion
- C_F thrust coefficient (relates chamber pressure to ideal nozzle thrust)
- $C_{F\text{-}opt}$ optimized thrust coefficient (p_e=p_a)
- F motor thrust
- I_{sp} specific impulse (is also the ratio of thrust to mass flow rate)
- k ratio of specific heats (often referred to as gamma (γ))
- p_o chamber pressure
- p_a atmospheric pressure
- p_e pressure at the nozzle exit plane
- $\dot{m}$ mass flow rate
- η_{c^*} c^* efficiency (is the ratio of measured to ideal chamber combustion efficiency)
- η_{nozz} ratio of measured to ideal thrust coefficient
- v_e velocity at the exit plane of the nozzle

A close examination of η_{c^*} shows that it is a measure of the combustion efficiency of the propellant in the chamber, while η_{nozz} is a measure of the efficiency of the conversion of chamber

energy to thrust in the nozzle. These two factors must be determined experimentally. For a new motor design, these are estimated based on experience with similar motor configurations.

In conjunction with the grain design, the first step in motor design optimization focuses on the nozzle contour and expansion ratio optimization. Once the grain design has been established, the geometry and operational characteristics are used as input to a nozzle performance program, such as the industry-standard Solid Performance Prediction (SPP) program [7-9], which is a companion to the grain design program described in Reference 6. The basic approach used for the nozzle design performance prediction is divided into two steps:

1. Calculation of nozzle performance for ideal isentropic flow at either a vacuum or sea level ambient pressure. This becomes the reference for predicting nozzle design efficiency. A typical figure-of-merit used for assessing efficiency is the specific impulse, I_{sp}, which is the ratio of thrust to mass flow rate. The I_{sp} efficiency is the ratio of the actual I_{sp} to the ideal I_{sp}.
2. Calculation of nozzle performance loss factors. The loss factors considered in SPP are:
    a) Two-phase flow. The loss due to finite velocity and temperature differences between the gas- and condensed-phase combustion products. For an aluminized propellant, this is the largest loss factor and accounts for as much as 4 to 5% of the loss in specific impulse.
    b) Two-dimensional or divergence. The loss due to the momentum of the rocket exhaust that is not in line with the axis of the nozzle. This is usually calculated to be a fraction of a percent.
    c) Finite-rate kinetics. The loss due to incomplete transfer of latent heat to sensible heat caused by the finite time required for gas-phase chemical reaction to occur. This is also calculated to be around a fraction of a percent.
    d) Boundary layer. The loss due to viscous forces adjacent to the nozzle wall and heat transfer to the nozzle wall. This is typically a fraction of a percent.
    e) Combustion efficiency. As used in SPP, the loss that refers only to the degradation of I_{sp} due to a departure of chamber total temperature from the theoretically calculated total temperature. This is typically calculated to be a fraction of a percent and is usually so low that it is neglected.
    f) Erosion. The loss due to the erosion of the nozzle throat, which only encompasses the change in area ratio resulting from throat erosion during the motor burn time. This is usually around 1%.

Another loss included in legacy versions of SPP is referred to as a nozzle submergence loss, which is associated with the propellant grain extending axially past the nozzle inlet (Figure 1). This loss mechanism has been removed in the most current version of SPP, but it does represent a loss associated with the motor-nozzle configuration that needs to be accounted for because its magnitude averaged over 0.5% for motors with submergence [9].

Using method-of-characteristics approaches, for a given length and expansion ratio, the nozzle contour is optimized to maximize the performance efficiency.

As an example of the motor design approach, the grain and nozzle designs for the advanced demonstration motor (DM) discussed previously are shown in Figure 5, and the corresponding pressure and thrust versus time are shown in Figure 6. For this motor, the theoretical value of I_{sp}

at vacuum was calculated to be 303.6 lbf-s/lbm, while the pre-test delivered I_{sp} predicted by SPP after accounting for the performance losses was 287.2 lbf-s/lbm. The value of I_{sp} measured in the motor test was 283.4 lbf-s/lbm. Unfortunately, while improved motor performance technologies were demonstrated, not all the predicted performance gain was realized. The difference of 3.8 lbf-s/lbm is greater than typically found with SPP, which usually has a prediction accuracy within 1% of the measured value.

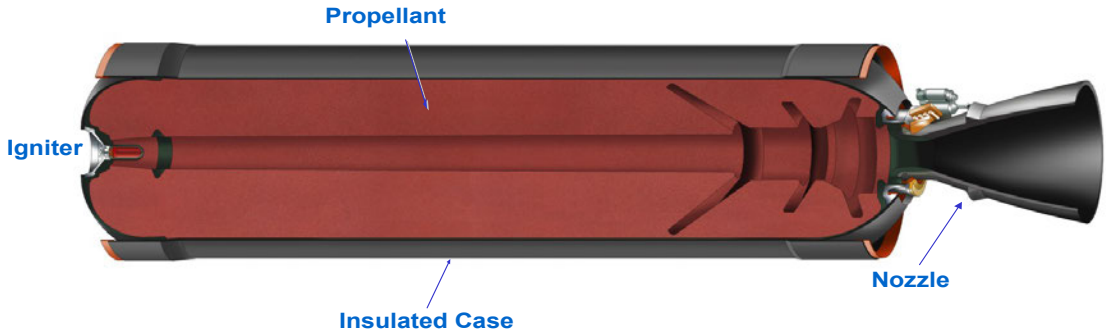


Figure 5. DM motor and nozzle configuration.

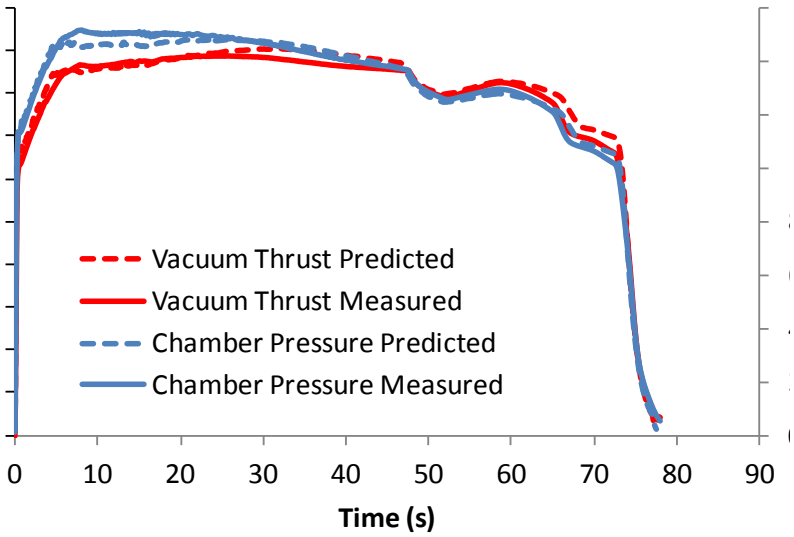


Figure 6. DM thrust and pressure versus time.

Once the grain has been designed, the essential load/boundary conditions (combustion product environment pressures and temperatures) are available for component design, which comprises material selections and configurations to meet design reliability and safety requirements. These designs often depend on empirical approaches based on motor histories and one-dimensional flow field predictions for the motor chamber and nozzle. The modeling and simulation approaches used in component design, such as structural analysis and thermal analysis models, are not considered here.

Table 1. CFD-Based Multi-Physics Modeling for Predicting SRM Environments

Reality	Conceptual/Mathematical Model	Computerized Model
Multi-phase Turbulent Fluid Dynamics	Reynolds-averaged Navier-Stokes, energy equation, dilute/dispersed droplets	Lagrangian/Eulerian CFD solver for chamber and nozzle flow
Turbulent 'Gas-phase' Reactions	Only Al in large agglomerate droplets is left to react with gaseous oxidizing species	Fast reaction of aluminum to frozen chamber products
Aluminum/Alumina	AL/agglomerate droplet reaction rate	Hermsen or Widener-Beckstead burn rate with single-step, fast reaction to gaseous species
Droplet Breakup	Droplet instability	Taylor Analogy breakup or critical Weber number criteria
Droplet Coalescence	Small droplets overtake and collide with large droplets	Collision/coalescence efficiency of small droplets colliding with agglomerates
Droplet Impingement, Accumulation	Under development	Under development
Wall Heat Loss	Energy equation with heat flux, temperature, or ablating surface boundary conditions	Resolve computational grid to laminar sublayer, wall functions
Multi-phase Radiant Heat Transfer	Radiative transfer equation for optically thick medium	P-1 spherical harmonics or discrete ordinates

4.2. Computational Fluid Dynamics

In recent years, CFD-based multi-physics models have been developed to better simulate motor environments and provide reliable boundary conditions for component design. An example was shown previously for motor ignition predictions. Sabnis and his team [10, 11] did pioneering work in this field, and Ebrahimi [12, 13] performed validation analyses. Whitesides and his team [14, 15] used the computer code CELMINT, which was the fruit of these efforts, in a variety of simulations related to Space Shuttle SRMs. More recently, the Center for Simulation of Advanced Rockets (CSAR) was established at the University of Illinois at Urbana-Champaign to develop the complex multi-physics models associated with SRMs. Typical results from their efforts in the area of CFD simulations are shown in References 16 and 17. Mississippi State University has also developed a multi-physics CFD-based code called Loci-CHEM [18-20] that has been adapted for use in simulations of SRM environments by NASA. "Loci" is a C++ library and declarative programming framework that efficiently maps numerical algorithms onto parallel computing architectures. "CHEM" refers to the particular numerical algorithms for solving the Navier-Stokes equations for non-equilibrium flow involving chemical reactions.

Commercial CFD codes are also being used to develop advanced multi-physics models for SRMs. Through the use of user-defined functions, these efforts build on the work done in developing CELMINT and in many ways parallel the work done at CSAR and with Loci-CHEM. Examples of this work are described in References 21-23. Table 1 summarizes the physicochemical mechanisms shown in Figure 1 together with the corresponding mathematical and numerical models used in the CFD-based multi-physics models described in the references. A brief description of the CFD-based models and the solution procedure that correspond to the physicochemical mechanisms shown in Figure 1 is provided in the following sections.

4.2.1. Multi-Phase Turbulent Fluid Dynamics

The foundation for the multi-phase CFD-based SRM models that is common to all the approaches described in the referenced work is turbulent, compressible gas-fluid dynamics. Details of these modeling approaches are provided in countless references and are not described in detail here. These models are available in both research codes and commercial CFD codes. The codes provide a variety of turbulence models that have also been described in the literature, and, again, are not described in detail here. Of the variety of turbulence models available, the two-equation models, such as the Realizable k-ε and k-ω, are those that have been typically used in modeling SRM internal flows.

In performing the motor simulations, analyses are traditionally divided into two parts. First are the predictions of the motor ignition transient, and second is the prediction of the motor flow field after ignition. An important assumption in modeling the motor flow field after ignition is that the flow field solution can be approximated as a series of steady-state flow simulations at various times during the motor burn. Given the motor flow field geometry with the grain burn-back profile at the burn-time of interest, the quasi-steady flow approximation has been judged reasonable for times after the onset of steady-state motor operation at about one second into the burn time. Recent work has been done to simulate grain recession as part of the simulation process, but this work is still in the validation stage and is not described here.

In considering two-phase flow in SRMs, the three major approaches that have been used are shown in Table 2. The different alternatives are available to simplify the solution procedure if various phenomena are not judged to be important in the problem of interest. For example, in the heavy gas approach, a single-phase flow solution is modeled where all the aluminum oxide is assumed to be in thermal and mechanical equilibrium (equal temperature and velocity) with the gaseous flow. The presence of the droplets in the gaseous phase is accounted for in the determination of a mixture of molecular weight and specific heat, but the droplets are assumed to occupy negligible volume.

In the Eulerian-Lagrangian simulation technique, the gas phase is modeled as a continuum that is simulated using the Reynolds Averaged Navier-Stokes (RANS) equations and the energy equation. Because of the low volumetric fraction, the droplets can be modeled as a discrete phase with trajectories passing through the gas continuum. The size distribution for the droplets is based on data measured in QPCB with distribution such as that shown in Figure 3. Typically, only the droplet sizes greater than 10 microns are included in the discrete phase. This solution scheme is illustrated in Figure 7. The addition of the radiation field solution is also included in this scheme.

Table 2. Modeling Assumptions Used in Two-Phase Flow Simulations

Single-Phase Heavy Gas Flow	Al_2O_3 is in mechanical, thermal, and chemical equilibrium with the gas phase Gas-droplet mixture equilibrium properties Gas properties frozen at chamber equilibrium conditions
Two-Phase Frozen Chemistry (Eulerian-Lagrangian Non-Reacting Flow)	Large Al_2O_3 (> 10 microns) tracked in discrete phase Gas-droplet mixture properties based on fraction of Al_2O_3 droplets in the discrete-phase Gas properties frozen at chamber equilibrium conditions Droplet breakup as a function of Weber number
Two-Phase Reactive Chemistry (Eulerian-Lagrangian Reacting Flow)	Large Al_2O_3 (> 10 microns) tracked in discrete phase Gas-droplet mixture properties based on fraction of Al_2O_3 droplets in the discrete-phase Unburned Al tracked in agglomerates Mixture properties adjusted to reflect tracked Al_2O_3 and Al Equilibrium gas properties throughout Particle breakup as a function of Weber number

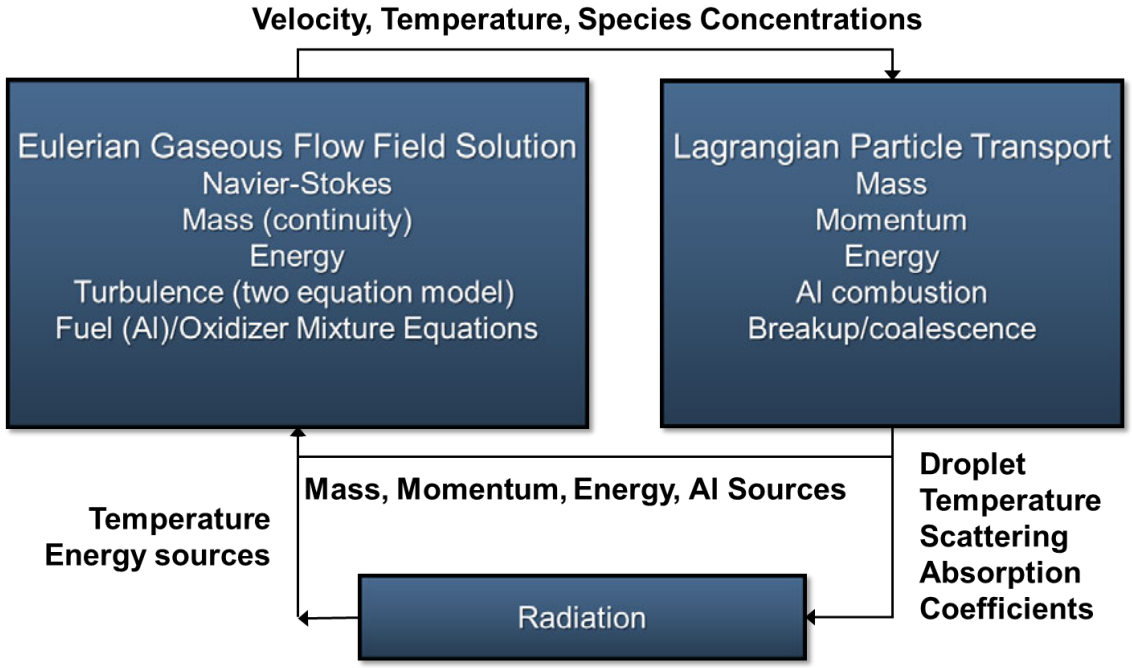


Figure 7. The Eulerian-Lagrangian two-phase flow simulation scheme.

In the non-reacting two-phase flow approach, it is assumed that sufficient time is available for the combustion processes to arrive at equilibrium in the motor chamber and the two-phase predictions can proceed as if the combustion process has reached equilibrium and the flow field is then chemically frozen. This is a reasonable assumption at distances a few inches away from the

propellant surface. This approach can be further simplified by assuming that the largest droplet sizes (the agglomerates referred to previously) are tracked and the remaining Al_2O_3 is included in the gas phase using the heavy gas approach.

Details for the Al reacting flow model and the additional submodels summarized in Table 1 are discussed in the following section.

4.2.2. Aluminum Droplet Combustion With Gaseous Oxidizing Species

In modeling Al combustion for aluminized propellant in an SRM, it is assumed that the Al combustion process is a two-step sequence [11, 23]. In the first step, the Al droplets vaporize to form Al in a gaseous state, which is referred to as smoke. An important assumption in this step is that the Al already exists in a bi-modal sized distribution as described previously. This initial condition is unique for each propellant and is based on QPCB measurements. During the droplet transport, the Al combustion mechanism from Widener and Beckstead [32] is used to calculate the rate of Al vaporization, which relates the change in mass of the droplets as a function of time to the local oxidizing species concentrations. To represent the bulk features of aluminum agglomerate combustion, the agglomerates are kept at the boiling point of Al while they are burning. Once the Al has burned off, the agglomerates are allowed to exchange heat with the surrounding gas and smoke mixture.

In the second step of the process, it is assumed that the kinetics of the gaseous combustion are very fast and can be ignored when compared with the mixing rate of the reacting species. With this assumption, once the fuel and oxidizer mix, the combustion products are at an equilibrium condition. To track the mixing, this non-premixed combustion model utilizes two transport equations with one equation for the Al (the fuel stream) and the other for the AP/binder/smoke (the oxidizer stream). An equilibrium chemistry table, covering the possible range of Al mixture fraction with the oxidizing stream and temperature, is built in a pre-processing stage. The following species mass fractions are retrieved from the equilibrium table at each local computational cell: Al_2O_3-liquid-smoke, CO, CO_2, Cl, HCl, H, OH, H_2, H_2O, N_2, O_2, O, AlCl, AlClO, $AlCl_2$, $AlHO_2$, NO, and Al-liquid. The equilibrium table does not include solid (frozen) Al_2O_3-smoke since it is assumed that liquid droplets will not have time to freeze in the short-lived residence times in the nozzle. Mass (removal of Al and O atoms) and energy balances were made at the propellant inlet boundary for the oxidizer stream to account for the high temperature ($T_{Al\ boiling\ point}$ = 4,184 K at 1,000 psia) Al/Al_2O_3 agglomerates that are tracked in the discrete phase.

4.2.3. Droplet Breakup And Coalescence

As the droplets approach regions of rapidly changing velocity, such as flow from slots or in the converging section of the nozzle, aerodynamic forces can be large enough to cause agglomerates to deform and breakup. To account for the change in droplet drag due to deformation, a dynamic drag model has been implemented based on experimental observations by Hsiang and Faeth [25]. They found that the normalized cross-stream droplet diameter (deformed/original diameter) was a function of the local Weber (We) number:

$$\frac{d_{c\max}}{d_0} = 1 + 0.19\sqrt{We} \tag{2}$$

They also developed a correlation between the drag coefficient and the normalized cross-stream diameter. The drag becomes ~3 to 4 times higher for a flattened droplet compared to a spherical droplet.

Droplet breakup is allowed to occur once the droplets reach a critical Weber number. Based on comparisons to droplet breakup data for liquid mercury [26] it was found that the critical Weber number was a function of original agglomerate size according to:

$$We_c = 0.0185 D_0 + 1.45 \tag{3}$$

However, the magnitude of the critical Weber number continues to be a source of uncertainty. Once the droplet Weber number reaches the critical value, the model will break the parent droplet into two smaller droplets. Numerically, the agglomerates undergo a series of breakup events until the final size is reached somewhere near the nozzle throat. The actual breakup event, where the droplet forms a bag or umbrella shape with many smaller droplets and several larger ligaments, is not resolved in the CFD model. The simple critical Weber number model can only provide the same final mean droplet size and not the full droplet size distribution.

A sweeping coalescence model that accounts for the growth of agglomerates in the motor has been implemented in a manner similar to the approach used in CELMINT [11]. The Lagrangian-phase tracks the large droplets, which grow as smoke collides with the larger droplets while the smoke is depleted. The coalescence model includes a collection efficiency that is related to the small droplet Stokes number. A small coalescence factor (0.0003) was needed to produce reasonable coalescence rates and no coalescence is assumed to occur in the nozzle.

4.2.4. Droplet Impingement, Accumulation

Two options are available for modeling droplets impinging on a solid surface: trapping (ending the trajectory) or reflection with a specified degree of momentum restitution.

The accumulation is estimated by integrating the mass flow of the droplets that are trapped as a function of time from the beginning to the end of the motor burn time. The modeling of the accumulation as a film flowing along motor chamber or nozzle surfaces, or as a pool that accumulates in volumes such as those created in the aft-end of a motor by a submerged nozzle, is currently under development and are not reviewed here.

4.2.5. Wall Surface Boundary Condition

Three options are available for the wall surface boundary condition that is required for the energy equation solution: specified surface temperature, specified surface heat flux, or an ablating surface that uses a specified temperature and adds a wall off-gassing source and composition. The wall surface boundary condition is usually specified in conjunction with analyses performed using thermal-ablation heat transfer codes.

4.2.6. Multi-Phase Radiant Heat Transfer

Radiative heat transfer is a key physical process in an SRM environment that strongly influences wall temperature and events such as ignition. The transfer of radiative energy is accounted for by solving the transport equation for radiative energy (RTE).

$$\frac{dI(\vec{r},\vec{s})}{ds} + (a+\sigma_s)I(\vec{r},\vec{s}) = an^2\frac{\sigma T^4}{\pi} + \frac{\sigma_s}{4\pi}\int_0^{4\pi} I(\vec{r},\vec{s}')\ \Phi(\vec{s}\cdot\vec{s}')\ d\Omega' \tag{4}$$

In this equation, I is the radiative intensity, $\vec{r}$ is the position vector, $\vec{s}$ is the direction vector, a is the absorption coefficient, σ_s is the scattering coefficient, n is the refractive index, σ is the Stefan-Boltzmann constant, Φ is the scattering phase function, and Ω represents the solid angle. The RTE is an integral-differential equation that is a function of both wavelength and direction. The spectral and directional dependence of the RTE make it difficult to solve numerically and simplifications are necessary to solve the equation within the framework of CFD codes. Two approaches have been commonly used for SRM models [23], the P-1 and Discrete Ordinates methods.

- **P-1 Method.** One of the most common approaches for solving the RTE is the P-N method, which is derived by expanding the RTE into an orthogonal series of spherical harmonics [27, 28]. P-1 is the simplest approximation and is based on approximations that are valid only for an optically-thick medium. This approximation converts the RTE into an ordinary differential equation that can be solved within the same framework and using the same computational grid as the other transport equations for momentum and energy. The influence of droplets/particles can be included in the radiation model by modification of the absorption and scattering terms.

- **Discrete Ordinates Method.** Another common approach used in CFD to solve the RTE is the discrete ordinates method (DOM) [27, 28]. While the P-1 model is only valid in optically-thick mediums, in the DOM approach, the full RTE equation is discretized over a finite number, N, of solid angles. The DOM method is valid over the range of optically-thick and optically-thin regimes and accounts for both transmissivity and reflectivity. Accuracy of the solution can be increased through the use of finer discretization. However, the resulting increase in the number of transport equations makes solutions increasingly time consuming. In the particular case of solid rocket propellants, the two-phase mixture is in most regions optically thick and the assumptions used in developing the P-1 model are valid. In this environment, predictions using the P-1 approach are similar to those obtained with the DOM method.

5. Example of SRM Modeling and Simulation Results

As an example of results for the CFD analysis of SRM environment and performance, the predictions for the DM are presented. Predictions were made using all three of the modeling approaches shown in Table 2, but only a few representative results of the reacting flow simulation are included to illustrate results from the use of the multi-physics modeling approach.

Figure 8 shows examples of three of the six burn-time geometries that were modeled for the DM. In Figure 9, predicted Mach number contours are shown for the DM at initial and near the end of the burn. The erosion pattern in the nozzle produced a shock wave that reduces the peak Mach numbers along the centerline and the influence of the droplets on the gas reduces Mach numbers toward the outside of the nozzle. Near the walls, Mach numbers increase since the droplet concentrations in this region are low.

Figure 10 shows both the 40 and 200 μm mean droplet trajectories colored by droplet diameter. In all cases, the agglomerates break up to somewhere between 14 to 18 μm in the throat region and all are about the same size in the nozzle.

Figure 11 shows plots of droplet temperature for the 200-mm trajectories at both initial and mid-burn. The maximum droplet size tracked for the DM simulations was 200 μm. The agglomerate droplets stay at their boiling point until all the Al is vaporized, at which time the droplet that is now pure Al_2O_3 cools to equilibrium with the gaseous species. As shown in Figure 11, essentially all of the Al is burned prior to the nozzle throat, even at initial burn time. Later in burn, with lower velocities in the chamber, all combustion takes place fairly close to the propellant surface.

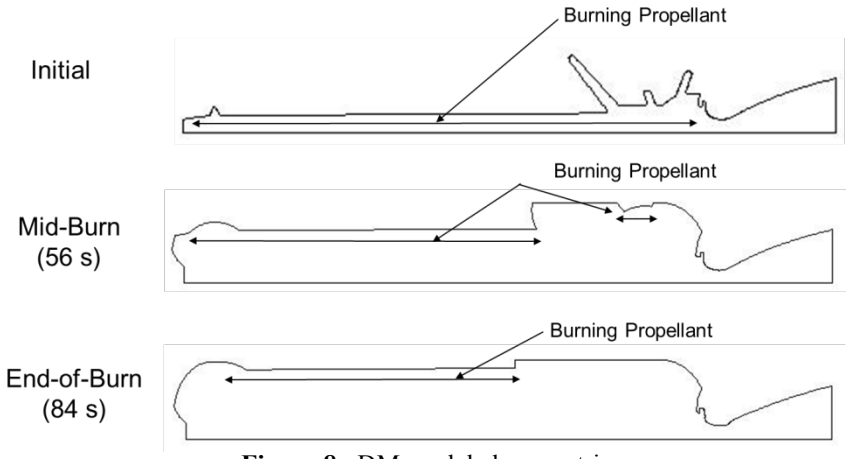


Figure 8. DM modeled geometries.

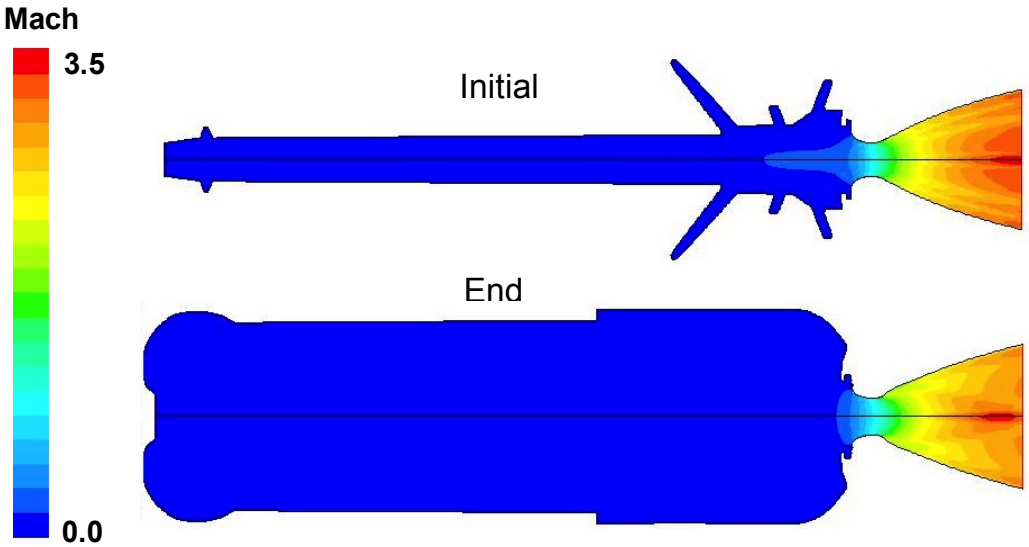


Figure 9. Gaseous flow Mach numbers.

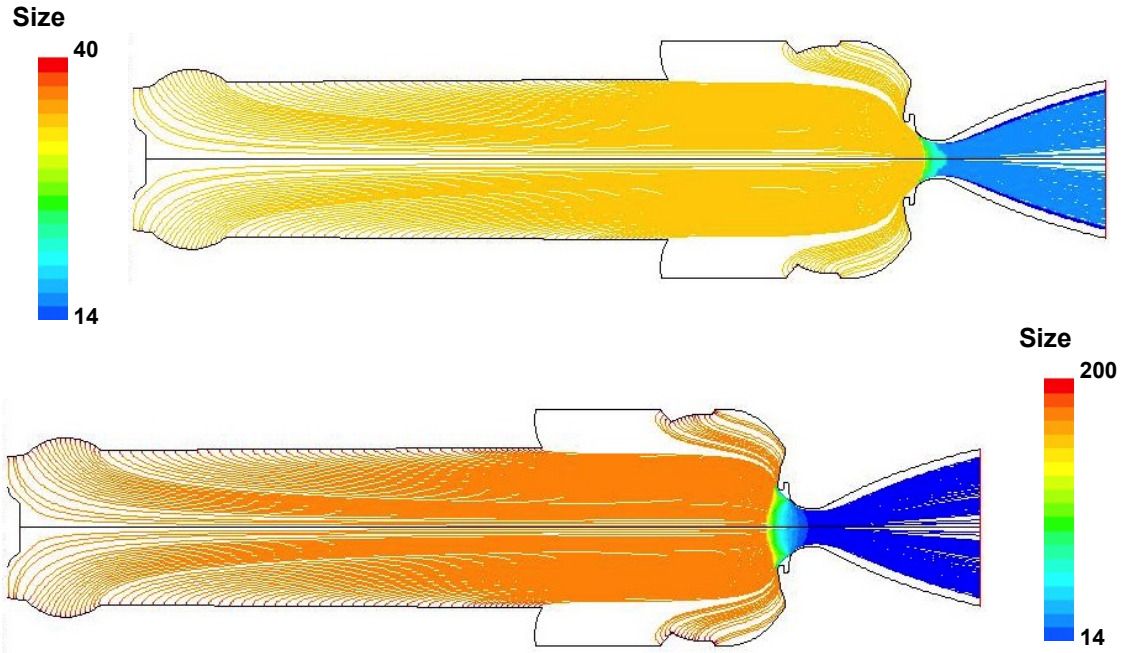


Figure 10. Droplet sizes at mid-burn along trajectories that start at 40 and 200 μm.

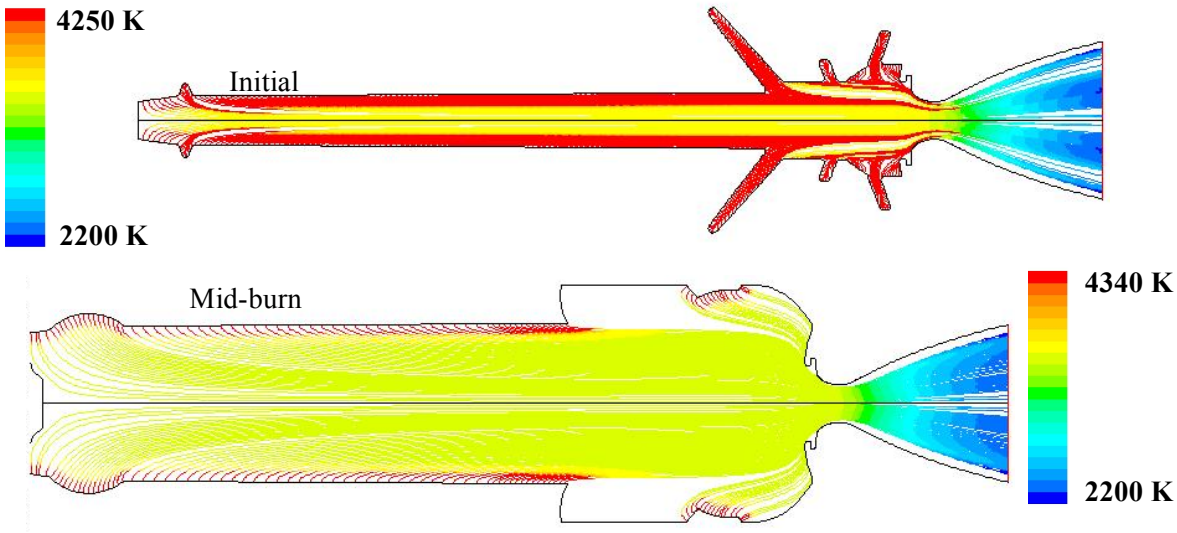


Figure 11. Droplet temperatures along the 200-mm droplet trajectories.

The specific impulse (I_{sp}) compared with data for the DM is shown in Figure 12. Results are shown for the six burn times analyzed using the three different modeling approaches from Table 2. The predicted values are in good agreement with measured values, and the results show that, at least for performance predictions, there is little difference in the modeling approaches. Since there was only one firing for this motor, there are no statistically-related error bands associated with the measured data.

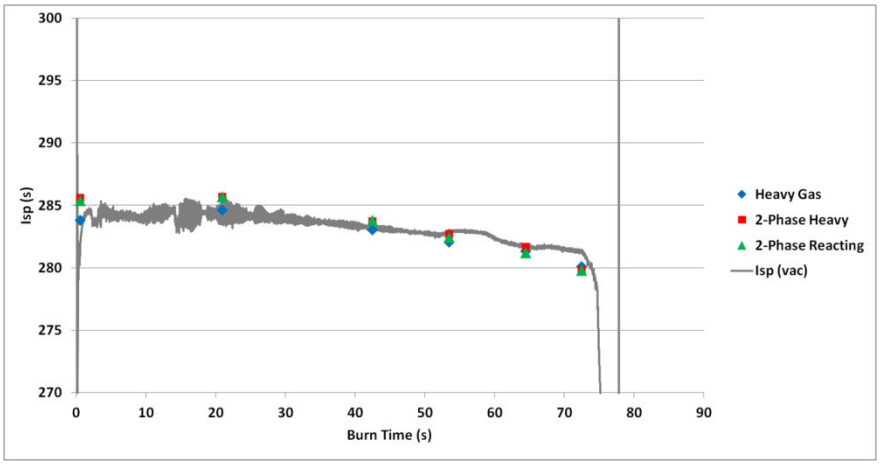


Figure 12. Comparisons of predicted and measured motor I_{sp} at six burn times.

6. Sources of Modeling and Simulation Uncertainties

While SRMs have been successfully built and fired for many decades, there are still many unknowns associated with the flow environments in motor chambers and nozzles that make it challenging to validate the predictions made with detailed multi-physics approaches and establish levels of uncertainty. Limited uncertainty estimates of models from a numerical standpoint, such as those described in References 20 and 21, are on going. In addition to the uncertainties associated with CFD simulations of turbulent compressible gas dynamics in general, which are not reviewed here, other sources of major modeling and simulation uncertainties related to the physics associated with SRM performance and environment predictions are dominated by the following:

1. Uncertainties in burn-rate data. These uncertainties are greatest for a new propellant or new motor design and can be attributed to two main sources. First, first-time estimates of burn rates are obtained from subscale testing and there are uncertainties associated with scaling these data to full-scale motor geometries. This uncertainty is on the order of ±3%. There are also variations in burn-rate that occur because of the way propellant is cast in a motor chamber. This uncertainty is initially on the order of 2%. However, factors to account for these variations are applied as needed to the burn-rate after the first full-scale motor firing, and uncertainty in propellant burn rate decreases as the number of motor firings increase.
2. Lack of flow field data at typical operating temperatures and pressures. This is particularly true for an SRM that utilizes an aluminized propellant because of the high flame temperature and high-pressure multi-phase flow environment associated with the propellant combustion. Because of this harsh environment, the motor becomes essentially a black box that precludes measurement of almost any variable of interest in the chamber but pressure and thrust. For example, no data are available for measurements of local velocities or temperatures. Environments must be implied based on post-test measurements and observations related to the material response of components, but the material response models also have several areas of uncertainty.

3. Lack of measured heat flux data to validate the predicted convective and radiative heat flux magnitudes, which makes is difficult to estimate a magnitude of uncertainty.

4. Two-phase flow uncertainties, starting with initial droplet sizes and the evolution of the droplets size distribution from propellant surface to exit plane. In return for a fuel that provides attractive density/impulse performance gains, the rocket motor designers must contend with condensed-phase combustion products that make up about one-third of the mass flow in the form of liquid alumina droplets. As described previously, the droplet size distributions in the combustion products are typically bi-modal in nature, with one mode around 1 to 3 microns and the other mode on the order of 10 to 500 microns. Uncertainty in the small-sized mode is on the order of ±1 micron, while the uncertainty on the large sizes can be around an order of about 100 microns. These droplets become liquid projectiles in the motor chamber and nozzle and can lead to increased material erosion on almost any surface where they impinge. To make matters worse, the impinging droplets can accumulate, creating streams of hot slag flowing along surfaces or accumulating in gaseous recirculation zones, such as the aft-end of a motor with a submerged nozzle. These pools of useless slag must be carried along as dead weight and portions can be re-entrained in the gas and expelled at inopportune times causing pressure blips or phantom nozzle loads. It is probably not an understatement to say that the two-phase nature of the flow environment is the single most difficult source of modeling uncertainty that makes it difficult to scale results from one size rocket motor to another or from one design to another.

5. Material property uncertainties. For environment predictions, the greatest uncertainties are associated with condensed-phase combustion products like alumina and alumina-aluminum agglomerates, as discussed previously.

6. Combustion/reactive flow modeling uncertainties. As the results for the DM show, the SRM simulations with combustion from a performance standpoint can be as accurate as more simplified modeling approaches in predicting motor performance, which is on the order of 1% for average I_{sp}. There is much greater uncertainty in providing flow and thermal boundary conditions because there are no data directly available for validation. A measure of accuracy is provided by the material response data, but estimating the uncertainty in material thermal response models, which is not considered in this work, compounds this. More detailed data for both understanding basic mechanisms and for validation are required to develop better models in both flow environment and thermal response modeling.

References

1. Sutton, G. P., and Biblarz, O., *Rocket Propulsion Elements*, 8th ed., John Wiley & Sons, Inc., New York, 2010.

2. Kuo, K. K., *Principles of Combustion,* John Wiley & Sons, Inc., New York, 1986.

3. Kulkarni, A. K., Kumar, K., and Kuo, K. K., "Review of Solid Propellant Ignition Studies," AIAA Paper 80-1210, July 1980; *AIAA J.,* Vol. 20, No. 2, pp. 243-244, 1982.

4. Beckstead, M. W., "An Overview of Aluminum Agglomeration Effects," *JANNAF Journal of Propulsion and Energetics*, Vol. 1, No. 1, pp. 1-16, 2008.

5. Glaittli, S. R., et al., "Integrated High-Payoff Rocket Propulsion Technology: Phase I Solid Boost Demonstrator," AFRL-PR-ED-TR-2001-0025, Final Report, August 2002.

6. Dunn, S. S., and Coats, D. E., "3-D Grain Design and Ballistic Analysis Using the SPP97 Code," AIAA Paper 97-3340, 1997.

7. Coats, D. E., French, J. C., Dunn, S. S., and Barker, D. R., "Improvements to the Solid Performance Program (SPP)," 39th AIAA / ASME / SAE / ASEE Joint Propulsion Conference, AIAA Paper 2003-4504, Huntsville, AL, July 2003.

8. Coats, D. E., Nickerson, G. R., Dang, A. L., and Dunn, S. S., "Solid Performance Program (SPP)," AIAA Paper 87-1701, 1987.

9. Coats, D. E., and Dang, A. L., "Improvements to the Solid Performance Program (SPP'12) and a Review of Nozzle Performance Predictions," AIAA Paper 2014-1804, 2014.

10. Sabnis, J. S., Gibeling, H. J., and McDonald, H.,"Navier–Stokes Analysis of Solid Propellant Rocket Motor Internal Flows," *Journal of Propulsion and Power*, Vol. 5, No. 6, pp. 657–664, 1989.

11. Sabnis, J. S., "Numerical Simulation of Distributed Combustion in Solid Rocket Motors with Metallized Propellant," *Journal of Propulsion and Power*, Vol. 19, pp. 48-55, 2003.

12. Ebrahimi, H. B., "Validation and Evaluation of the CELMINT Code for Two-Phase Flow in Solid-Propellant Rocket Motors," AIAA Paper 96-0710, 1996.

13. Ebrahimi, H. B., "Validation Database for Propulsion Computational Fluid Dynamics," *Journal of Spacecraft and Rockets*, Vol. 34, pp. 642-650, 1997.

14. Whitesides, R. H., et al, "Design of a Subscale Propellant Slag Evaluation Motor Using Two-Phase Fluid Dynamic Analysis," AIAA Paper 96-2780, 1996.

15. Whitesides, R. H., et al. "Application of Two-Phase CFD Analysis to the Evaluation of Asbestos-Free Insulation in the RSRM," AIAA Paper 97-2861, 1997.

16. Dick, W., Heath, T., Fiedler, R., and Brandyberry, M., "Advanced Simulation for Solid Propellant Rockets", 41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, AIAA Paper 2005-3990, July 10-13, 2005.

17. Dick, W. A., Fiedler, R. A., and Heath, M. T., "Building Rocstar: Simulation Science for Solid Propellant Rocket Motors," *Proceedings of the AIAA Joint Propulsion Conference*, AIAA Paper 2006-4590, July 2006.

18. Luke, E. A., et al., "CHEM- 2: A Finite-Rate Viscous Chemistry Solver – The User Guide," Technical Report MSSU-COE-ERC-04-07, Mississippi State University, 2004.

19. Hebert, S., and Luke, E. A., "Honey, I Shrunk the Grids! A New Approach to CFD Verification," 43rd AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 2005-0685, January 2005.

20. Veluri, S. P., Roy, C. J., and Luke, E. A., "Comprehensive Code Verification for an Unstructured Finite Volume CFD Code," 48th AIAA Aerospace Sciences Meeting Including the New Horizon Forum and Aerospace Exposition, AIAA Paper 2010-127, January 2010.

21. Eaton, A. M., et al., "Modeling the Gas Dynamics Environment in a Subscale Solid Rocket Test Motor," AIAA Paper 2001-2584, July 2001.

22. Ahmad, R. A., Morstadt, R. A., and Eaton, A. M., "RSRM and ETM03 Internal Flow Field Comparisons," AIAA Paper 2003-5104, 2003.

23. Black, D. L., and Eaton, A. M., "A Reacting Multi-phase Flow Model for Solid Rocket Motor Analysis," 50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference, AIAA Paper 2014-3618, 2014.

24. Widener, J. F., and Beckstead, M. W., "Aluminum Combustion Modeling in Solid Propellant Combustion Products," AIAA Paper 98-3924, 1998.

25. Hsiang, L.-P., and Faeth, G. M., "Near-Limit Drop Deformation and Secondary Breakup," *International Journal of Multiphase Flow*, Vol. 18, No. 5, pp. 635-652, 1992.

26. Craig, J. E., "Conventional and Liquid Metal Droplet Breakup in Aerodynamic Nozzle Contractions," 22nd AIAA Aerospace Sciences Meeting, AIAA Paper 84-0201, 1984.

27. Siegel, R., and Howell, J. R., *Thermal Radiation Heat Transfer*, Hemisphere Publishing Corporation, Washington DC, 1992.

28. Brewster, M. Q., *Thermal Radiative Transfer & Properties*, John Wiley & Sons, Inc., New York, 1992.

Biographical Sketches of Primary Contributing Authors

"An expert is someone who knows some of the worst mistakes that can be made in his subject, and how to avoid them."

— Werner Heisenberg

Dr. Timothy Barth is a Research Scientist at NASA Ames Research Center. His work has included the development of finite volume methods, stabilized finite element methods, multiscale methods, error and uncertainty analysis, and turbulence modeling. He received his doctorate in aeronautics and astronautics from Stanford University. He holds the AIAA Lawrence Sperry Award and the NASA Medal for Exceptional Scientific Achievement. Dr. Barth has been a lecturer at the von Karman Institute for Fluid Dynamics and the summer programs at both Freiburg University and the ETH Zurich. He also serves on the editorial board for the Springer book series *Lecture Notes on Computational Science and Engineering* (LCSE) and *Textbooks in Computational Science and Engineering* (TCSE).

Dr. Peter Cavallo is a Senior Research Scientist at Combustion Research and Flow Technology, Inc. (CRAFT Tech). He has over 20 years of experience in applied and computational aerodynamics, with expertise in unstructured-grid computational fluid dynamics methods encompassing mesh generation, adaptation, and modification; data structures and search algorithms; moving grid techniques; finite volume flow solvers; the use of finite element analysis solvers for simulating coupled fluid/structural interaction problems; error quantification; and various applications of unstructured methods to complex geometries, including complete aircraft configurations. He has over 40 publications. Dr. Cavallo holds a doctorate in mechanical and aerospace engineering from Drexel University and he is an Associate Fellow of AIAA.

Dr. Richard DeLoach has been involved in testing technology at NASA Langley Research Center for over 40 years. He introduced the Modern Design of Experiments (MDOE) to Langley's experimental community and has taught the AIAA Short Course *Concepts in the Modern Design of Experiments*. During his time as chairman of the Standards Subcommittee of the AIAA Ground Testing Technical Committee, the subcommittee produced the guidelines for uncertainty assessment in physical experimentation for all AIAA journals. Dr. DeLoach is an Associate Fellow of AIAA and a recipient of the NASA Exceptional Technology Achievement Medal for pioneering the development and application of the modern design of experiments in aeronautical research, which revolutionized the wind tunnel testing process.

Dr. Andrew M. Eaton received his doctorate in mechanical engineering from Carnegie-Mellon University and has over 40 years of experience in multi-physics computational fluid dynamics modeling. He is currently Lead Engineer for the Fluid Dynamics Group at the Orbital ATK Propulsion Systems Division. As such, he provides direction and expertise primarily for simulating the complex multiphase, turbulent reactive fluid dynamic environments and performance required for the design of solid rocket motor propulsion systems, components, and manufacturing processes. Dr. Eaton has been involved with many projects supporting a variety of NASA, Air Force, and Navy programs.

Dr. Patrick Knupp is employed at Dihedral LLC, a research, development, and consulting firm. Previously, he was a distinguished member of the technical staff at Sandia National Laboratories. He has been involved in the field of verification and validation for the past 25 years, primarily with respect to methods for code verification. He learned the Method of Manufactured Solutions from his mentor, Dr. Patrick Roache. Dr. Knupp is co-author of the book *Verification of Computer Codes for Computational Science and Engineering* with Dr. Kambiz Salari. By training, Dr. Knupp is a computational mathematician, focusing on numerical methods for partial differential equations and mesh generation.

Dr. Marian Nemec is a Senior Research Scientist at Science & Technology Corporation, working in the Computational Aerosciences Branch of the NASA Advanced Supercomputing Division at Ames Research Center. He has 15 years of experience in the development of adjoint methods for discretization error analysis and shape optimization, embedded-boundary Cartesian mesh methods, adaptive mesh refinement, and the application of these methods to problems with complex geometry. Dr. Nemec was awarded the NASA Exceptional Public Achievement Medal for his contributions to computational aerodynamics. He obtained his doctorate from the University of Toronto Institute for Aerospace Studies.

Dr. Andrew Norris is an Aerospace Engineer with NASA's Hypersonic Air-Breathing Propulsion Branch at Langley Research Center. He has been involved with the X-51A scramjet project both in the return-to-flight investigations and in developing and maturing the ground-to-flight tools used in the design of hypersonic vehicles. Previously, he worked at the Ohio Aerospace Institute on the numerical modeling of combustion. Dr. Norris wrote the reacting part of the National Combustion Code, as well as working on the application of this code to gas turbine combustors. He wrote his doctoral dissertation on the subject of probability density function methods at Cornell University, studying under Dr. Steve Pope.

Dr. William Rider joined Sandia National Laboratory in 2007 after working at Los Alamos National Laboratory for nearly 18 years. During this time, he has authored more than 100 research papers and two books. His first book focused on high-resolution numerical methods for low-speed and incompressible flows; the second was a contributed volume on the topic of large eddy simulation. Currently, Dr. Rider works on method and code development for shock physics simulation as well as verification and validation of large-scale simulations. He received his doctorate in nuclear engineering from the University of New Mexico while working as a staff member at Los Alamos.

Dr. David Riggins is currently the Curators' Distinguished Teaching Professor of Aerospace Engineering at the Missouri University of Science and Technology. He has worked for over 30 years in hypersonic and high-speed propulsion systems, focusing on developing innovative approaches to propulsion and high-speed flight, and is a leading authority on the application of the second law of thermodynamics to the analysis of aerospace vehicles. Dr. Riggins has authored more than 70 papers, articles, and book chapters. He has served on numerous panels and working groups within both government and industry.

Dr. Vicente Romero is a Research Engineer in the Verification & Validation, Uncertainty Quantification, and Credibility Processes Department in the Engineering Sciences Center at Sandia National Laboratories. He has a modeling background in optical, thermal, fluid, and structural systems, specializing in complex coupled systems and applications where statistical or stochastic behavior is important. Dr. Romero also has extensive experience in developing and applying optimization and uncertainty quantification techniques for model calibration, validation, and risk assessment and reduction in nuclear weapon systems subjected to stressing thermal-mechanical-electrical-radiation environments. He has been with Sandia for 29 years.

Dr. Venke Sankaran serves as the Senior Research Scientist and Independent Researcher for modeling and simulation in the field of rocket propulsion at the Aerospace Systems Directorate, Air Force Research Laboratory, Edwards Air Force Base. He has authored more than 100 technical papers. Dr. Sankaran's research has been instrumental in the development of preconditioning algorithms for extending computational fluid dynamics methods to all flow regimes, including all Mach numbers, viscous-dominated regions, unsteady flows, generalized equations of state, and multiphase flows. His current research focuses on liquid rocket combustion instability prediction and on the development of advanced physical modeling and computational methodologies for solid, liquid, and electric propulsion systems.

Dr. Laura Swiler is a Distinguished Member of the Technical Staff in the Optimization and Uncertainty Quantification Department at Sandia National Laboratories. She has 22 years of experience in uncertainty quantification. Dr. Swiler's research includes sensitivity analysis for high-dimensional inputs; model validation incorporating both experimental and simulation uncertainty; use of surrogate or meta-models; calibration of model parameters in the presence of both experimental and model uncertainty; and non-probabilistic techniques to model epistemic uncertainty. She obtained her doctorate in Engineering and Public Policy from Carnegie Mellon University.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
11-07-2016	Technical	

4. TITLE AND SUBTITLE
Simulation Credibility
Advances in Verification, Validation, and Uncertainty Quantification

5a. CONTRACT NUMBER
NNM15AA02C

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Mehta, Unmeel B (Editor), Eklund, Dean R, Romero, Vicente J,
Pearce, Jeffrey A, and Keim, Nicholas S

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Ames Research Center
Moffett Field, CA 94035

Johns Hopkins University Energetics Research Group
10630 Little Patuxent Pkwy, Suite 202
Columbia MD 21046

8. PERFORMING ORGANIZATION REPORT NUMBER
JANNAF/GL-2016-0001

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
For Johns Hopkins University Energetics Research Group:
NASA/Marshall Spaceflight Center
Marshall Space Flight Center, AL 35812

10. SPONSOR/MONITOR'S ACRONYM(S)
NASA/MSFC

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
NASA/TP-2016-219422

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release. Distribution is Unlimited.

13. SUPPLEMENTARY NOTES
Document prepared by the Joint Army–Navy–NASA–Air Force (JANNAF) Interagency Propulsion Committee; Modeling and Simulation Subcommittee; Simulation Credibility — Uncertainty, Verification, Validation, and Risk — Panel. JANNAF Abstract Number 2016-0002.

14. ABSTRACT
Decision makers and other users of simulations need to know quantified simulation credibility to make simulation-based critical decisions and effectively use simulations, respectively. The credibility of a simulation is quantified by its accuracy in terms of uncertainty, and the responsibility of establishing credibility lies with the creator of the simulation. In this volume, we present some state-of-the-art philosophies, principles, and frameworks. The contributing authors involved in this publication have been dedicated to advancing simulation credibility. They detail and provide examples of key advances over the last 10 years in the processes used to quantify simulation credibility—verification, validation, and uncertainty quantification. The philosophies and assessment methods presented here are anticipated to be useful to all technical communities conducting continuum physics-based simulations; for example, issues related to the establishment of simulation credibility in the discipline of propulsion are discussed. We envision that simulation creators will find this volume very useful to guide and assist them in quantitatively conveying the credibility of their simulations.

15. SUBJECT TERMS
Simulation Credibility, Verification, Validation, Uncertainty Propagation, Uncertainty Quantification, Sensitivity Analysis, Design of Experiments, Computational Fluid Dynamics, Aerodynamics, Structural Mechanics, Propulsion.

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Unmeel Mehta
U	U	U	UU	530	19b. TELEPHONE NUMBER *(Include area code)* (650) 604-6744

Reset