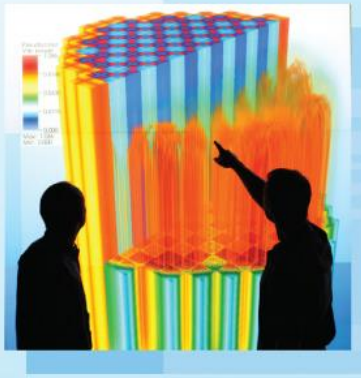
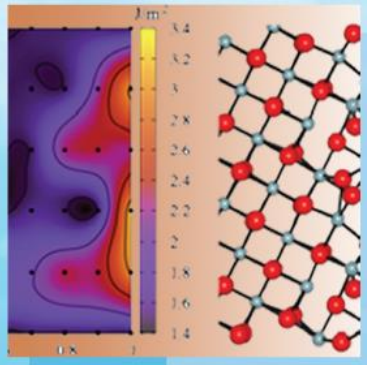
CASL-U-2013-0048-000

Power uprates
and plant life extension



Engineering design
and analysis

Science-enabling
high performance
computing



Fundamental science

Plant operational data

Consortium for Advanced Simulation of LWRs

L3.VUQ.SAUQ.P6.02; Apply VUQ Practices With Dakota to COBRA-TF for AMA-Relevant Problems

Vincent Mousseau
Brian Adams
Kostadin Ivanov
Maria Avramova
Robert Salko
Taylor Blyth
Sandia National Laboratories

April 2013:

U.S. DEPARTMENT OF ENERGY | Nuclear Energy

SANDIA REPORT

SAND2013-3093
Unlimited Release
Printed April 2013

CASL L3 Milestone Report: L3:VUQ.SAUQ.P6.02; Apply VUQ Practices with Dakota to COBRA-TF for AMA-Relevant Problems

Vincent Mousseau
Brian Adams
Kostadin Ivanov
Maria Avramova
Robert Salko
Taylor Blyth

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

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.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

CASL-U-2013-0048-000

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

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

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

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2013-3093
Unlimited Release
Printed April 2013

CASL L3 Milestone Report: L3:VUQ.SAUQ.P6.02; Apply VUQ Practices with Dakota to COBRA-TF for AMA-Relevant Problems

Vincent Mousseau
Multiphysics Simulation Technology, 1444
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-1322

Brian Adams
Optimization and UQ, 1441
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-1318

Kostadin Ivanov
Maria Avramova
Robert Salko
Taylor Blyth
Pennsylvania State University
University Park, PA 16802

Abstract

This report documents the completion of a CASL VUQ L3 milestone, whose goal was to exercise VUQ capabilities on AMA-relevant problems. It presents a simplified VUQ plan for the COBRA-TF code applied to the CIPS challenge problem. For the CIPS challenge problem the main contribution from COBRA-TF is to compute the boiling in a PWR fuel assembly. We present simplified verification, sensitivity analysis, calibration, validation, and uncertainty quantification for the COBRA-TF code, as applied to a 5x5 experimental PWR fuel assembly. Both manual and Dakota-based calculations are demonstrated. The experimental data for this study comes from the OECD benchmark for PWR Subchannel and Bundle Tests.

CASL-U-2013-0048-000

ACKNOWLEDGMENTS

This research was supported by the Consortium for Advanced Simulation of Light Water Reactors (http://www.casl.gov), an Energy Innovation Hub (http://www.energy.gov/hubs) for Modeling and Simulation of Nuclear Reactors under U.S. Department of Energy Contract No. DE-AC05-00OR22725. Sandia National Laboratories (SNL) is a core CASL partner.

The authors thank the reviewers of this report for their insights which have increased its quality; Laura Swiler, Jess Gehin, Jim Stewart, and Yixing Sung.

CONTENTS

1. INTRODUCTION 9
   1.1. Credible Software for Decision Making 9
   1.2. COBRA-TF and Coupled Application Problems 11
   1.3. Decomposition of Uncertainty 11

2. SOFTWARE QUALITY 13
   3.1. Linear Solvers 14
   3.2. CFL Number 15
   3.3. Iteration Tolerances 16
   3.4. Pseudo-Steady state Convergence Criteria 17

4. VALIDATION 19
   4.1. Data Uncertainty: Actual versus Theoretical 19
   4.2. Experimental Data 22
   4.3. Cobra-TF Results 22

5. CALIBRATION / VALIDATION 25
   5.1. Dakota Deterministic Calibration Study 26
   5.2. Dakota Calibration, then Validation 26
   5.3. Simultaneous Calibration to Multiple Data Sets 28

6. UNCERTAINTY QUANTIFICATION 30
   6.1. Verification of UQ Approach 30
   6.2. Validation of Sensitivity Analysis 31
   6.3. Replicate Analysis of UQ 33

7. SUMMARY AND FUTURE WORK 35
   7.1. Challenges 35
      7.1.1. Input Parameters 35
      7.1.2. Multiphysics Uncertainty Quantification 36
   7.2. Interactions With Other FA's 39
   7.3. Future Work 40

8. REFERENCES 42

APPENDIX A. DAKOTA VERIFICATION STUDY OF CFL VALUE 43

APPENDIX B. DAKOTA VERIFICATION STUDY OF LINEAR SOLVER CHOICE 46

APPENDIX C. DAKOTA VERIFICATION STUDY OF STEADY STATE CONVERGENCE CRITERIA 49

APPENDIX D. DAKOTA CALIBRATION EXAMPLE 52

APPENDIX E. DAKOTA SENSITIVITY AND UNCERTAINTY STUDIES: DETAILED RESULTS 53

DISTRIBUTION 56

FIGURES

Figure 1. Validation Study 23
Figure 2. Local Sensitivities versus Global Sensitivities 33
Figure 3. Multiphysics Uncertainty Quantification 36

TABLES

Table 1. Linear Solver Sensitivity 15
Table 2. CFL-controlled Time Step Sensitivity 16
Table 3. Time Step Convergence Criteria 16
Table 4. Steady State Convergence Criteria 17
Table 5. Experimental Uncertainty Deltas 20
Table 6. Repeated Experiment Void Fraction Data 20
Table 7. Experimental Uncertainty from the Repeat Experiment 21
Table 8. Void fraction Experimental Data 22
Table 9. Cobra-TF Void Fraction Results 22
Table 10. Void Fraction Differences between COBRA-TF and Experiment 24
Table 11. Calibration of COBRA-TF temperature to 6.1121 void fraction dataset 26
Table 12. Calibration of COBRA-TF temperature to 6.1122 void fraction dataset 27
Table 13. Prediction of void fraction, based on calibration 27
Table 14. Calibration of COBRA-TF temperature to five 6.xxx2 void fraction datasets 29
Table 15. Verification of Dakota SA/UQ 30
Table 16. Sensitivity Validation 32
Table 17. Sensitivity Errors 32
Table 18 Uncertainty Quantification of Wilks formula 34
Table 19. CFL verification for experiment 6.1121 43
Table 20. CFL verification for experiment 6.1122 43
Table 21. CFL verification for experiment 6.1451 44
Table 22. CFL verification for experiment 6.1452 44
Table 23. CFL verification for experiment 6.2441 44
Table 24. CFL verification for experiment 6.2442 44
Table 25. CFL verification for experiment 6.4561 45
Table 26. CFL verification for experiment 6.4562 45
Table 27. CFL verification for experiment 6.6561 45
Table 28. CFL verification for experiment 6.6562 45
Table 29. Linear solver verification for experiment 6.1121 46
Table 30. Linear solver verification for experiment 6.1122 46
Table 31. Linear solver verification for experiment 6.1451 46
Table 32. Linear solver verification for experiment 6.1452 47
Table 33. Linear solver verification for experiment 6.2441 47
Table 34. Linear solver verification for experiment 6.2442 47
Table 35. Linear solver verification for experiment 6.4561 47
Table 36. Linear solver verification for experiment 6.4562 47

Table 37. Linear solver verification for experiment 6.6561 47
Table 38. Linear solver verification for experiment 6.6562 48
Table 39. Steady state convergence criteria verification for experiment 6.1121..... 49
Table 40. Steady state convergence criteria verification for experiment 6.1122..... 49
Table 41. Steady state convergence criteria verification for experiment 6.1451..... 50
Table 42. Steady state convergence criteria verification for experiment 6.1452..... 50
Table 43. Steady state convergence criteria verification for experiment 6.2441..... 50
Table 44. Steady state convergence criteria verification for experiment 6.2442..... 50
Table 45. Steady state convergence criteria verification for experiment 6.4561..... 50
Table 46. Steady state convergence criteria verification for experiment 6.4562..... 51
Table 47. Steady state convergence criteria verification for experiment 6.6561..... 51
Table 48. Steady state convergence criteria verification for experiment 6.6562..... 51
Table 49. Scaled perturbation ΔT applied to each experiment 53
Table 50. Statistics for five experimental configurations, 59 samples. 53
Table 51. Statistics for five experimental configurations, 590 samples. 54
Table 52. Statistics for five experimental configurations, 5900 samples. 54
Table 53. Partial correlation of void fractions for five experiments with ΔT..... 55

NOMENCLATURE

ITEM	DESCRIPTION
AMA	Advanced Modeling Applications (CASL technical focus area)
CASL	Consortium for Advanced Simulation of LWRs
CFL	Courant–Friedrichs–Lewy
CILC	Crud-induced localized corrosion
CIPS	Crud-induced power shift
CPI	Challenge Problem Integrator
CTF	Cobra-TF
CRUD	Chalk River Unidentified Deposit
DAKOTA	Design and Analysis ToolKit for Optimization and Terascale Applications
DNB	Departure from nucleate boiling
DOE	Department of Energy
DTK	Data transfer toolkit.
EPRI	Electric Power Research Institute
FTE	Full Time Equivalent (one man year)
FOM	Figure Of Merit
HPC	High Performance Computing
L3	Level 3 (milestone)
LANL	Los Alamos National Laboratory
LWR	Light water reactor
MAMBA	MPO Advanced Model for Boron Analysis
MPACT	Michigan Parallel Characteristic based Transport
NRC	Nuclear Regulatory Commission
OECD	Organization for Economic Co-operation and Development
PNNL	Pacific Northwest National Laboratory
PSBT	PWR Subchannel and Bundle Tests
PSU	Pennsylvania State University
PWR	Pressurized water reactor
SA	Sensitivity analysis
SNL	Sandia National Laboratories
SQA	Software Quality Assurance
SUSA	System for Uncertainty and Sensitivity Analysis
UQ	Uncertainty quantification
TVA	Tennessee Valley Authority
VERA	Virtual Environment for Reactor Applications
VRI	Virtual Reactor Integration (CASL technical focus area)
VUQ	Validation and Uncertainty Quantification (CASL technical focus area)
WEC	Westinghouse Electric Company LLC

1. INTRODUCTION

1.1. Credible Software for Decision Making

There are a variety of reasons for writing software, including, but not limited to:

1. To provide a tool for someone else to use to make an important decision that impacts financial or safety concerns,
2. To demonstrate parallel scalability on HPC platforms,
3. To demonstrate new algorithms, or
4. To generate results for journal papers.

Each of these motivations for writing software gives rise to different validation and uncertainty quantification (VUQ) requirements. In this document, we will focus on the first reason listed above – providing a tool for customers– which aligns with the Consortium for Advanced Simulation (CASL) goal of delivering software to industry. We have blended software quality assurance (SQA) and VUQ because they are both relevant to the goal, and can be used to answer a basic question:

> "*How confident are we that someone else can use this software and make a quality decision based on the output*?"

The sections in this document outline a process to build a case that answers that basic question. It consists of the following steps:

1. SQA – Confidence that the software has an acceptably small number of bugs *as proved by supporting testing and documentation*.
2. Verification – Confidence that the algorithms and models in the code are implemented as intended, *as proved by supporting testing and documentation*.
3. Validation – Confidence that we are solving the correct equations, *as proved by supporting testing and documentation*.
4. Uncertainty Quantification – Confidence that the error bars placed on the code output are accurate enough for a quality decision to be made, *as proved by supporting testing and documentation*.

There are important challenges to be aware of in validation of nuclear reactor simulation codes. Because of the complex, nonlinearly coupled, multiphysics modeling required for simulating a nuclear reactor, the correct model form is not always obvious. What equations to solve or more precisely what physics is important and needs to be included in the model and what physics is not important and can be excluded from the model is not intuitively obvious. Often one needs to acquire experimental data to determine if the correct physics is included in the model. This helps to answer the question, "Are we solving the correct equations?"

Additionally experimental data is often not available at the pressures or energies that are typical of operating conditions for a PWR. Because of this one has to scale the experimental data from the values that you have to the values that you need. This scaling of information is done by the simulation code. This leads to an uncertainty in the results that comes from validating the simulation code at one point in state space and then applying it in another. The ability of a

nuclear reactor simulation code to scale its validation database has been a highly contended issue based on the nonlinearity of the system.

In this document, we demonstrate what a VUQ plan looks like. The purpose is not to provide a detailed VUQ plan, but to simply to demonstrate most of the steps. Another important point to make is that this VUQ process is an iterative process. Although there is a logical flow of continually increasing confidence which depends on the previous steps, each new equation, model, or algorithm, requires the process to be repeated; SQA then Verification, then Validation, then Uncertainty Quantification. This provides two important pieces of evidence:

1. Confidence in the new software (equation, model, or algorithm).
2. Confidence that the new software has not had a negative impact on the previous software.

Specifically number 2 in the list above refers to the unintentional changes in software that can occur when adding a new model that has a negative impact on the solution technique. This is especially important with poorly documented legacy code where concepts of modularity and data encapsulation may have eroded over time or never existed in the first place.

Recognizing that this is an iterative process, it is important to automate the testing and documentation as much as possible. This ensures that when the package is delivered to the customer, they get the software, the documentation, and the testing. Enabling the customer to reproduce the testing and the results presented in the documentation greatly increases customer confidence in the software. In addition, this provides a large number of examples regarding how to use the software.

From a high level view, the computer science algorithm is the same for verification, validation, calibration, and uncertainty quantification. All four of these do the same five basic steps:

1. Run the code.
2. Extract a FOM.
3. Obtain a different FOM.
4. Calculate a norm between the two FOM's.
5. Modify the input and go back to step 1.

Our vision and goals are to improve the automation of the VUQ process, by using the Dakota software not only for uncertainty quantification, but also for the verification, validation, and data-based parameter calibration. Such studies will be based on a single, robust interface with the Virtual Environment for Reactor Applications (VERA) common input system. This will improve consistency and reliability of analyses by standardizing the most common VUQ workflows around CASL simulations.

All of the sensitivity, verification, and validation studies reported herein were conducted both manually for limited experiments, and automatically with Dakota for a broader set of experiments. The calibration and uncertainty quantification results are based on Dakota studies. All Dakota input files, CTF application templates, run scripts, and Dakota output are archived in the VERA Git repository at casl-dev:/git-root/VUQDemos/cobra_tf/psbt_series6.

1.2. COBRA-TF and Coupled Application Problems

This document will focus on using experimental data from the OECD benchmark based on the PWR Subchannel and Bundle Tests (PSBT) [3]. The two codes that will be included will be COBRA-TF (CTF) [1], configured to simulate the 5x5 bundle benchmark experiment, coupled with Dakota [2] for VUQ methods. The PSBT tests provide PWR-relevant data to assess CTF. We will assume for concreteness that a decision needs to be made based on the void fraction at three different locations in the fuel assembly:

1. Low .....2.216 meters
2. Mid.....2.669 meters
3. High.....3.177 meters

It is important to note that Low, Medium, and High are three different locations in the assembly not different levels of void fraction. This becomes more confusing because the void fraction is lowest in the Low region of the assembly and highest in the High region of the reactor.

For this study, the FOM is void fraction at these locations, either as calculated by CTF, or from experimental data. A representative VUQ goal is to provide a confidence interval (error bars) around that FOM so a decision can be made based on the void fraction in the fuel assembly.

This VUQ study is a step towards a VUQ plan for coupled physics simulation of the CASL challenge problem CRUD-Induced Power Shift (CIPS). The current plan is to use VERA-coupled

Dakota + MPACT + CTF + MAMBA

The "heat" will be computed by MPACT and passed to CTF which will compute the boiling which will be passed to MAMBA. In this study we are assessing CTF's ability to compute boiling which is its key contribution to the CIPS milestone. Specifically, CIPS is based on subcooled boiling which occurs in the lower part of the assembly in the PBST experiments. Some challenges and alternate approaches related to exercising VUQ processes on such coupled physics models are discussed in the conclusions.

1.3. Decomposition of Uncertainty

Note that in uncertainty quantification we are assuming that the software has gone through SQA, verification, and validation before uncertainty quantification begins. That means that we have assumed that the correct partial differential equations are solved and that we do not need to address uncertainty do to not solving the correct partial differential equation.

Another perspective on the VUQ plan is offered by this simple equation for the FOM confidence interval:

$$CI_{FOM} = FOM \pm \left(\frac{\partial FOM}{\partial Y}\right)\left(\frac{\partial Y}{\partial p}\right)\Delta p\,. \tag{1.1}$$

Equation 1.1 gives the confidence interval of the figure of merit, CI_{FOM} , in terms of the predicted value of the FOM and its uncertainty. The uncertainty in the FOM is represented as

the product of three terms, the dependency of the figure of merit on the solution "Y", the dependency of the solution on a parameter "p", and the distribution of values for "p" represented symbolically as Δp.

The first sensitivity term represents the dependence of the FOM on the physics. The solution "Y" comes from solving conservation laws, like mass momentum and energy. Verification focuses on the first term. The question is, "Are we accurately solving the Partial Differential Equations?" We need to know that we have an accurate value of the solution before we begin to look for model dependencies. Inherent in this is the assumption that we are solving the "correct" PDE.

The next term represents the dependency of the solution on model parameters. These are the closure laws employed by the code, e.g., wall friction, wall heat transfer. Closure laws are empirically based and confidence in a closure law comes from validation data.

It is important to note that the closure law's impact on the FOM uncertainty shows up as a derivative. Not only should the closure law predict the correct value, but it should also predict the sensitivity, $s = \frac{\partial Y}{\partial p}$.

A validation study should not only gather data on the values but also on the derivative of the value with respect to the parameter.

The final term represents the distribution of the model parameter. This distribution can come from limiting arguments, expert opinion, or by attempting to measure this distribution in an experiment.

2. SOFTWARE QUALITY

Software quality is the foundation of, and first step in, building any uncertainty quantification evidence case. Developers must be confident that solution variations are due to physics and not bugs in the software.

COBRA is a family tree of codes that have evolved from the original COBRA from PNNL about 30 years ago. The challenge for CASL is that although there has been extensive software quality and assessment work (assessment is a combination of verification and validation) done on the COBRA family tree, we have to present evidence for software quality, verification, validation and uncertainty quantification of our branch of the COBRA family tree, COBRA-TF.

COBRA-TF is currently developed and supported by both Penn State and CASL. The CASL version of COBRA-TF originated at Pacific Northwest National Laboratory (PNNL) and arrived by way of Westinghouse. Like many of the codes currently employed by the nuclear power industry, COBRA started at a DOE lab and then was later adapted by industry. Over this 20 plus years of evolution from PNNL to Westinghouse to Penn State, COBRA-TF has been verified and validated (or at least assessed) numerous times.

The assumption for the CASL version of COBRA-TF is that software quality has been done at the previous organizations. This software quality can be inherited and CASL is only responsible for software quality of its changes to the base code. The COBRA-TF software package includes a well-written "CTF Theory Manual" [1] and a user manual "CTF – A Thermal-Hydraulic Subchannel Code for LWRs Transient Analysis: User's Manual" [4]. Discussion with Robert Salko (Pennsylvania State University [PSU]) from the VRI team, have helped improve the "CTF Theory Manual" by refining the discussion of the pseudo-steady state option in COBRA-TF.

3. VERIFICATION

Verification provides evidence for confidence in later validation and uncertainty quantification. The confidence comes from evidence that equations are accurately solved accurately and knowing that changes in the solution are due to physics and not errors in the solution algorithm. Traditionally this includes mesh convergence, time step convergence, and iteration tolerance convergence. Mesh convergence has been omitted from this study, due to the nature of CTF. Solution parameters and time step convergence (through a CFL study) are included, even though this is a steady state problem. Note that verification is a math exercise. The question being asked is, "Are you solving the equations accurately?" Verification does not address the correctness or efficiency of the solution method.

There are a variety of verification methods that apply to new software like the method of manufactured solutions and automatic differentiation methods. For legacy codes like COBRA-TF the solution method and the conservation equations have blended together over time. The result is it is often difficult to determine the exact form of the conservation equation being solved. Therefore verification methods that require detailed knowledge of the form of the conservation equations, like the method of manufactured solutions, are not easily applied. It is important to recall that modern transient nonlinear solvers (from Trilinos and PETSc) did not exist in the 80's when COBRA was written.

Four verification studies were based on solver parameters. These studies do not include the breadth or the quality required to undertake a real VUQ study, but they serve as an example of the type of work that should be completed. These studies focused on the FOM to be used for validation and uncertainty quantification. For this study, the FOM is the average void fraction around a specific "fuel pin." This is computed as the average of the four flow channels that surround the center fuel pin in the 5x5 bundle (it is interesting to note that these four fuel channels are nearly perfectly symmetric in the simulation code and therefore often give four copies of the same answer). We consider the verification study converged when the average void fractions at the three assembly locations (Low, medium, and High) are the same to four digits.

Traditional mesh convergence studies are not possible in COBRA-TF. Based on conversations with Kostadin Ivanov (PSU) and Maria Avramova (PSU) of the VRI team, COBRA-TF only works for two geometries. The two configurations are one flow channel per assembly and one flow channel per fuel pin. Since the mesh is fixed in the plane perpendicular to the axial direction, it is not clear what one should expect from a mesh convergence study in the axial direction. Hence, the problem nodalization was fixed and "mesh convergence" was not addressed.

The studies in this section are for a single experimental configuration, but are representative of all configurations considered. For examples of automated, Dakota-driven verification studies for all ten series 6 experiments of interest, see Appendices B, C and D.

3.1. Linear Solvers

There are five different linear solvers available in COBRA-TF [4]:

1. A direct solve using Gaussian elimination

2. BSGS + ILUT
3. GMRES
4. BSGS
5. GMRES + ILUT

All five solvers were exercised. The fifth item, GMRES + ILUT, failed. The other four converged and gave the same steady state answer as measured by the average void fraction around the center fuel pin in a 5x5 fuel assembly. Therefore, the sensitivity to the linear solver is zero as long as it produces an answer. This provides confidence that the condition numbers on the matrix are reasonable, and that convergence tolerances on the linear solvers are acceptably tight.

Table 1 shows the average void fraction around the central pin computed from COBRA-TF for experiment 6.6562. The value of the average void is given at three different locations, 2.216 meters (lower), 2.669 meters (middle), and 3.177 meters (high). Data is presented for four different linear solvers. All four answers are the same for all three bundle heights. The base value for linear solver (that is the one used for most of these studies) is highlighted in the fourth row of Table 1.

Table 1. Linear Solver Sensitivity

Solver	Lower	Middle	High
Direct	0.4110	0.6720	0.7625
BSGS+ILU	0.4110	0.6720	0.7625
GMRES	0.4110	0.6720	0.7625
BSGAS	0.4110	0.6720	0.7625

Note that the robustness of COBRA-TF was never an issue in any of these studies. We ran the code thousands of times and only had one code "crash," noted in the appendices. It is important to note that these void fractions are much higher than one would see in a PWR in standard operating mode. Void fractions this high would only occur for during accident scenarios.

3.2. CFL Number

Since this is a steady state problem, one would expect the problem to be insensitive to transient fluid stability constraints like the Courant–Friedrichs–Lewy (CFL) number. However, since CTF does not have a steady state option, a transient is run until the problem reaches steady state as defined by the pseudo-steady state convergence criteria. Therefore, the steady state solution may depend on the path taken by the transient to reach steady state**.**

Table 2 presents the results of four different choices for the CFL number for experiment 66562, run with COBRA-TF. This reveals that the solution is sensitive to CFL numbers greater than one, and a slight sensitivity in the average void fraction at the high level for the base case, which is highlighted in the table, below. This sensitivity indicates that the studies should be rerun with a CFL number of 0.4 to present solver independent solutions.

Table 2. CFL-controlled Time Step Sensitivity

CFL	Lower	Middle	Upper
1.6	0.4090	0.6690	0.7760
0.8	0.4110	0.6720	0.7625
0.4	0.4110	0.6720	0.7610
0.2	0.4110	0.6720	0.7610

Note that the COBRA-TF manual suggests using a CFL number of 3.0 for steady state runs. This is currently being fixed by Robert Salko of the VRI team. Results here are more consistent with the idea that since the stability limit is a CFL number of 1.0, conservatism normally requires the CFL number to be less than one half. In this study we are measuring the sensitivity of the input parameters separately. Different combinations of other input parameters may result in a converged solution at higher CFL numbers.

3.3. Iteration Tolerances

Since this is a steady state problem, there is no expected dependence on the path used to reach the steady state value, although minor differences may be possible. This study focused on the convergence criteria for each individual time step. Three parameters were studied, but one (IITMAX) was later determined to be dependent on a second parameter (NSIM) that was not changed. When NSIM = 1, as it is in all of these input decks, the problem does not depend on IITMAX at all. Note that this dependency between different input parameters should be a part of the VERA common input processor. The key here is that COBRA-TF inputs are interdependent. However, it is implemented in software, it is important that the VERA common input processor mark inconsistent input as an error. That way in future uncertainty studies, one will not bother changing input parameters that are overwritten by the code. The other parameters studied were the nondimensional outer iteration tolerance, EPSO, and the maximum number of outer iterations, OITMAX.

The solution as measured by the FOM is insensitive to all tolerances studied. Table 3 shows the average void fraction around the central pin for the 66562 experiment. Because NSIM=1 in the input decks IITMAX= 1, and the input number shown in column 3 of the table (IITMAX) is irrelevant. Significant increases in the outer iteration (the pressure iteration) convergence tolerance (EPSO) and the maximum number of outer iterations (OITMAX) had no impact on the solution. The baseline values are highlighted in the table, below.

Table 3. Time Step Convergence Criteria

EPSO	OITMAX	IITMAX	Lower	Middle	Upper
0.0005	10	80	0.4110	0.6720	0.7625
0.001	5	40	0.4110	0.6720	0.7625
0.002	2	20	0.4110	0.6720	0.7625
0.004	1	10	0.4110	0.6720	0.7625
0.008	1	5	0.4110	0.6720	0.7625
0.016	1	2	0.4110	0.6720	0.7625

3.4. Pseudo-Steady state Convergence Criteria

Because COBRA-TF does not have a steady state option, it has to solve a transient and run the transient until steady state is achieved (for more detail see the end of Chapter 2 in the CFT theory manual[1]). Four criteria are used by COBRA-TF to determine if steady state has been achieved:

1. Heat Balance: .....The sum of the energy sources and sinks in the system,
2. Mass Balance: .....The sum of the mass sources and sinks in the system,
3. Heat % Deviate: .....Time rate of change of the energy in the system, and
4. Mass % Deviate: .....Time rate of change of the mass in the system.

Unfortunately, these steady state convergence criteria parameters are not accessible through the COBRA-TF input processor. Therefore, the source code had to be modified for this study. It is important to note that for legacy codes like COBRA-TF, parameters that are important for uncertainty quantification are not exposed through the input. Therefore the user has to modify the source code to gain access to these important parameters. This change in the source code has implications to software quality. The code that is being tested is not the same code as the one that has the software quality pedigree. If time and budget permitted, one could reproduce the software quality pedigree of the modified code. Otherwise, whenever possible, the parameters that are important for uncertainty quantification should be exposed through the input processor.

Table 4 presents the results from the sensitivity study on the steady state convergence criteria. All of these numbers are nondimensionalized percentages relative to the inflow mass and energy. Note that all four parameters are modified by the same scaling factor. The average void fractions at the lower and middle regions are insensitive to the steady state convergence criteria. However, the upper average void fraction is sensitive. Note that the FOM has converged for the convergence criteria for the nominal case, which is highlighted in the table, below.

Table 4. Steady State Convergence Criteria

Heat Balance	Mass Balance	Heat % Deviate	Mass % Deviate	Lower	Middle	Upper
2.0	2.0	0.04	0.04	0.4100	0.6710	0.7605
1.0	1.0	0.02	0.02	0.4100	0.6720	0.7615
0.5	0.5	0.01	0.01	0.4100	0.6720	0.7625
0.25	0.25	0.005	0.005	0.4100	0.6720	0.7625
0.125	0.125	0.0025	0.0025	0.4100	0.6720	0.7625

Note that the temporal pseudo-steady state convergence criteria are signed quantities. This allows small numbers to be created from cancellation of large errors that are equal and opposite in sign. Also noteworthy is the experimental pseudo-steady state convergence criteria are currently unknown. Thus, it is not clear that the experimental definition of pseudo-steady state convergence criteria and the COBRA-TF definition of pseudo-steady state convergence criteria are consistent. Discussions of the pseudo-steady state convergence criteria with Robert Salko of the VRI team have resulted in modifications to the "CTF Theory Manual" [1] to better describe these criteria. We are also working with VRI to improve the sensitivity of the pseudo-steady state convergence criteria.

CASL-U-2013-0048-000

4. VALIDATION

This section discusses the comparison of experimental data with the code results. It is important to account for both errors in the experimental data as well as the error in the code results. Quite often in the nuclear power industry, the experimental data available is not at the same pressure, temperature, and energy where the code will be applied. This scaling effect adds another level of experimental uncertainty, and must be understood. There is also another level of experimental error that impacts validation data. For experimental data to be used for empirical correlations (like those employed in COBRA-TF) it is important to capture the correct geometry. However, this "correct" geometry often limits the quality of the data that can be acquired. Experimental data for validation of first principle codes, like Hydra, is often done in very simplified geometry where detailed experimental measurements are possible. The simulation code, Hydra, is then required to scale the data to the correct geometry. Validation data is an exercise in compromise and it is important to understand what assumptions (compromises) were made by the experimentalist before the data is used to validate a code. It is important to note here that the PSBT data is more relevant to CILC or DNB than CIPS which occurs at very low void fractions.

4.1. Data Uncertainty: Actual versus Theoretical

One of the first steps in any validation study is to "vet" the data. This means that one needs to assess the completeness and the quality of the experimental data. We did not find any problem with the completeness of the data mainly because the CTF input decks and data existed in the CTF repository. In this study we assumed that the data collection and that the input deck preparation already done by the VRI team were correct. Spot checks of this did not reveal any problems. In a "real" VUQ study there would be a SQA procedure to document and test that the building of the COBRA-TF input decks was done correctly.

The next step is to apply error bars to the data. We need to compare the experimental data and its uncertainty to the CTF simulation data and its uncertainty. The reported void fraction experimental error indicates that it is accurate to 0.04. There is some confusion here if accurate to 0.04 means plus/minus 0.02 or plus/minus 0.04. If we had better documentation of the calculation of the experimental error we would know whether 0.04 represented one standard deviation or two. We chose the conservative value of plus/minus 0.04. This experimental error uncertainty came from the manufacturer of the x-ray densitometer that was used to measure void fraction in this study. During discussions with VRI team member Maria Avramova and VUQ team member Nam Dinh, we learned that x-ray densitometers do not measure the density near a solid wall. In this experiment it means that the void fraction near the fuel pin is not measured. This indicates a higher uncertainty in low void fractions where the experiment would under predict the void fraction. Here it is important to note that in subcooled boiling the entire void is generated near the fuel pin. Similarly the experiment over predicts the void fraction for high values of void fraction. That is the experiment would not measure water films on the fuel pin.

Upon further discussion with Maria Avramova, we found out that as part of the PSBT benchmark, there was a repeat of an experiment. One of the best ways to measure the uncertainty in the experimental data is to run the experiment once, run other experiments, and then try to repeat the same experiment. The difference between the two experimental data sets, which are designed to be the same, provide an "actual" (not theoretical) experimental uncertainty

measurement which accounts for inter-experiment or replicate variability. Note that this replicate process cannot reveal systematic biases like under-predicting low void fractions and over-predicting high void fractions.

Unfortunately for this report, this information came late. Earlier in the work, we chose to use data from PSBT series 6. The experimental repeat came from first using series 5, and then series 5 data was recomputed in series 8. Although this measurement of experimental uncertainty does not directly apply to our series 6 data, its serves as an estimate of experimental uncertainty (using series 5 as a surrogate for series 6).

Table 5 shows the differences in the input parameters of the two experiments. This set of nine experiments were chosen from the total set of 31 experiments based on having the smallest differences in pressure, mass flow rate, power, and inlet temperature.

Table 5. Experimental Uncertainty Deltas

Experiment	Delta P kg/cm^2 a	Delta m 10^6 kg/ m^2 hr.	Delta Q kW	Delta T K
1342	0.07	0	6.7	0.3
1452	0.01	0.06	4.2	0.5
2231	0.74	0.09	7.1	0.2
2232	1.27	0.01	0.7	0.1
2351	0.19	0.01	4.2	0.2
2352	0.37	0.01	8.1	0.3
2452	0.56	0.03	1.6	0.7
3442	0.17	0.02	7.7	0
6551	0.13	0.08	2.8	0.5

Table 6 presents the void fractions from series 5 and the repeat of series 5 in series 8. Quick observation notes that there is discrepancy between these two data sets.

Table 6. Repeated Experiment Void Fraction Data

Experiment	5 Low assembly location	5 Mid assembly location	5 High assembly location	8 Low assembly location	8 Mid assembly location	8 High assembly location
1342	0.0700	0.2304	0.4147	0.1297	0.3358	0.4588
1452	0.1233	0.2844	0.4709	0.2056	0.3983	0.5287
2231	0.0085	0.0927	0.2937	0.0000	0.0925	0.3043
2232	0.1474	0.3236	0.4634	0.1876	0.3745	0.4682
2351	0.1177	0.2832	0.4186	0.1718	0.3464	0.4316
2352	0.2534	0.3732	0.4884	0.3016	0.4280	0.5110
2452	0.2811	0.3952	0.5121	0.3078	0.4471	0.5383
3442	0.1161	0.3499	0.5803	0.1451	0.4500	0.6088
6551	0.0393	0.4011	0.6921	0.0062	0.3537	0.6645

Table 7 gives the absolute and percent uncertainty measure by the difference in void fraction. The data values highlighted in the table below have absolute errors greater than 0.04 or percent errors greater than 20%. This provides a measure of the experimental uncertainty and clearly shows that one should have more confidence in the High void fraction measurements and a less confidence in the Low void fraction measurements. It is important to note here that subcooled boiling, which has the largest impact on CIPS, takes place in the Low data region. Note that this small void fraction is still larger than normal PWR operating conditions.

Table 7. Experimental Uncertainty from the Repeat Experiment

Experiment 5 vs. 8	Low assembly location absolute difference	Low assembly location percent difference	Mid assembly location absolute difference	Mid assembly location percent difference	High assembly location absolute difference	High assembly location percent difference
1342	0.0597	46.03	0.1054	31.39	0.0441	9.61
1452	0.0823	40.03	0.1139	28.60	0.0578	10.93
2231	-0.0085	100.00	-0.0002	-0.22	0.0106	3.48
2232	0.0402	21.43	0.0509	13.59	0.0048	1.03
2351	0.0541	31.49	0.0632	18.24	0.0130	3.01
2352	0.0482	15.98	0.0548	12.80	0.0226	4.42
2452	0.0267	8.67	0.0519	11.61	0.0262	4.87
3442	0.0290	19.99	0.1001	22.24	0.0285	4.68
6551	-0.0331	-84.22	-0.0474	-13.40	-0.0276	-4.15

This section provides a simple example of how experimental data used for validation and/or calibration should be "vetted." It is important to understand enough about the experimental equipment and how the experiment was run, in order to determine if the experiment is relevant for the validation needed for a given challenge problem. In addition, it is important to have a firm understanding of the quality of the data to understand how it should be used in the calibration and validation process relative to other data sets. As a rule of thumb, if the experimental uncertainty is represented as a single number (like 0.04 for void fraction) then you do not have the information you need.

In a "real" VUQ study we would require the "raw" data and SQA on the processes that convert "raw" data to the experimental results presented. This would include a detailed description of how the experimental uncertainty (error bars) were computed.

A variety of ways to measure experimental uncertainty are preferred, and it is always important to know that experimental uncertainty varies in space and in time. For example, if we are interested in data to validate subcooled boiling, then maybe we do not want to use experimental measurements like x-ray densitometry that cannot measure the void fraction "near" the surface. However, if this is the "best" data that you have, then you need to figure out a way to "interpret" the experimental data so it can be best employed in validation and calibration.

4.2. Experimental Data

The experimental data for series 6 of the PSBT study are presented in Table 8. For more detail see reference [3], specifically Table 4.2.2.2 on pages 62 and 63.

Table 8. Void fraction Experimental Data

Name	Inlet temperature	Low assembly location	Mid assembly location	High assembly location
61121	301.5	0.0000	0.0055	0.1011
61122	306.7	0.0000	0.0699	0.2394
61451	267.4	0.0000	0.0606	0.2350
61452	272.5	0.0000	0.1237	0.3019
62441	233.5	0.0000	0.0902	0.3052
62442	228.4	0.0000	0.1491	0.3690
64561	192.6	0.0015	0.1430	0.3757
64562	213.0	0.1268	0.4376	0.5636
66561	144.0	0.1174	0.4753	0.6882
66562	158.5	0.3703	0.6634	0.7799

4.3. Cobra-TF Results

We ran COBRA-TF at the same pressures, temperatures, and mass flow rates as the PSBT experiment and produced the results in Table 9. It is interesting to note that the void fraction in COBRA-TF is only output to three digits. Because of the symmetry of the problem, the average void fraction around the center pin is simply the average of the four symmetric channels and almost all of the time the four channels have the same void fraction to three digits in steady state. Occasionally the void fractions are different due to numerical reasons (see set 66562 at the high assembly location in Table 9).

Table 9. Cobra-TF Void Fraction Results

Name	Inlet temperature	Low assembly location	Mid assembly location	High assembly location
61121	301.5	0.0040	0.0770	0.1470
61122	306.7	0.0510	0.1600	0.2330
61451	267.4	0.0080	0.0990	0.2270
61452	272.5	0.0190	0.1640	0.2980
62441	233.5	0.0100	0.1300	0.2860
62442	228.4	0.0260	0.1940	0.3550
64561	192.6	0.0460	0.2870	0.4960
64562	213.0	0.2580	0.5260	0.6500
66561	144.0	0.1830	0.5520	0.6920
66562	158.5	0.4110	0.6720	0.7625

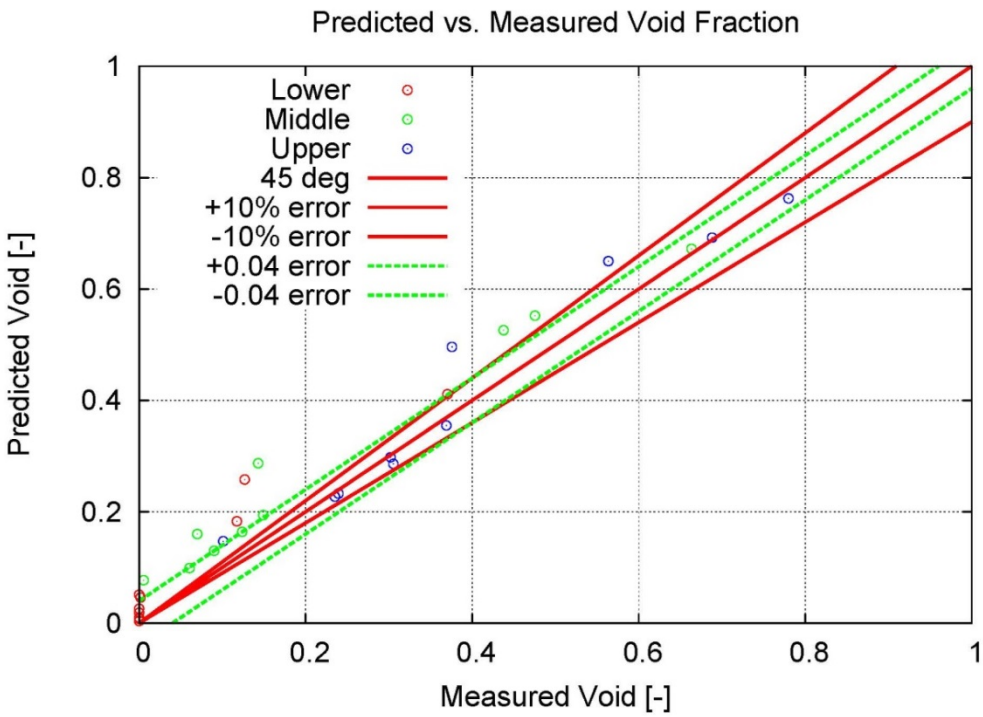


Figure 1. Validation Study

Figure 1 shows a plot of the validation study. The green dashed lines are the experimental error of 0.04 (plus or minus). The red lines are plus 10%, exact, and minus 10%. The red lines assume that the void fraction measurement error is smaller for low void fractions. This contrasts the green lines which assume that COBRA-TF or the experiment may give negative void fractions. Both the red and green lines are assumptions and are include in the plot to help interpret the data. We note that the experiment and the code differ where the experiment predicted zero void fraction (large number of red circles in the lower left corner) which may be caused by the experiments inability to correctly measure subcooled boiling.

The data plotted on Figure 1 (above) is shown in Table 10 (below). We have highlighted differences greater than 0.04 (the theoretical experimental uncertainty) or greater than 20%. It is important to note that most of the data in the lower and middle regions are off by more than 20%. Because of the lower quality of COBRA-TF's ability to match the experimental data one would question the value of continuing on with uncertainty quantification. Based on the results of Table 10 at this point one may choose to investigate different models or to use some of the PSBT data to calibrate the parameters in COBRA-TF.

On the other hand, based on the impact on the FOM for the challenge problem being studied, these results may be determined to be acceptable.

Table 10. Void Fraction Differences between COBRA-TF and Experiment

Name	Low Assembly Location Absolute Delta	Low Assembly Location Percent Delta	Mid Assembly Location Absolute Delta	Mid Assembly Location Percent Delta	High Assembly Location Absolute Delta	High Assembly Location Percent Delta
61121	.0040	100	.0715	93	.0459	31
61122	.0510	100	.0901	56	-.0064	-3
61451	.0080	100	.0384	39	-.0080	-4
61452	.0190	100	.0403	25	-.0039	-1
62441	.0100	100	.0398	31	-.0192	-7
62442	.0260	100	.0449	23	-.014	-4
64561	.0445	97	.1440	50	.1203	24
64562	.1312	51	.0884	17	.0864	13
66561	.0656	36	.0767	14	.0038	0.6
66562	.0407	10	.0086	1	-.0174	-2

5. CALIBRATION / VALIDATION

This section includes simple examples to demonstrate how data-based calibration can be used to adjust or "tune" parameters to improve validation and lower uncertainty. For the application at hand, we demonstrate the capability with a contrived example, acknowledging it is distant from a realistic physical application. In this document we aim to present and exercise the steps in a VUQ plan, not deliver a complete VUQ study for COBRA-TF applied to CIPS.

For our calibration examples, we will fix non-calibrated parameters at experimental configuration values, for example the pressure = 164 $\left(\frac{kg}{cm^2 a}\right)$, the power = 37 (kW), and the mass flow rate mdot=10 $\left(10^6 \frac{kg}{m^2 \bullet hr}\right)$. The unit of pressure is a little unique and is defined precisely by the following equation.

$$1\frac{kg}{cm^2 a} = \begin{cases} 0.980665 & \text{bar} \\ 14.223 & \text{psi} \end{cases} \tag{1.2}$$

We will then calibrate the inlet temperature(s) to best match the measured void fractions for one or more experiments, using Dakota's local, gradient-based NL2SOL solver. Most calibrations conducted required 10 to 15 COBRA-TF executions, orchestrated by Dakota. It is important to note that we are only calibrating one parameter (inlet temperature) and that calibration of more parameters will require more than 15 runs of COBRA-TF.

In a more realistic study we would likely determine the most important parameters lie in the wall heat transfer, boiling, and inter-channel mixing closure laws. These would need to be exposed through the input processor where they could be manipulated by Dakota. It would probably be best to keep the parameter input separate from the normal input file that the code user sees. There are parameters that are important for uncertainty quantification that you do not want to be adjusted by a code user. It is good to expose a large number of parameters for uncertainty quantification. It may not be good to allow code users to "tune" a large number of parameters by hand. Therefore it may be best to "hide" uncertainty quantification parameters from the code user and only expose them to DAKOTA.

Discussions have been initiated with John Turner and Russell Hooper of the VRI team to make progress toward this goal. Further we would separate our data into two groups, one for calibration and one for validation, and then perform the calibration study, followed by reassessing validation. Efforts to do this more complete study will be coordinated with Jeff Secker and Nam Dinh to find as much quality experimental data for CIPS as possible.

It is also important to note that for this calibration study we had a large amount of data and a small number of parameters, namely one inlet temperature. As the amount of parameters goes up and the relative amount of data goes down, the minimization problem solved as part of calibration becomes more difficult, and may even not have a unique solution. This difficulty in the calibration problem will increase the amount of computer resources required.

5.1. Dakota Deterministic Calibration Study

The first study calibrated temperature to void fraction data from experiment 6.1121. Optimization was initialized at the measured experimental inlet temperature and was then adjusted through the COBRA-TF input processor for about 15 different COBRA-TF runs. Table 11 summarizes the results of the initial calibration study. We achieved a one order of magnitude reduction in residual norm with a temperature adjustment of three degrees (one percent) by changing the inlet temperature from 301.5 to 297.92.

We note that this is not a physically realistic approach. In a "real" VUQ study we would calibrate closure model parameters like interfacial friction not inlet temperature which is a measured boundary condition.

Based on numerical local sensitivity analysis of the COBRA-TF code, together with a statistical test, Dakota computes a confidence interval for the estimated parameters. With 95% confidence, the "true" value of the parameter giving rise to the data lies in the calculated interval. In some sense, this provides a measure of the quality of the calibration. If the initial guess is within the confidence interval around the final value, then the amount of parameter adjustment was small, and perhaps there isn't a significant difference before and after the calibration. However, if the initial value is well outside of the confidence interval, then significant parametric changes were made to fit the data. In Table 11 we see that the final value for the inlet temperature is 297.92 with a confidence interval of 302.38 to 293.47. The initial value, based on the experimentally measure inlet temperature, of 301.5 is within the confidence interval. This indicates that a small calibration has been done.

In this "artificial" study the "uncertainty" in the inlet temperature is measured by how it impacts the void fraction at three levels in the assembly. In a "real" study void fraction where temperature is measured, the uncertainty in inlet temperature would be a fraction of a degree.

Table 11. Calibration of COBRA-TF temperature to 6.1121 void fraction dataset

	initial	final
temperature	301.5	297.92
confidence interval on temperature	-	[302.38, 293.47]
residual α_{lower}	0.0050	0.0030
residual α_{middle}	0.0715	0.0245
residual α_{upper}	0.0459	-0.0161
$f(t) = 0.5 \sum_i r_i^2$	3.6220e-03	4.3423e-04

5.2. Dakota Calibration, then Validation

The second calibration example demonstrates a more defensible use of calibration. The data is partitioned into two bins. One bin is used for calibration and the other for validation. This makes it possible to validate before and after calibration to measure the improvement in validation. As in the first calibration study, we hold pressure, mass flow rate, and power constant at the experimental configuration values. We then use experiment set 6.1122 to

calibrate a change in the inlet temperature that better matches the measured void fraction. We then present validation results from experiment set 6.1121 before and after this calibration update to temperature.

Table 12 summarizes the calibration of the inlet temperature from experiment set 6.1122. A small three degree change in temperature resulted in a two order of magnitude decrease in the least squares error. This is likely a quality calibration because the change is small relative to the numerical uncertainty. This is evidenced by the fact that the initial value 306.7 (the experimentally measured inlet temperature) is with the confidence range 312.95 to 294.43 of the calibrated value 303.69.

In a "real" study we should be varying closure law parameters not measured boundary conditions like inlet temperature. However, all of the closure law parameters are not available through the input processor. Since DAKOTA can only access COBRA-TF parameters through input processing. For demonstration purposes, we are showing how a parameter available through the input processor, like inlet temperature, can be calibrated with DAKOTA. Work needs to be done to expose the important parameters in COBRA-TF through its input processor.

Table 12. Calibration of COBRA-TF temperature to 6.1122 void fraction dataset

	initial	final
temperature	306.7	303.69
confidence interval on temperature	-	[312.95, 294.43]
residual α_{lower}	0.0510	0.0210
residual α_{middle}	0.1600	0.0421
residual α_{upper}	0.2300	-0.0554
$f(t) = 0.5 \sum_i r_i^2$	4.0551e-02	2.6413e-04

We then used this calibrated adjustment of the inlet temperature based on experiment 6.1122 and revalidate experiment 6.1121. These results are shown in Table 13. It is apparent that calibration from a different experiment reduces the least squares error by an order of magnitude. This is the proper way to do parameter calibration. Partition the data into two sets. Use one set to calibrate parameters and then use the other set to compare validation results. Validation on the data set used for calibration is a meaningless exercise. Note that there is an inherent assumption that there is a strong correlation between the calibration experiments and the validation experiments. In a "real" VUQ study the strength of the correlation between the calibration experiments and the validation experiments would have to be measured and documented.

Table 13. Prediction of void fraction, based on calibration

	Low Assembly Location	Mid Assembly Location	High Assembly Location	Error Norm
experimental data	0.0000	0.0055	0.1011	-
prediction w/o calibration	0.0040	0.0770	0.1470	3.6175e-03
prediction w/ calibration	0.0030	0.0370	0.0950	5.1923e-04

The importance of automating this verification, validation, calibration, and uncertainty quantification (preferably in Dakota) cannot be overstated. When new experimental data sets are acquired (through data mining or from running new experiments) one needs to quickly be able to partition the new data into calibration and validation bins and quickly rerun the calibration studies. When a "new code version" is available we need to re-verify and revalidate before we can determine whether the new data increased or decreased the uncertainty.

In addition to automation of VUQ it is important to have some place to store large amounts of experimental data and allow for easy access to that data. This is called a validation database. In addition to a database to store the data, one also needs to store the metadata (drawings, reports, journal papers generated from the experiment). This combination of data and metadata is called a knowledge base.

Note that automated VUQ test problems can also contribute to the SQA practice of regression testing. It is often difficult to get enough test problems for code coverage and by design VUQ test problems are designed to test the important software.

5.3. Simultaneous Calibration to Multiple Data Sets

Ideally, calibration approaches will simultaneously leverage data from multiple, potentially heterogeneous experiments, to best equip the computational model to predict various scenarios. For milestones occurring later in 2013, VUQ will demonstrate Bayesian calibration approaches that yield joint input parameter distributions consistent with the data. The deterministic calibration demonstrated here offers an example workflow that can later be generalized to the Bayesian calibration methods.

This was the third Dakota calibration study. It focused on the five data sets 6.xxx2, with the values of mdot, pressure, and power held at the experimentally reported values. A single ΔT is sought via calibration to maximize computational and experimental agreement for the 15 void fractions (low, middle, upper) x (five experiments). The results are shown in Table 14. Calibration of COBRA-TF inlet temperatures to five sets of three void fraction data sets show that fitting all datasets simultaneously is more challenging than a single data set.

Table 14. Calibration of COBRA-TF temperature to five 6.xxx2 void fraction datasets

		initial	final
Δ temperature		0.000	2.800
confidence interval on Δ temperature		-	[1.083, 4.586]
residuals (model – data)	61122L	0.0510	0.0230
	61122M	0.0901	0.0451
	61122U	-0.0064	-0.0514
	61452L	0.0190	0.0090
	61452M	0.0403	0.0033
	61452U	-0.0039	-0.0419
	62442L	0.0260	0.0110
	62442M	0.0449	0.0089
	62442U	-0.0140	-0.0510
	64562L	0.1312	0.0832
	64562M	0.0884	0.0474
	64562U	0.0864	0.0644
	66562L	0.0407	-0.0403
	66562M	0.0086	-0.0234
	66562U	-0.0174	-0.0389
$f(t) = 0.5 \sum_i r_i^2$		0.0251	0.0134

Table 14 highlights one challenge with heterogeneous data. The calibration results clearly reduced all of the errors in the Low and Mid regions of the assembly (the postfix L and M on the data set numbers). However, this occurred at the cost of increasing the error in the High region of the assembly (the postfix H on the data set number). Because the focus of this study is to assess uncertainty of COBRA-TF for CIPS, one may decide that subcooled boiling is most likely to occur in the Low and Mid assembly regions and therefore larger errors in the High assembly region do not matter. This highlights the importance of focusing VUQ on a challenge problem so one can make clear decisions on what is and is not important.

6. UNCERTAINTY QUANTIFICATION

Uncertainty quantification is separated into the following subsections: uncertainty verification, sensitivity validation, and uncertainty quantification (variability assessment) of uncertainty quantification. It is important to note that although COBRA-TF does not have a steady state option, it does have a restart option. For uncertainty quantification work it would make sense to use the steady state solution from the nominal (unperturbed solution) as the restart initial guess for the perturbed values used in the uncertainty quantification runs. This should improve the speed and robustness of the uncertainty quantification work.

6.1. Verification of UQ Approach

There is an uncertainty quantification method that has been growing in popularity within the nuclear power community. It has gained acceptance from the NRC and is employed by industry as part of their licensing work [5]. The method is based on what is now called the Wilks formula [6]. The Wilks formula provides the number of samples required for 95% coverage of the results domain with a 95% confidence. This 95/95 coverage/confidence is the level recognized by the NRC as an acceptable level of conservatism.

A number of assumptions go into the Wilks formula. Understanding these assumptions is crucial for CASL because industry standard software like the software System for Uncertainty and Sensitivity Analysis (SUSA) [7] from GRS in Germany [8] is based on the Wilks formula.

After some simplification, the one sided Wilks formula is given by

$$\gamma = 1 - \beta^n \tag{1.3}$$

Where β is the coverage, γ is the confidence, and n is the number of samples. Substituting $\beta = 0.95$ and $\gamma = 0.95$ and solving for n , one gets $n = 59$. So for verification of our uncertainty quantification with Dakota, we will perform a "code-to-code" comparison with Wilks formula, and a convergence study. Dakota input examples and detailed results for these studies are available in Appendix E. For this study the inlet temperature was varied by plus/minus one degree Celsius around the experimentally measured value.

Table 15. Verification of Dakota SA/UQ

	mean			standard deviation					
samples N	CI lower	value	CI upper	CI lower	value	CI upper	95th percentile	Wilks 95/95	dα/dT
59	0.1441	0.1467	0.1492	0.0083	0.0098	0.0120	0.1620	0.1640	-0.0169
590	0.1459	0.1467	0.1475	0.0092	0.0098	0.0104	0.1620	0.1640	-0.0169
5900	0.1464	0.1467	0.1469	0.0096	0.0098	0.0099	0.1620	0.1640	-0.0169

Table 15 shows the convergence study of the Dakota statistics. We show results for one, ten, and 100 times the Wilks formula number of 59. With this data, we do not see any change in the estimate of the mean, 0.1467, or the standard deviation, 0.0098, although our confidence is

increased as the number of samples increases (note the lower and upper bounds on the mean and standard deviation get tighter).

An estimate of the 95% response level is given by $\overline{\mu}+2\sigma$ where $\overline{\mu}$ is the mean and σ is the standard deviation. Thus we get three estimates of the 95% limit

1. 0.1663 = 0.1467 + (2 X 0.0098)
2. 0.1620 which is the 95th percentile from interpolation
3. 0.164 which is the 95/95 from the Wilks formula which is the largest value

The three approaches are identical to roughly two digits. Based on our convergence study and our "code-to-code" comparison, we are confident that the uncertainty calculations are being done correctly and our uncertainty quantification is verified. However, due to limitations in the precision of the calculated FOM, we could not verify the expected order of convergence of the sampling method.

Because we do not always understand all of the underlying assumptions of uncertainty quantification methods, it is prudent to do some verification of the method with each new application. It is possible that a new application will violate an assumption that was not violated in previous uncertainty quantification studies.

6.2. Validation of Sensitivity Analysis

Because PSBT data was gathered with the pressure, mass flux and power the same for two sets of data and only the inlet temperature changed, we can use the data to compute the sensitivity, s, of the void fraction with respect to the inlet temperature from the following simple equation,

$$s=\frac{\alpha\left(T+\Delta T\right)-\alpha\left(T\right)}{\Delta T} \tag{1.4}$$

where ΔT is the difference in inlet temperature between the two experiments. We can then compare the sensitivity from the experiment with the sensitivity of the code to validate our sensitivity calculation. Results are shown below in Table 16.

Table 16. Sensitivity Validation

Experiment	Low Assembly Location Experiment Sensitivity	Low Assembly Location CTF Sensitivity	Mid Assembly Location Experiment Sensitivity	Mid Assembly Location CTF Sensitivity	High Assembly Location Experiment Sensitivity	High Assembly location CTF Sensitivity
6112	0	0.0090	0.01238	0.015967	0.02660	0.01650
6145	0	0.0022	0.01237	0.012745	0.01311	0.01392
6244	0	0.0033	0.01202	0.01306	0.01302	0.01408
6456	0.0061	0.0104	0.01444	0.01171	0.00921	0.00755
6656	0.0174	0.0157	0.01297	0.00828	0.00632	0.00486

Table 16 shows the sensitivity data from the experiments and COBRA-TF for the five experiment pairs. The values in terms of absolute and relative errors in sensitivity are provided in Table 17.

Table 17. Sensitivity Errors

Experiment	Low Assembly Location Absolute	Low Assembly Location Percent	Mid Assembly Location Absolute	Mid Assembly Location Percent	High Assembly Location Absolute	High Assembly Location Percent
6112	0.0090	100	0.00359	22	-0.0101	-61
6145	0.0022	100	0.00038	3	0.00084	6
6244	0.0033	100	0.00104	8	0.00106	8
6456	0.0043	41	-0.00273	-23	-0.00166	-22
6656	-0.0017	-11	-0.00469	-57	-0.00146	-30

Based on the results in Table 17, due to the large errors in sensitivity, one would have to question the accuracy in the uncertainty quantification. Note that the error in the sensitivity is not that far out of line with the error in validation given in Table 10. This study would indicate that better data and better models may be needed to develop a high level of confidence in the uncertainty quantification.

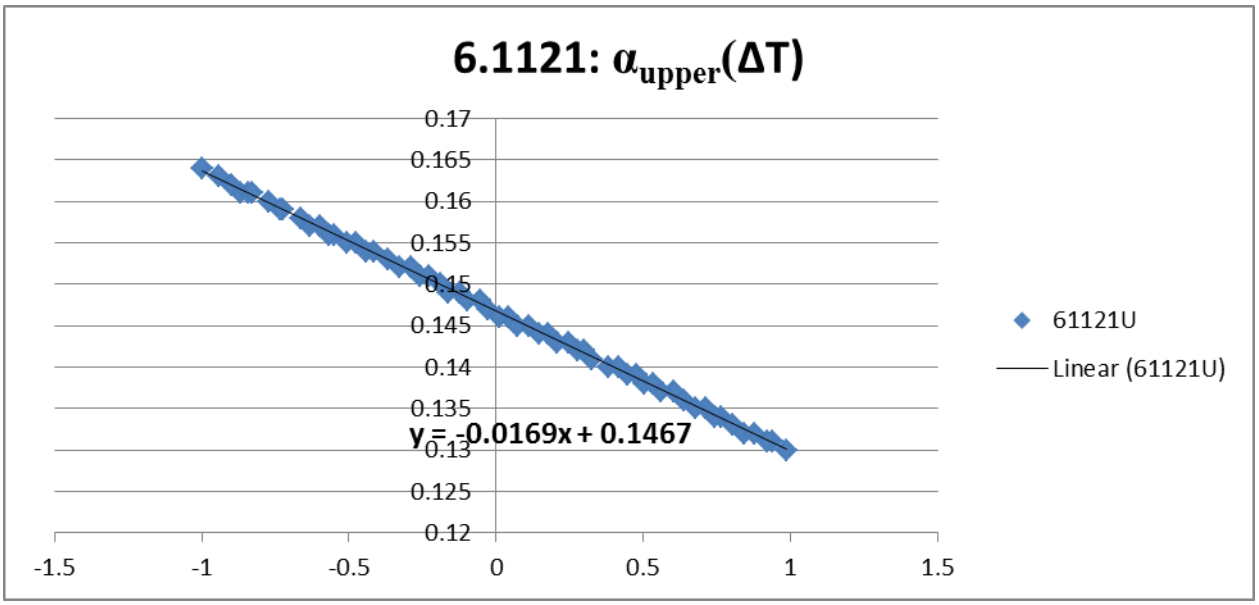


Figure 2. Local Sensitivities versus Global Sensitivities

Figure 2 shows a scatterplot of the Monte Carlo samples, for the 59 separate runs used for the Wilks study, together with an estimate of $s = \frac{\Delta\alpha}{\Delta T}$, obtained via linear regression in Excel. The plot clearly shows along with the table in appendix A that there is a linear dependence between the inlet temperature and the measured void fraction. It is due to this linearity that all of the uncertainty quantification methods are behaving so well. In a linear response, the answer is the same if the perturbation is small or large and the response is independent of where you start. Dakota reports partial correlation coefficients indicating -0.999299, which is commensurate with the scatter plot and linear regression. Full correlation results are included in Appendix A.

6.3. Replicate Analysis of UQ

Table 18 shows a study of the uncertainty in the Wilks formula uncertainty calculation. What we have done is to take 590 samples and put them into 10 bins of 59 samples. This effectively gives 10 replicates of Wilks formula. We see that there is very little difference in the 95/95 from Wilks formula so its predicted value is relatively insensitive to the exact sample. Therefore the uncertainty quantification of this uncertainty quantification method shows small uncertainty.

The following discussion came from the review by Laura Swiler of the VUQ team. Replicate analysis is used to understand the convergence of statistics across sample sets. For example, is the mean obtained from the first 59 samples "statistically the same" or "statistically different" from the mean of the second 59 samples? If we can say that the means are statistically the same, which indicates that 59 is enough samples to ensure a mean estimate which is stable. Helton and Davis (reference) explain the use of replicates in understanding the variation across sample sets.

In a "real" VUQ study, one would show an example of a t-test for comparison of the two means being equal. Laura did this for the two most extreme means: 0.1488 vs. 0.1453. This test, null hypothesis Ho: mean from replicate 2 = mean from replicate 9, was rejected in the t-test: it has a value of 2.01 but the t-test statistic at a 5% level of significance is 1.66. The other

combinations do pass, so this is not a huge discrepancy. However, in a "real" VUQ study we should use formal methods and not just say, "The uncertainty looks small."

Table 18 Uncertainty Quantification of Wilks formula

Replicate	Mean	Standard Deviation	$\bar{\mu}+2\sigma$	95th percentile	Wilks 95/95
1	0.1462	0.0095	0.1652	0.1620	0.1630
2	0.1488	0.0098	0.1684	0.1630	0.1640
3	0.1467	0.0099	0.1665	0.1630	0.1630
4	0.1469	0.0102	0.1673	0.1630	0.1630
5	0.1468	0.0094	0.1656	0.1620	0.1630
6	0.1463	0.0099	0.1661	0.1620	0.1630
7	0.1453	0.0091	0.1635	0.1600	0.1620
8	0.1454	0.0104	0.1662	0.1610	0.1630
9	0.1477	0.0090	0.1657	0.1620	0.1620
10	0.1468	0.0106	0.1680	0.1620	0.1630

7. SUMMARY AND FUTURE WORK

We have demonstrated many of the important steps in the VUQ process with Dakota and COBRA-TF, and documented findings in this report. The key idea is that to have confidence in the uncertainty quantification (which will be used in the decision making process) one needs to have good software quality, a solid verification of the solution methods (space, time and solver) and a validation process that demonstrates both the codes ability to match the data and the code's ability to match sensitivity.

7.1. Challenges

In the course of this study we have illuminated at least two challenges: requirements on VERA component input processors and effective strategies for multiphysics uncertainty quantification.

7.1.1. Input Parameters

Uncertainty quantification and data calibration can only be done with parameters that are exposed through the input processor. This creates the awkward situation of having to defend why we have assumed that the parameters not available through the input processor have a small uncertainty (small sensitivity) and are therefore unimportant.

Recognizing this problem, we have initiated a two pronged approach on how to get better uncertainty quantification into VERA. For legacy codes we have laid out a strategy to expose important parameters in an evolutionary way (see future direction section below). For new software, the VUQ and VRI FA leads need to work with the other FA leads to make sure that parameters that may have an impact on uncertainty are exposed through the input processor. This either needs to be done before or immediately after, the software is incorporated into VERA.

We are currently working on a September 2013 Level 3 Milestone to interface Dakota to the VERA input processor. Therefore, as long as the VERA input processor has access all of the important parameters, Dakota can provide high quality uncertainty quantification and data calibration in a consistent way to all of the VERA components.

7.1.2. Multiphysics Uncertainty Quantification

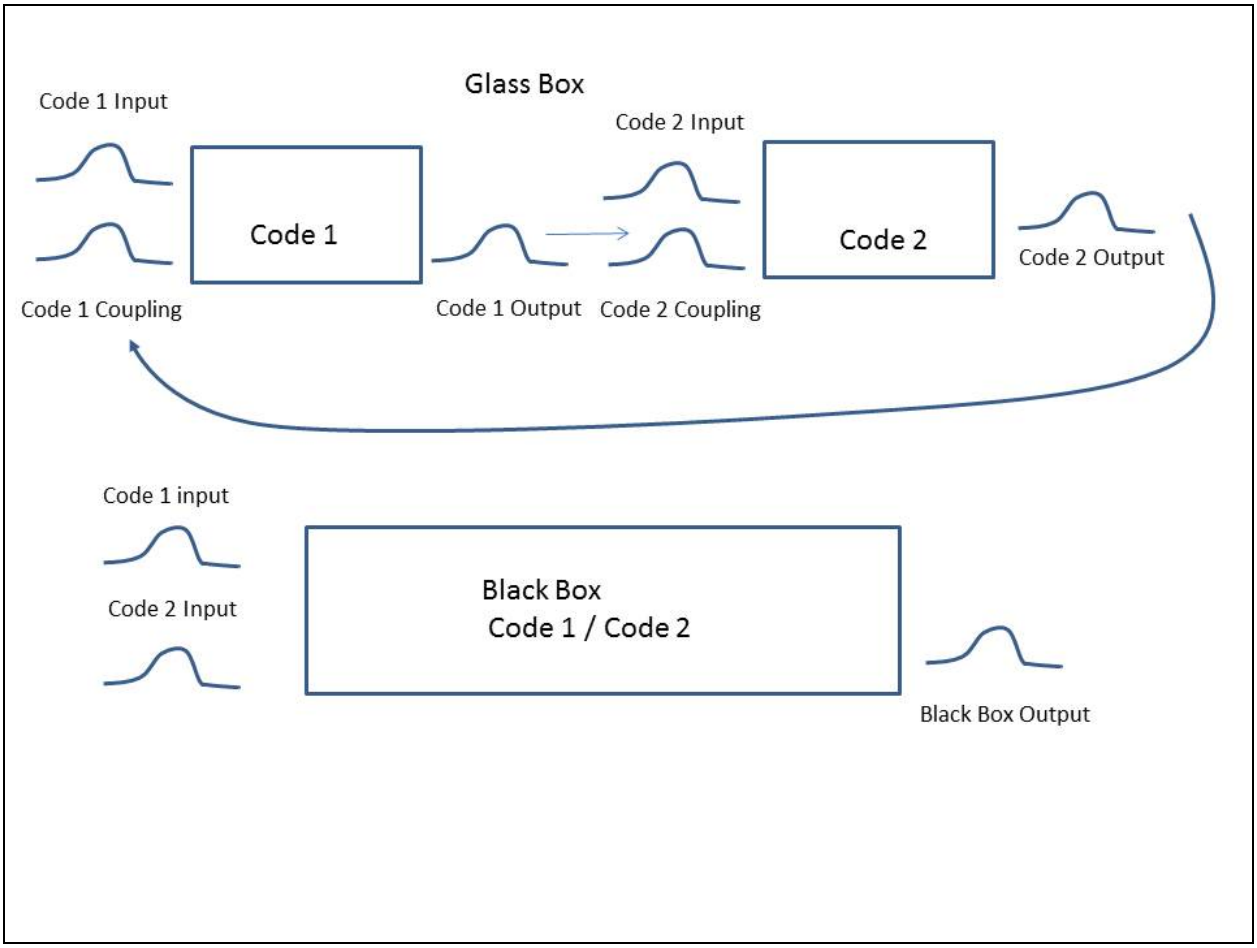


Figure 3. Multiphysics Uncertainty Quantification

There seem to be at least two distinct ways to do multiphysics uncertainty quantification. For discussion, we will call these a "glass box approach" and a "black box approach." It is important to note here that we are assuming a specific type of multiphysics coupling that can be represented by the coupling between DENOVO and COBRA-TF. The state variables in DENOVO are neutron flux, the state variables in COBRA-TF are pressure, velocity, fluid enthalpy, and fuel temperature (fluid density and temperature come from the EOS called with pressure and enthalpy). Neutron flux is not passed from DENOVO to COBRA-TF, the coupling is through energy addition. The neutron flux is multiplied by a parameter for fission per flux and a parameter for energy per fission and energy is passed. Fluid (moderator) density and fuel temperature are not passed to DENOVO from COBRA-TF. The coupling is through a change in cross-section. The density is multiplied by a parameter for change in cross-section per density and the fuel temperature is multiplied by a parameter for change in cross-section per fuel temperature. Note that for steady state calculations the density and temperature are passed to a third code (Scale) which then passes the new cross sections to DENOVO.

In this type of coupling there is uncertainty in the value being computed in the first code (neutron flux or fluid density and fuel temperature) but there is also uncertainty in the parameters that are

used to convert to the actual coupling variables passed between codes (energy source or cross-sections).

Rod Schmidt from the Virtual Reactor Integration (VRI) team pointed out that placement of the multiphysics boxes between code one and code two is somewhat arbitrary. Clearly, COBRA-TF, which has multiphase flow coupled to nonlinear heat conduction, is in itself a black box multiphysics code coupling liquid physics, to vapor physics, to droplet physics to wall heat conduction physics. Distinguishing between multiphysics inside of a code and multiphysics outside of a code is unclear at this point. Rod Schmidt also noted that currently the code couplings inside of VERA are following the black box model. It is clear from an uncertainty quantification point of view; it does seem to make a difference. Perhaps a study that looks at uncertainty quantification of COBRA-TF coupled to DENOVO (or MAMBA or MPACT) with both a glass box and black box approach is warranted. If the glass box and black box approaches give roughly the same uncertainty then we can use that as evidence for why we chose to ignore the additional uncertainties from the glass box approach. However, if significant differences are discovered in the uncertainty analysis of the glass box and black box approaches, then further discussion shall be warranted.

Discussions with Noel Belcourt of the VRI team revealed that injecting DAKOTA into the coupled code solution inside of VERA may be possible. After DTK maps the data correctly from code one to code two, the data could then be passed to DAKOTA for perturbation before it is passed to code two. This would require a second DAKOTA interface with VERA, one with the VERA common input processor and one with DTK. These two interfaces would allow CASL to address uncertainty of both the input parameters and the data passed between codes.

7.1.2.1. Glass Box

The glass box UQ approach is represented in the top half of Figure 3. In the glass box approach, we emphasize that there is uncertainty in the output from code one (code two) and that this uncertainty impacts the input to code two (code one). As shown in Figure 3, code one has its own input but it also has inputs from code two. Since there is uncertainty both in the parameters in the input file and in the parameters coming from code two, there should be distributions given to these parameters to represent their uncertainty. Based on distributions from code one input and code one coupling inputs (that come from code two) we get a distribution of outputs from code one. This distribution of outputs from code one then becomes a distribution of code two coupling inputs. This distribution of code two coupling inputs is then combined with distributions of code two input parameters to provide input to code two. These two input distributions to code two produce a distribution of code two outputs. The code two output distributions then becomes code one coupling input.

This provides a nonlinear system that takes as input the following four distributions:

1. Code one input distribution (fixed),
2. Code one output distribution (computed based on code one input distribution and code two output distribution),
3. Code two input distribution (fixed), and
4. Code two output distribution (computed based on code two input distribution and code one output distribution).

This nonlinear system needs to be iterated until the code one output distribution and the code two output distributions converge. We then know the uncertainty of the figure of merit in terms of the uncertainty in the input parameters and the uncertainty in the coupling between code one and code two (represented by the code one output distribution and the code two output distribution).

7.1.2.2 Frosted Glass Box

The "Frosted Glass" box approach was suggested by Rod Schmidt of the VRI team. In this approach one only does a single iteration of the "glass" box approach. That is the problem is run to completion without perturbing the data transferred between codes. One then adds an uncertainty perturbation to the variable passed between codes once and measures the impact on the solution. This approach is likely to be very accurate for steady state problems and significantly cuts down on the amount of compute resources required for the glass box approach. This is a good compromise position.

7.1.2.3 Black Box

The black box approach is the traditional way of doing multiphysics uncertainty quantification. This is represented in the bottom half of Figure 3. Here we treat the coupled code as a black box that we cannot see into. This black box takes input distributions that are the same as the input distributions to code one and code two. The black box then produces an output distribution. From this black box output distribution, we can compute the uncertainty of the figure of merit in terms of the uncertainty in the code one input and the code two input.

It appears that the black box approach has assumed that there is no uncertainty in the code one output (that is used as input to code two) and no uncertainty in the code two output (that is used as input into code one.)

7.1.2.4 Impact

It seems that CASL needs to make a choice on how it does multiphysics uncertainty quantification. It can choose to employ a glass box approach which will require:

1. Additional work on the code coupling to allow for output distributions to become input distributions
2. Significant increase in run time for multiphysics uncertainty quantification to allow for iteration to convergence of the code coupling parameters distributions.

Alternatively, we can choose to employ a black box approach, which will require justification of the assumption that there is no uncertainty in the code coupling parameters. It is important to note that as VERA applications include more coupled codes the complexity of the coupling uncertainty distributions goes up factorially with the number of coupled codes (assuming that all codes communicate with all other codes). By definition, since it is a black box approach, this method is the same whether two codes or ten codes are coupled together in the black box.

At a minimum, additional investigation into previous studies in this area is warranted. For example, the work of Phipps and Constantine on network-based and multiphysics UQ may be relevant. [9,10]

7.2. Interactions With Other FA's

This work was strongly supported by the Penn State COBRA-TF team of Kostadin Ivanov, Maria Avramova, Robert Salko, and Taylor Blyth. The future work of how to better integrate COBRA-TF with VERA and Dakota was discussed with John Turner and Russell Hooper. Weekly status meetings with Rod Schmidt and Randy Summers have been conducted to keep them apprised of the work.

This is the first step in the process of building VUQ plans (for the challenge problems and/or the codes). With the first step completed, VUQ is now ready for tighter interactions with AMA and the Challenge Problem Integrators (CPIs) to build VUQ plans for the challenge problems (starting with CIPS) and the VERA components.

7.3. Future Work

Based on discussions with Robert Salko of the VRI team, there may be a way to improve the sensitivity of the pseudo-steady state convergence criteria. By slightly modifying these convergence criteria we could refine the detection of steady state. The refinement is based on the simple idea of adding absolute values to quantities that are summed before they are compared to the pseudo-steady state convergence criteria. Because this summation is currently done with signed quantities, it is possible for large negative errors to cancel with large positive errors.

Although Dakota is an uncertainty quantification tool, there are enough similarities in the processes of verification, validation, data calibration, and uncertainty quantification, that it makes sense to do all of these processes in a single interface between Dakota and the VERA common input processor. We will continue to work toward this goal.

One of the key challenges to accurate and useful uncertainty quantification is to get access to the parameters in the code. There are literally hundreds of parameters in COBRA-TF where the definition of a parameter from an uncertainty quantification perspective is:

Parameter – any constant value in a code that impacts the solution.

The reason for the large number of parameters in COBRA-TF is there are many flow regimes and each flow regime is composed of many correlations and each correlation is built up from combinations of other correlations and any given correlation contains three to five constants. If we were building a subchannel code from scratch, we would provide access to all of these parameters through the input processor. We would then have them available for calibration and uncertainty quantification. However, since COBRA-TF is a legacy code we need to develop an evolutionary strategy. This work is based on a discussion with Robert Martin who is currently employed at MPOWER and is based on work that he did at AREVA [5]

Since COBRA-TF is a three field code, liquid, vapor, droplets there are the following important interactions where parameters are important (closure relations):

1. Mass exchange between liquid and vapor,
2. Momentum exchange between liquid and vapor,
3. Energy exchange between liquid and vapor,
4. Mass exchange between liquid and droplets,
5. Momentum exchange between liquid and droplets,
6. Energy exchange between liquid and droplets,
7. Momentum exchange between liquid and the wall,
8. Energy exchange between liquid and the wall,
9. Mass exchange between vapor and droplets,
10. Momentum exchange between vapor and droplets,
11. Energy exchange between vapor and droplets,
12. Momentum exchange between vapor and the wall,
13. Energy exchange between the vapor and the wall,
14. Momentum exchange between the droplets and the wall,
15. Energy exchange between the droplets and the wall.

Since we would like to conserve mass, momentum, and energy we assume that these interactions are symmetric. For example, we assume that the energy taken from the wall and deposited in the liquid is equal to the energy deposited in the liquid from the wall. This may sound obvious, however, in legacy operator split codes like COBRA-TF, this is often not the case and should be verified.

The proposal is to first add fifteen new parameters available through input that multiply these exchange terms. After discussing this with John Turner and Russ Hooper, it was determined that this work should be delayed until a good set of regression tests are available in COBRA-TF.

Given these 15 parameters, the first thing to do is set them all to zero. With all of the exchange terms set to zero, it will be much easier to verify COBRA-TF. For a given challenge problem, perform a sensitivity study on the challenge problem's FOM. Next list these 15 parameters in order of importance based on their impact on the FOM. Given time and resources, one can then start to add parameters that decompose the tree of each of the 15 trunk parameters starting with the one with the largest uncertainty. Next, slowly add new parameters that are exposed to the input processor that help to further refine the sources of the uncertainty in CTF.

Through this evolutionary process, uncertainty quantification for each of the challenge problems will start to define a set of important parameters. These important parameters need to be accessible to the VERA common input processor so uncertainty quantification (and calibration) can be done.

8. REFERENCES

1. R.K. Salko and Maria N. Avramova, "CTF Theory Manual," The Pennsylvania State University, Feb 1, 2013.

2. B.M. Adams et al, "Dakota User Manual Version 5.1," Sandia National Laboratories, SAND2010-2183, Jan. 21, 2011.

3. A. Rubin et al, "OECD/NRC Benchmark Based on NUPEC PWR Subchannel and Bundle Tests (PSBT): Volume I: Experimental Database and final Problem Specification," United States Nuclear Regulatory Commission, Nov. 2010.

4. M. Avramova, "CTF – A Thermal-Hydraulic Subchannel Code for LWRs Transient Analysis: User's Manual," The Pennsylvania State University, Feb. 2009.

5. R.P. Martin and L.D. O'Dell, "AREVA's realistic large break LOCA analysis methodology," *Nuclear Engineering and Design*, **Vol. 235**, pp. 1713-1725, 2005.

6. S.S Wilks, "Determination of Sample Sizes for Setting Tolerance Limits," *The Annals of Mathematical Statistics*, **Vol. 12, No. 1**, pp. 91-96, 1941.

7. M. Kloos and E. Hofer, "SUSA-PC, a personal computer version of the program system for uncertainty and sensitivity analysis of results from computer models, version 3.2, user's guide and tutorial, Gesellschaft fur Anlagen- und Reaktorsicherheit, Garching, Germany, August 1999.

8. H. Glaeser, "GRS Method for Uncertainty and Sensitivity Evaluation of Code Results and Applications," *Science and Technology of Nuclear Applications*, **Vol. 2008**, 7 pages, 2008.

9. M. Arnst, R. Ghanem, E. Phipps, J. Red-Horse, "Dimension reduction in stochastic modeling of coupled problems," *International Journal for Numerical Methods in Engineering*, **Vol. 92**, pp. 940-968, 2012.

10. M. Arnst, R. Ghanem, E. Phipps, J. Red-Horse, "Measure transformation and efficient quadrature in reduced-dimensional stochastic modeling of coupled problems," *International Journal for Numerical Methods in Engineering*, **Vol. 92**, pp. 1044-1080. 2012.

11. J. Helton and F. Davis**, "**Latin Hypercube Sampling and the Propagation of Uncertainty in Analyses of Complex Systems. *Reliability Engineering and System Safety*, **Vol. 81(1)**, pp. 23-69, 2003.

APPENDIX A. DAKOTA VERIFICATION STUDY OF CFL VALUE

A Dakota list parameter study can efficiently evaluate the effect of the time step-controlling CFL condition (COBRA-TF parameter COURANT) for all ten experimental configurations. Here values of 0.2, 0.4, 0.8, and 1.6 were chosen as multiples (0.8 is the default), 0.5 as it is typical, and 3.0 as some COBRA-TF documentation suggest. The relevant Dakota method and variables specifications are shown here:

```
method
        list_parameter_study
        list_of_points = 0.2 0.4 0.5 0.8 1.0 1.6 3.0

variables
        continuous_design = 1
        descriptors = "courant"
```

Table 19 through Table 28 report the void fractions (lower L, middle M, upper U) predicted by COBRA-TF for various CFL conditions, for all ten experimental configurations. Deviant cases are highlighted.

Table 19. CFL verification for experiment 6.1121

courant	61121L	61121M	61121U
0.2	0.0040	0.0770	0.1470
0.4	0.0040	0.0770	0.1470
0.5	0.0040	0.0770	0.1470
0.8	0.0040	0.0770	0.1470
1.0	0.0040	0.0770	0.1470
1.6	0.0040	0.0770	0.1470
3.0	0.0040	0.0770	0.1470

Table 20. CFL verification for experiment 6.1122

courant	61122L	61122M	61122U
0.2	0.0510	0.1600	0.2330
0.4	0.0510	0.1600	0.2330
0.5	0.0510	0.1600	0.2330
0.8	0.0510	0.1600	0.2330
1.0	0.0510	0.1600	0.2330
1.6	0.0510	0.1600	0.2330
3.0	0.0510	0.1600	0.2330

Table 21. CFL verification for experiment 6.1451

courant	61451L	61451M	61451U
0.2	0.0080	0.0990	0.2270
0.4	0.0080	0.0990	0.2270
0.5	0.0080	0.0990	0.2270
0.8	0.0080	0.0990	0.2270
1.0	0.0080	0.0990	0.2270
1.6	0.0080	0.0990	0.2270
3.0	0.0080	0.0990	0.2280

Table 22. CFL verification for experiment 6.1452

courant	61452L	61452M	61452U
0.2	0.0190	0.1640	0.2980
0.4	0.0190	0.1640	0.2980
0.5	0.0190	0.1640	0.2980
0.8	0.0190	0.1640	0.2980
1.0	0.0190	0.1640	0.2980
1.6	0.0190	0.1640	0.2980
3.0	0.0190	0.1640	0.2990

Table 23. CFL verification for experiment 6.2441

courant	62441L	62441M	62441U
0.2	0.0100	0.1300	0.2860
0.4	0.0100	0.1300	0.2860
0.5	0.0100	0.1300	0.2860
0.8	0.0100	0.1300	0.2860
1.0	0.0100	0.1300	0.2860
1.6	0.0100	0.1300	0.2860
3.0	0.0100	0.1300	0.2860

Table 24. CFL verification for experiment 6.2442

courant	62442L	62442M	62442U
0.2	0.0260	0.1940	0.3550
0.4	0.0260	0.1940	0.3550
0.5	0.0260	0.1940	0.3550
0.8	0.0260	0.1940	0.3550
1.0	0.0260	0.1940	0.3550
1.6	0.0260	0.1940	0.3550
3.0	0.0260	0.1940	0.3550

Table 25. CFL verification for experiment 6.4561

courant	64561L	64561M	64561U
0.2	0.0460	0.2870	0.4960
0.4	0.0460	0.2870	0.4960
0.5	0.0460	0.2870	0.4960
0.8	0.0460	0.2870	0.4960
1.0	0.0460	0.2870	0.4960
1.6	0.0460	0.2870	0.4960
3.0	0.0460	0.2870	0.4960

Table 26. CFL verification for experiment 6.4562

courant	64562L	64562M	64562U
0.2	0.2580	0.5260	0.6500
0.4	0.2580	0.5260	0.6500
0.5	0.2580	0.5260	0.6500
0.8	0.2580	0.5260	0.6500
1.0	0.2580	0.5260	0.6500
1.6	0.2580	0.5260	0.6500
3.0	0.2580	0.5260	0.6418

Table 27. CFL verification for experiment 6.6561

courant	66561L	66561M	66561U
0.2	0.1830	0.5520	0.6920
0.4	0.1830	0.5520	0.6920
0.5	0.1830	0.5520	0.6920
0.8	0.1830	0.5520	0.6920
1.0	0.1830	0.5520	0.6930
1.6	0.1810	0.5460	0.6715
3.0	0.1830	0.5520	0.6768

Table 28. CFL verification for experiment 6.6562

courant	66562L	66562M	66562U
0.2	0.4110	0.6720	0.7610
0.4	0.4110	0.6720	0.7610
0.5	0.4110	0.6720	0.7610
0.8	0.4110	0.6720	0.7625
1.0	0.4110	0.6720	0.7570
1.6	0.4090	0.6690	0.7760
3.0	0.4110	0.6720	0.7670

APPENDIX B. DAKOTA VERIFICATION STUDY OF LINEAR SOLVER CHOICE

The linear solver in COBRA-TF is chosen by the input parameter ISOL, with values from 0 to 4. As noted previously, solver 4 did not converge. Here, a Dakota study considers a discrete variable from 0 to 3 using a grid parameter study with 3 partitions (4 values). The Dakota variable linsolv maps directly to the COBRA-TF parameter ISOL. The relevant Dakota method and variables specifications are shown here:

```
method
        multidim_parameter_study
        partitions = 3

variables
        discrete_design_range = 1
        lower_bounds = 0
        upper_bounds = 3
        descriptors = "linsolv"
```

Table 29 through Table 38 report the void fractions (lower L, middle M, upper U) predicted by COBRA-TF for various linear solver choices, for all ten experimental configurations. Suspect cases where convergence was not reached are highlighted.

Table 29. Linear solver verification for experiment 6.1121

ISOL	Solver	61121L	61121M	61121U
0	Direct	0.0040	0.0770	0.1470
1	BSGS-ILUT	0.0040	0.0770	0.1470
2	GMRES	0.0040	0.0770	0.1470
3	BSGS	0.0040	0.0770	0.1470

Table 30. Linear solver verification for experiment 6.1122

ISOL	Solver	61122L	61122M	61122U
0	Direct	0.0510	0.1600	0.2330
1	BSGS-ILUT	0.0510	0.1600	0.2330
2	GMRES	0.0510	0.1600	0.2330
3	BSGS	0.0510	0.1600	0.2330

Table 31. Linear solver verification for experiment 6.1451

ISOL	Solver	61451L	61451M	61451U
0	Direct	0.0080	0.0990	0.2270
1	BSGS-ILUT	0.0080	0.0990	0.2270
2	GMRES	0.0080	0.0990	0.2270
3	BSGS	0.0080	0.0990	0.2270

Table 32. Linear solver verification for experiment 6.1452

ISOL	Solver	61452L	61452M	61452U
0	Direct	0.0190	0.1640	0.2980
1	BSGS-ILUT	0.0190	0.1640	0.2980
2	GMRES	0.0190	0.1640	0.2980
3	BSGS	0.0190	0.1640	0.2980

Table 33. Linear solver verification for experiment 6.2441

ISOL	Solver	62441L	62441M	62441U
0	Direct	0.0100	0.1300	0.2860
1	BSGS-ILUT	0.0100	0.1300	0.2860
2	GMRES	0.0100	0.1300	0.2860
3	BSGS	0.0100	0.1300	0.2860

Table 34. Linear solver verification for experiment 6.2442

ISOL	Solver	62442L	62442M	62442U
0	Direct	0.0260	0.1940	0.3550
1	BSGS-ILUT	0.0260	0.1940	0.3550
2	GMRES	0.0260	0.1940	0.3550
3	BSGS	0.0260	0.1940	0.3550

Table 35. Linear solver verification for experiment 6.4561

ISOL	Solver	64561L	64561M	64561U
0	Direct	0.0460	0.2870	0.4960
1	BSGS-ILUT	0.0460	0.2870	0.4960
2	GMRES	0.0460	0.2870	0.4960
3	BSGS	0.0460	0.2870	0.4960

Table 36. Linear solver verification for experiment 6.4562

ISOL	Solver	64562L	64562M	64562U
0	Direct	0.2580	0.5260	0.6500
1	BSGS-ILUT	0.2580	0.5260	0.6500
2	GMRES	0.2580	0.5260	0.6500
3	BSGS	0.2580	0.5260	0.6500

Table 37. Linear solver verification for experiment 6.6561

ISOL	Solver	66561L	66561M	66561U
0	Direct	0.1830	0.5520	0.6920
1	BSGS-ILUT	0.1830	0.5520	0.6920
2	GMRES	0.0060	0.0070	0.0040
3	BSGS	0.1830	0.5520	0.6920

For the GMRES solver, COBRA-TF reported

```
 fatal error
*trans* outer iteration did not converge

 timet= 3.39566E-01   delt=2.191010E-05   nstep=1380 oitno=1
```

The third row of table 37 is the only code crash in this study.

Table 38. Linear solver verification for experiment 6.6562

ISOL	Solver	66562L	66562M	66562U
0	Direct GEM	0.4110	0.6720	0.7625
1	BSGS-ILUT	0.4110	0.6720	0.7625
2	GMRES	0.4120	0.6720	0.7625
3	BSGS	0.4110	0.6720	0.7625

APPENDIX C. DAKOTA VERIFICATION STUDY OF STEADY STATE CONVERGENCE CRITERIA

The steady state convergence criteria in COBRA-TF are controlled by parameters specified in the source code, not through input. A Dakota study was conducted over discrete variables from 0 to 4 using a grid parameter study with 4 partitions (5 values). The analysis driver run_cobra.sh ran the appropriate COBRA-TF binary corresponding to the desired convergence criteria. The relevant Dakota method and variables specifications are shown here:

```
method
        multidim_parameter_study
        partitions = 4

variables
        discrete_design_range = 1
        lower_bounds = 0
        upper_bounds = 4
        descriptors = "ssconv"
```

Table 39 through Table 48 report the void fractions (lower L, middle M, upper U) predicted by COBRA-TF for various steady state convergence criteria, for all ten experimental configurations. Values deviating from the result with tightest tolerance are highlighted.

Table 39. Steady state convergence criteria verification for experiment 6.1121

tolerance	61121L	61121M	61121U
0.125	0.0040	0.0770	0.1470
0.250	0.0040	0.0770	0.1470
0.500	0.0040	0.0770	0.1470
1.000	0.0040	0.0770	0.1460
2.000	0.0040	0.0760	0.1460

Table 40. Steady state convergence criteria verification for experiment 6.1122

tolerance	61122L	61122M	61122U
0.125	0.0510	0.1600	0.2330
0.250	0.0510	0.1600	0.2330
0.500	0.0510	0.1600	0.2330
1.000	0.0510	0.1600	0.2320
2.000	0.0510	0.1600	0.2320

Table 41. Steady state convergence criteria verification for experiment 6.1451

tolerance	61451L	61451M	61451U
0.125	0.0080	0.0990	0.2280
0.250	0.0080	0.0990	0.2280
0.500	0.0080	0.0990	0.2270
1.000	0.0080	0.0990	0.2270
2.000	0.0080	0.0990	0.2260

Table 42. Steady state convergence criteria verification for experiment 6.1452

tolerance	61452L	61452M	61452U
0.125	0.0190	0.1640	0.2990
0.250	0.0190	0.1640	0.2990
0.500	0.0190	0.1640	0.2980
1.000	0.0190	0.1640	0.2980
2.000	0.0190	0.1640	0.2970

Table 43. Steady state convergence criteria verification for experiment 6.2441

tolerance	62441L	62441M	62441U
0.125	0.0100	0.1300	0.2860
0.250	0.0100	0.1300	0.2860
0.500	0.0100	0.1300	0.2860
1.000	0.0100	0.1300	0.2850
2.000	0.0100	0.1300	0.2840

Table 44. Steady state convergence criteria verification for experiment 6.2442

tolerance	62442L	62442M	62442U
0.125	0.0260	0.1950	0.3560
0.250	0.0260	0.1950	0.3560
0.500	0.0260	0.1940	0.3550
1.000	0.0260	0.1940	0.3550
2.000	0.0260	0.1940	0.3530

Table 45. Steady state convergence criteria verification for experiment 6.4561

tolerance	64561L	64561M	64561U
0.125	0.0460	0.2870	0.4970
0.250	0.0460	0.2870	0.4970
0.500	0.0460	0.2870	0.4960
1.000	0.0460	0.2870	0.4960
2.000	0.0460	0.2870	0.4940

Table 46. Steady state convergence criteria verification for experiment 6.4562

tolerance	64562L	64562M	64562U
0.125	0.2580	0.5260	0.6500
0.250	0.2580	0.5260	0.6500
0.500	0.2580	0.5260	0.6500
1.000	0.2580	0.5260	0.6490
2.000	0.2580	0.5250	0.6480

Table 47. Steady state convergence criteria verification for experiment 6.6561

tolerance	66561L	66561M	66561U
0.125	0.1830	0.5520	0.6920
0.250	0.1830	0.5520	0.6920
0.500	0.1830	0.5520	0.6920
1.000	0.1830	0.5520	0.6920
2.000	0.1830	0.5510	0.6910

Table 48. Steady state convergence criteria verification for experiment 6.6562

tolerance	66562L	66562M	66562U
0.125	0.4110	0.6720	0.7625
0.250	0.4110	0.6720	0.7625
0.500	0.4110	0.6720	0.7625
1.000	0.4110	0.6720	0.7615
2.000	0.4100	0.6710	0.7605

Results from Tables 41, 42, 44, and 45 indicate that the default steady state convergence criteria value of 0.500 is not tight enough and the whole study should be reproduced with Heat Balance and Mass Balance set to 0.25 and Heat % Deviate and Mass % Deviate should be set to 0.005 (See Table 4).

APPENDIX D. DAKOTA CALIBRATION EXAMPLE

Representative Dakota input file for calibration:

```
# Calibrate a deltaT variable, nominally 0, to five experiments
# 6.xxx2 simultaneously, to find the best temperature offset in the code

# Use application templates that have default values for mdot, pressure, power
# And templatize using multipliers on deltaT for low pressures

# Template needs to look like { OrigTemp + DeltaTemp * factor }

strategy
        single_method
          tabular_graphics_data

method
        nl2sol
          convergence_tolerance = 1.0e-3
          output verbose

variables
        continuous_design = 1
          initial_point = 0.0
          upper_bounds = 30.0
          lower_bounds = -30.0
          descriptors = 'DeltaTemp'

interface
        fork
          asynchronous evaluation_concurrency = 3
          analysis_driver = 'run_cobra.sh'
          template_directory 'templatedir'
          work_directory named 'workdir'
          directory_tag directory_save
          parameters_file 'params.in'
          results_file 'results.out'
          file_save
          aprepro

responses
        # Exit void fraction for 5 experiment configurations
        calibration_terms = 15
        calibration_data_file = '6.xxx2.dat'
          freeform
        descriptors =   "61122L"        "61122M"        "61122U"
                        "61452L"        "61452M"        "61452U"
                        "62442L"        "62442M"        "62442U"
                        "64562L"        "64562M"        "64562U"
                        "66562L"        "66562M"        "66562U"

        # Since we have deltaT near zero, try absolute FD step of 0.5
        numerical_gradients
          method_source dakota
          interval_type central
          # Consider step of 0.1 in subsequent studies
          fd_gradient_step_size = 0.5
          absolute
          # Want to get a perturbation of at least 1.0 K; try relative init
#         fd_gradient_step_size = 0.01
        no_hessians
```

APPENDIX E. DAKOTA SENSITIVITY AND UNCERTAINTY STUDIES: DETAILED RESULTS

This appendix summarizes Dakota-based global sensitivity and uncertainty studies of five select experimental configurations with various Monte Carlo sample sizes. For these studies, Dakota was configured to generate 59, 590, and 5900 random samples over a uniform distribution of temperature perturbations on the interval [-1°C, +1°C]. This perturbation was scaled based on the nominal pressure of the various experimental configurations, according to Table 49. The size of the perturbation is scaled by the latent heat so one roughly sees the same change in void fraction. The latent heats for this scaling are from
G. J. Van Wylen and R.E. Sonntag, "Fundamentals of Classical Thermodynamics 3rd edition," John Wiley & Sons, 1985.

Table 49. Scaled perturbation ΔT applied to each experiment

experiment	Pressure MPa	Latent heat kJ/kg	perturbation scaling
6.112x	16.8	893.4	1.00
6.145x	16.9	893.4	1.00
6.244x	15	1027.9	1.15
6.456x	10	1326.0	1.48
6.656x	5	1662.5	1.86

Uncertainty: Table 50 through Table 52 summarizes mean and standard deviation statistics, together with confidence intervals on them based on sample sizes, for five experimental configurations.

Table 50. Statistics for five experimental configurations, 59 samples.

	mean			standard deviation		
response	CI Lower	value	CI Upper	CI lower	value	CI Upper
61121L	0.005	0.006	0.007	0.002	0.003	0.003
61121M	0.075	0.077	0.079	0.007	0.008	0.010
61121U	0.144	0.147	0.149	0.008	0.010	0.012
61451L	0.008	0.008	0.008	0.000	0.000	0.000
61451M	0.098	0.099	0.101	0.005	0.006	0.007
61451U	0.225	0.227	0.230	0.007	0.008	0.010
62441L	0.010	0.010	0.010	0.000	0.000	0.000
62441M	0.129	0.130	0.132	0.006	0.007	0.008
62441U	0.284	0.286	0.288	0.007	0.008	0.010
64561L	0.044	0.045	0.047	0.004	0.005	0.006
64561M	0.284	0.287	0.290	0.009	0.010	0.013
64561U	0.494	0.496	0.498	0.007	0.009	0.011
66561L	0.179	0.183	0.187	0.013	0.015	0.018
66561M	0.548	0.552	0.555	0.010	0.012	0.014
66561U	0.691	0.692	0.693	0.005	0.006	0.007

Table 51. Statistics for five experimental configurations, 590 samples.

	mean			standard deviation		
response	CI Lower	value	CI Upper	CI lower	value	CI Upper
61121L	0.006	0.006	0.006	0.002	0.002	0.003
61121M	0.076	0.077	0.078	0.008	0.008	0.009
61121U	0.146	0.147	0.147	0.009	0.010	0.010
61451L	0.008	0.008	0.008	0.000	0.000	0.000
61451M	0.099	0.099	0.100	0.006	0.006	0.006
61451U	0.227	0.227	0.228	0.008	0.008	0.009
62441L	0.010	0.010	0.010	0.000	0.000	0.000
62441M	0.130	0.130	0.131	0.006	0.007	0.007
62441U	0.285	0.286	0.287	0.008	0.008	0.009
64561L	0.045	0.045	0.046	0.005	0.005	0.005
64561M	0.286	0.287	0.288	0.010	0.010	0.011
64561U	0.496	0.496	0.497	0.008	0.009	0.009
66561L	0.182	0.183	0.184	0.014	0.015	0.016
66561M	0.551	0.552	0.553	0.011	0.012	0.012
66561U	0.692	0.692	0.692	0.005	0.006	0.006

Table 52. Statistics for five experimental configurations, 5900 samples.

	mean			standard deviation		
response	CI Lower	value	CI Upper	CI lower	value	CI Upper
61121L	0.006	0.006	0.006	0.002	0.002	0.003
61121M	0.077	0.077	0.077	0.008	0.008	0.009
61121U	0.146	0.147	0.147	0.010	0.010	0.010
61451L	0.008	0.008	0.008	0.000	0.000	0.000
61451M	0.099	0.099	0.099	0.006	0.006	0.006
61451U	0.227	0.227	0.228	0.008	0.008	0.008
62441L	0.010	0.010	0.010	0.000	0.000	0.000
62441M	0.130	0.130	0.131	0.007	0.007	0.007
62441U	0.286	0.286	0.286	0.008	0.008	0.008
64561L	0.045	0.045	0.045	0.005	0.005	0.005
64561M	0.287	0.287	0.287	0.010	0.010	0.010
64561U	0.496	0.496	0.496	0.009	0.009	0.009
66561L	0.183	0.183	0.183	0.015	0.015	0.015
66561M	0.551	0.552	0.552	0.013	0.014	0.014
66561U	0.692	0.692	0.692	0.010	0.011	0.011

Sensitivity: Partial correlation is a measure of the strength of the linear influence of a parameter on a given response, scaled to [-1, 1]. Table 53 shows the partial correlation of void fractions for five experiments with perturbations in temperature. Where negligible void fraction is predicted by the model, there is no correlation. However, where significant void fraction is present, there is nearly perfect negative correlation between perturbations in temperature and the model response.

Table 53. Partial correlation of void fractions for five experiments with ΔT.

	ΔT Partial Correlation
61121L	-0.906
61121M	-0.999
61121U	-1.000
61451L	0.000
61451M	-0.996
61451U	-0.999
62441L	0.000
62441M	-0.999
62441U	-0.999
64561L	-0.995
64561M	-1.000
64561U	-0.999
66561L	-0.999
66561M	-0.999
66561U	-0.998

DISTRIBUTION

1	MS0899	RIM-Reports Management	9532 (electronic copy)

CASL-U-2013-0048-000

Sandia National Laboratories

CASL-U-2013-0048-000