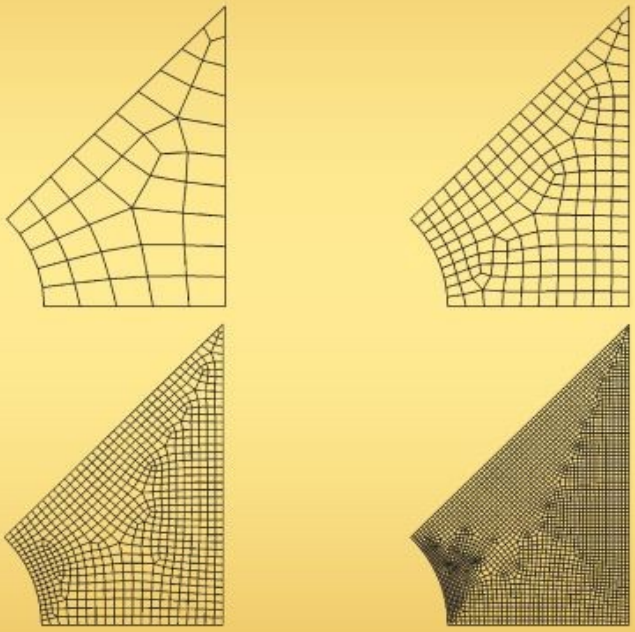
FUNDAMENTALS OF VERIFICATION AND VALIDATION



PATRICK J. ROACHE

Fundamentals
of
Verification and Validation

by

Patrick J. Roache

hermosa
publishers

P. O. Box 9
Socorro, New Mexico 87801
USA
http://www.hermosa-pub.com/hermosa

International Standard Book Number 978-0-913478-12-7
Library of Congress Catalog Card Number 2009929828
Printed in the United States of America

Cover Illustration

The cover background illustration, generated by Dr. K. Dowding, is from Figure 7.2 of *ANSI Standard V&V 20, Guide on Verification and Validation in Computational Fluid Dynamics and Heat Transfer* (2009) by ASME Committee PTC-61. The unstructured grid sequence was used to evaluate the effectiveness of the concept of effective grid refinement ratio for unstructured grids, as described in Section 6.4.2 of this book.

Warnings - Disclaimer

This book is intended to provide information in regard to the subject matter covered. Reprinted material is quoted with permission where required, and sources are indicated. It is sold with the understanding that the publisher and author are not engaged in rendering legal, accounting, professional engineering or other professional services. If legal or other expert assistance is required, the services of a competent professional should be sought. Efforts have been made to make this book as complete and accurate as possible. However, there may be mistakes both typographical and in content. Therefore, the text should be used only as a general guide and not as the ultimate source of information. The purpose of this book is to educate. The publisher and author have neither liability nor responsibility to any person or entity with respect to any loss or damage caused, or alleged to be caused, directly or indirectly by the information contained in this book.

If you do not wish to be bound by the above, you may return this book to the publisher for a full refund. Direct all inquiries to Hermosa Publishers, P. O. Box 9, Socorro, New Mexico 87801, USA.

Errata and Addenda

Errata and Addenda for this book are posted on *www.hermosa-pub.com*.

Dedicated to
Orestes Augustus Brownson, 1803 – 1876.

"He lived a life of integrity."

Preface

This book is essentially a second edition of *Verification and Validation in Computational Science and Engineering* (V&V1), although with about 1/3 new material it deserves the new title. As I begin writing, it is a decade since the first book was published, and a century since Richardson's first paper. The reception of the first book, published in 1998, has been gratifying. In the subsequent decade substantial progress has been achieved in three senses. First, new methods have been developed and evaluated, notably the Least Squares GCI (Grid Convergence Index) for Solution Verification by Eça and Hoekstra, and the application of the internationally accepted approach for experimental uncertainty to total Validation uncertainty as described in *ASME ANSI Standard V&V 20*. Second, the methods presented in the first book, already having been demonstrated to be successful, have been further applied extensively, notably the MMS (Method of Manufactured Solutions) for Code Verification, and the original GCI, which now has been accepted as a publication standard (not required) by the *ASME Journal of Fluids Engineering*. Third, there have been significant new publications[*], including the book on Code Verification by my colleagues Knupp and Salari, the ASME V&V 10 for Computational Solid Mechanics, the ASCE Monograph for V&V of Free Surface Flow Models (with a wealth of physical data and examples) the three Proceedings of the Lisbon V&V Workshops, the second edition of the *Handbook of Numerical Heat Transfer*, the (forthcoming) book by Oberkampf and Roy, and especially ASME V&V 20.

While attempting to maintain the organization of the first book where possible, I have noted significant departures. Notably, the Least Squares GCI required a new Section (5.11), and the V&V 20 methodology for Validation Uncertainty required a new Chapter (11), moving the old Chapter 11 on "Code Quality Assurance and Certification" to new Chapter 12 (abbreviated).

To alert those readers who have previously used the first book, I have noted with symbol Δ any Section that contain significant changes (new or corrected material), and with symbol § any completely new Section, Chapter, or Appendix. Several of the original Appendices have been deleted because of diminishing value.

The new Appendix D gathers the simplest V&V formulas together for easy reference. Combined with the new Chapter 11 on V&V 20 methodology for Validation Uncertainty, this should serve as a usable summary outline of the entire V&V procedure proposed herein.

The book layout is designed for readability rather than cosmetics. For example, page 199 is half blank so the reader can consider the next 3½ pages of graphics before reading the Section summary. I hope the reader appreciates it this way.

Errata and Addenda for this book are posted on *www.hermosa-pub.com*.

[*] Full citations will be given in Ch. 1.

Preface to V&V1

The title of this book, like most titles, is a compromise. Most of my experience, and therefore most of the examples in this book, are taken from that branch of Computational Science (or Computational Engineering, or Computational Physics, or Computational Mathematics) known as Computational Fluid Dynamics or CFD, and the closely related Computational Heat Transfer or CHT. Often, this term CFD is taken to apply to only computational aerodynamics, but I use it here more broadly to include computational aeroacoustics, groundwater flow and transport, ocean modeling, meteorological modeling, combustion modeling, free convection heat transfer, neutron transport, magnetohydrodynamics, etc. I also have some experience in computational electrodynamics, examples from which appear in this book as well, and some experience in (only) the V&V aspects Computational Solid Mechanics. However, the principles covered in the book are much more general, applying to virtually all numerical solutions of partial differential equations, including solid mechanics, structural dynamics, chemistry, etc., even some finance models. [In this second book, I have adopted the convenient description computational PDEs for computational partial differential equations]. Use of the term "partial differential equations" in the title would be descriptive but would have led to an overly long title, and perhaps would have seemed too mathematical. Much of the book would be properly covered in the title "Computational Mathematics", but "Validation" (as used herein - see Chapter 2) strictly speaking is not a mathematical issue (as is "Verification") but is an issue of physical science. Also, the related issue of "Certification" is most obviously an engineering term, related to somewhat specific engineering projects.

Within the broader realm of Computational Science and Engineering, or PDEs, I do not treat "real time" problems, e.g. on-line nuclear reactor simulations, which need robust input error checking. This is not a Verification and Validation issue in Computational Physics or Mathematics. (It certainly is a consideration for code quality, but not for Verification and Validation, as defined herein.) The term "Computer Science" as used in university departments would sometimes cover this material as well, but more often refers to issues of hardware and system software, rather than applications codes for partial differential equations as considered herein. The somewhat amorphous term "Computational Science and Engineering" includes applications codes but also (in one definition) is concerned with providing "total problem-solving environments" (i.e., computer systems), and this book would not qualify for such an inclusive definition. In the end, I chose the more inclusive title, and the text will use "computational PDEs" as the descriptive term covering partial differential equations in science and engineering.

The examples used herein obviously cannot cover the entire range of computational solutions of physically important partial differential equations. Some notable omissions are radiation equations, Kortweg-de Vries equations, molecular modeling, etc. But most readers of the *Journal of Computational Physics*, or the SIAM Numerical journals, or the AIAA and ASME and AIChE Journals, should have an interest in the book's subject.

Topics other than Verification and Validation are treated, such as Confirmation, Certification, Benchmarks, and Quality Assurance. These are related to but distinct from Verification and Validation. The terms are not as universally defined nor as centrally important as Verification and Validation. "Quantification of Uncertainty" is an excellent inclusive term; frankly, the main reason I did not use it is that I already have two other publications with that title. (Also, some authorities now suggest that it should not be used synonymously for errors that are not due to lack of physical information.)

The background required for a reader of this book includes a first level course in partial differential equations and basic numerical methods including some linear algebra and some numerical solutions for partial differential equations. It should not be necessary to be an expert in any particular discipline within computational PDEs. The methods and issues discussed are applicable to general purpose commercial codes, as well as specialized scientific codes. For the specific examples, it would be advantageous to have some background in *either* aerodynamics, fluid dynamics, heat transfer, groundwater flow and transport, or electrodynamics. I do not by any means intend to discourage applied mathematicians, but the discussions of validation would benefit if the reader had some minimal experience with *physical* experiments. I do not consider "CFD Lab" courses, as valuable as they may be, to be an adequate substitute for all physical experimental courses. (I understand that it is now possible at some universities to earn a bachelor's degree in engineering without ever taking a physical laboratory course; it is my opinion that the recipients of these degrees have been cheated in their education.)

Many readers will feel that I have not done justice to the extensive literature on error estimation in Finite Element methods. As described in Chapters 4 and 7, my opinion is that these methods are useful for guiding grid adaptation but less useful for the title subject of this book. More basically, I hope to excuse myself by quoting the legendary mathematical physicist S. Chandrasekhar, explaining why, in his classic *Hydrodynamic and Hydromagnetic Stability* (Oxford, 1961), he had no reference to viscous shear flow. "[T]his is a large and highly specialized field and I have not felt myself competent to write about it. This argument is indeed a general one: in the last analysis an author chooses to write about only those matters in which he has some confidence of his understanding."

I have taken two minor liberties when quoting sources. In this book, I have capitalized terms like Validation, Verification, Calibration, etc. in order to emphasize that the terms are used in a limited *technical* (vs. general) context - a major theme, especially in Chapter 2. To maintain consistent emphasis, I have capitalized these terms in quotations even when the original authors did not. Also, I have replaced underlines appearing in the original sources with italics, just because it looks better in type. The emphasis of the original authors is maintained.

In my reading of papers by respected colleagues in the title area, my experience has invariably been mixed. Even when reading what I consider to be the very best papers, those which provide significant insights and which I have quoted extensively in this book, I find many points of disagreement. Sometimes these points are minor or semantic, but often they are major and substantive. It has not seemed to be worthwhile, or even feasible, to discuss each of these points individually with citations of the original papers, as would be done in a review (e.g., for *AIAA Journal* or *Applied Mechanics Reviews*). I know for a fact that some of the disagreement is due simply to time-lag in the evolution of the concepts in the communities, and that many of the authors would now change their position (and terminology) upon further thought. This has certainly been my own experience. Extrapolating, I am confident that I will change my opinion on some of my ideas expressed herein. As a corollary, I am sure that this book contains internal inconsistencies. I ask the reader's tolerance.

Acknowledgments

It may surprise (and, if so, hopefully disappoint) some readers to discover that not everyone *cares* about numerical accuracy. I have been most fortunate in receiving research funding support over years from a minority in the applied mathematics, science, and engineering communities who believed it to be an important issue, not by giving just lip service, but also money and work. Foremost have been Larry Rapagnani, formerly of the U. S. Air Force Phillips Laboratory, Rip Anderson, Mel Marietta, and Mert Fewell of Sandia National Laboratories, Jim McCroskey of U. S. Army–Ames, and Tom Stauffer of the U. S. Air Force Armstrong Laboratories. Thanks to them for the years they supported my approach to numerical accuracy, i.e. they bought what I was selling. Likewise, Dr. Jack L. Melchor, through his Melchor Foundation, and the University of Notre Dame provided me with a Visiting Professorship in the Department of Civil Engineering and Geological Sciences for the fall of 1996, during which much of this manuscript was written. My thanks also go to Notre Dame Professors Joannes Westerink, William Gray, and Jerry Marley for setting up the visit.

For moral support and intellectual stimulation, I am deeply indebted to my colleagues at Ecodynamics, Sandia, Notre Dame, and throughout the CFD community, and to my wife, Dr. Catharine Stewart-Roache. I apologize for omissions, but these come to mind (listed alphabetically): Dan Aeschliman, John Bergstrom, Rebecca Blaine, Fred Blottner, Peter Bradshaw, Ismail Celik, Ray Chin, the late Tom Davis, Dave Dietrich, Roger Eaton, Kathy Economy, Dick Ewing, Mert Fewell, Joe Ferziger, Chris Freitas, Mohammed Gad-El-Hak, Kirti Ghia, Urmila Ghia, Bill Gray, Phil Gresho, Fred Habashi, Scott Hagen, Hank Happ, Daniel Haworth, Eric Jumper, Patrick Knupp, Benny Leonard, Eddie Leonardi, Clark Lewis, Robert MacKinnon, Mel Marietta, Steve McCormick, Jim McCroskey, Unmeel Mehta, Tom Mueller, Bill Oberkampf, Sam Paolucci, Dominique Pelletier, Steve Piacsek, Andrew Pollack, Joe Powers, Michael Rabbitt, the late John Rakich, Helen Reed, Art Rizzi, John Ruge, Tom Russell, Patrick Ryan, Kambiz Salari, Bill Saric, Tom Stauffer, the late Joe Steger, Stan Steinberg, Catharine Stewart-Roache, Demetri Telionis, Joe Thompson, Alex Treadway, Randy Truman, Graham de Vahl Davis, Paul Van Gulick, David Walker, Bob Warming, Joannes Westerink, Frank White, Dave Wilcox, David W. Zingg.

This work was partially supported by Sandia National Laboratories and the United States Department of Energy under Contract DE-AC04-76DP00789, Dr. M. G. Marietta and Dr. M. E. Fewell, Technical Monitors; by the United States Army Aviation Systems Command, Aeroflightdynamics Directorate under Contract NAS2-13618, Dr. W. J. McCroskey, Technical Monitor; by United States Air Force contract "Adaptive Grids for Raven", Dr. N. L. Rapagnani, Technical Monitor , and United States Air Force contract F08635-93-C-0085, Dr. T. Stauffer, Technical Monitor.

For this second book, the list of acknowledgements is extended to include the following colleagues: Luis Eça and Martin Hoekstra, the organizers of the exemplary Lisbon Workshops on V&V in 2004, 2006 and 2008, as well as all the participants; the committee for ASME PTC-60 (V&V 10), especially the

chairman Len Schwer; the committee for ASME PTC-61 (V&V 20), Hugh Coleman (chairman and instigator), Ben Blackwell, Kevin Dowding, Chris Freitas, Urmila Ghia, Richard Hills, and Roger Logan; the committee for ASCE V&V for Free Surface Flows, Sam S. Y. Wang (chairman), Yafei Jia, Richard Schmaltz, and Peter Smith; and Jeffrey Kasser, Richard Sonnenfeld, Thomas Zang, Dale Pace, Christopher Roy, Francois Hemez, James Kamm, Gerry Brock, Tim Trucano, Jon Helton, Joseph Morrison, and Vicente Romero. I have again been fortunate to have some funding from someone who cares about accuracy: Richard Schultz at Idaho National Laboratories.

TABLE OF CONTENTS *

* Symbol **Δ** indicates a Section with significant new material added to the first edition.
Symbol **§** indicates a completely new Section, Chapter or Appendix.

PART I

OVERVIEW

Part I will present an overview of the subject, approached by way of historical development. The context in all of this book is that of non-real-time numerical solution of partial differential equations. The history will include not only ideas but attitudes, and involves several controversies. Considerations of terminology and semantics have proven to be essential, as attested by confusion in the literature. However, unraveling the semantics can be tiresome, and the reader could proceed directly to the more substantive material in Part II if preferred.

CHAPTER 1

INTRODUCTION

1.1 SKETCH OF HISTORICAL DEVELOPMENT OF COMPUTATIONAL SCIENCE AND ENGINEERING

As noted in the Preface, the subject of numerical solution of partial differential equations or Computational Science and Engineering, is broad. I will use the description "computational PDEs" (for computational partial differential equations) to include computational fluid dynamics, aeroacoustics, heat transfer, groundwater flow, electrodynamics, plasma dynamics, solid mechanics, structural dynamics, heat conduction, neutron transport, groundwater flow and transport, ocean modeling, meteorological modeling, combustion modeling, free convection heat transfer, neutron transport, magnetohydrodynamics, chemistry, enhanced oil recovery processes, etc., even some finance models. The methods and issues to be discussed are equally applicable to general purpose commercial codes and to specialized scientific codes.

The historical roots of computational PDEs can be traced to Richardson's 1910 paper on the computation of stresses in a masonry dam. Although performed with hand calculations by "computing boys," paid piecework, the work was truly multidimensional, and the paper itself is highly recommended reading for any practitioners interested in their own cultural heritage. See also the brief history in Roache (1972b, 1998b). The paper by Emmons (1970) was a landmark study for its systematic approach. The popular science article by Harlow and Fromm (1965) alerted many in the scientific community to the potential of the computer simulation approach, presenting at least qualitatively accurate solutions of previously intractable problems without the constricting assumptions used by theoreticians. By the time of the publication of the first book with "CFD" in the title (Roache, 1972b) it was beginning to be realized that we were in the midst of a revolution, in fluid dynamics and indeed in all areas of science and

engineering. The traditional categories of "theoretical" and "experimental" were being forced to make way for a third category: "computational." For a while, people tried to fit the computational work into the old "theoretical" category, but it did not work. As I wrote in 1972, "it is this author's contention that computational fluid dynamics is a separate discipline, distinct from experimental fluid dynamics and from theoretical fluid dynamics, with its own techniques, its own difficulties, and its own utility."

The revolution was enabled by the phenomenal progress in computing power, which continues today. Yet it cannot be emphasized too much that progress was and is *equally* achieved by improvements in algorithms. For example, Cramer's method for solving linear systems of equations, which I was taught in high school and undergraduate school, is so inefficient for large systems that its operation count is literally astronomical. For modern problems with $O(10^6)$ grid points, the algorithmic improvement over Cramer's rule (and other primitive algorithms, e.g. the naive discrete Fourier transform replaced by the modern Fast Fourier Transform) can overshadow the hardware increases spanning the age of the abacus to parallel architecture supercomputers.

Nevertheless, we are now living with the historical heritage of marginal computing power. As computing power increased exponentially, perhaps doubling every two to three years, the vision of the practitioners increased even faster, abetted by the increase in complexity of the physics modeled and by the dimensionality. If one could solve (to at least qualitative accuracy) a two-dimensional incompressible steady laminar separated flow problem on a grid of $O(100 \times 100)$ nodes at some calendar date (say 1970), then one could swallow up a two-order of magnitude increase in computing power by going to three-dimensional flows, another two orders of magnitude by going to time-dependent flows, another one order of magnitude by going to turbulent flow with Reynolds-Averaged Navier-Stokes (RANS) equations and a two-equation model of turbulence, another order of magnitude with a Large Eddy Simulation of turbulence, another one or two orders of magnitude with brute-force calculation of turbulence by Direct Numerical Simulations (DNS) in square channels at low Reynolds numbers. Further generations of computing power can be eaten up, several orders of magnitude at a time, by increases in Reynolds numbers and by engineering geometries (easily one order of magnitude, two for full aircraft). Grid resolution also has a healthy appetite; for a three-dimensional time-dependent problem, a factor of 2 increase in computer speed is eaten up by merely a 19% increase in resolution, even assuming optimally efficient methods (in which computing cost is merely proportional to the number of unknowns). In fact, the commonly used methods are far from the promise of optimally efficient methods promised by true Multigrid Performance (Brandt, 1977), and the result is another order of magnitude penalty for increased resolution.[1] Even with optimally efficient methods, an order of magnitude increase in computer speed does not allow even a doubling of resolution, but only a 78% increase. There is more. Going from *analysis* to *design optimization* is good for a minimum of one order of magnitude for even the simplest system with one or two design parameters; two or even three orders of magnitude are more realistic. In groundwater flow and transport, a geologic disposal site for radioactive waste originally designed with back-of-the-envelope calculations was finally analyzed with *thousands* of geostatistical realizations of property fields and scenarios, each calculation involving two-dimensional and three-dimensional time dependent calculations with three different computational PDEs codes (WIPP PA Dept., 1992). Detailed geochemistry calculations in groundwater remediation studies, coupled with groundwater flow, could saturate any conceivable computer system. As is now well

[1] This penalty is optimistic because it applies to increments. The base comparison is terrible. True Multigrid Performance (Brandt, 1977) somewhat idealistically set the benchmark at 10-20 iterations to achieve convergence, regardless of problem size. But good Multigrid codes have indeed achieved $O(100)$ even for large and difficult problems. Currently, aerodynamics publications shamelessly cite $O(10{,}000)$ to $O(100{,}000)$ iterations. The Multigrid literature is enormous. For references to earlier literature, see McCormick (1989) or Roache (1998b, Ch. 10).

recognized, the post-processing of data, especially from time-dependent simulations, demands more computing power than previous levels of simulations. That is, the computer resources devoted in 2000 to visual post-processing probably exceeded the power devoted to performing the simulations themselves in 1980. In many installations, "antique" computers purchased 3 to 5 years earlier as computing engines are kept around for post-processing tasks.

The point is that the universe of possible problems is so extensive, and the power of simulations is so great, that practitioners have often focused on *qualitative* simulation of the *next* more difficult problem class, rather than on achieving *quantitative* accuracy on the *previous* problem class. The result of this situation sometimes has been to cause a *decrease* in the quality of simulations published.

It is easy to be sympathetic to the wave of virtually unbridled optimism accompanying this revolution. In the aerodynamics community, the touchstone was the 1975 article in *Aeronautics and Astronautics* by Chapman, Mark and Pirtle (1975), which predicted (and called for) computational PDEs (specifically CFD) to "supplant" wind tunnel experiments. The lack of restraint was perhaps due in part to the natural necessity of "selling" the CFD potential to management, composed as always of an older generation of technocrats who were not directly involved in the newer technical field of CFD, and who were more inclined to skepticism than to enthusiasm, who held the purse-strings, and who were to make the decisions about relative funding of CFD vs. experimental facilities.

To the credit of the CFD community, the ensuing debate was not simply characterized by "us vs. them," i.e. by CFD practitioners vs. experimentalists. From the CFD community, there were two immediate and critical responses to the Chapman et al article, one by Bradshaw (1975) and one by myself (Roache, 1975). Bradshaw pointed out the naiveté of the "supplanting" philosophy especially in regards to turbulence modeling, while I emphasized the limitations inherent in the mathematical approximations. Interestingly, we now recognize these as issues of the two title categories of this book, i.e. Validation and Verification, respectively. (See Chapter 2 for definitions.) This more cautious and realistic philosophy was reiterated by other conscientious CFD practitioners in a division of opinion that persisted for years, arguably to this day. Especially noteworthy were the editorial comments of three successive editors of the Proceedings of the AIAA Computational Fluid Dynamics Conferences. J. Rakich (1973), R. T. Davis (1975), and F. G. Blottner (1977) each took unequivocal stands against the overly optimistic view of Chapman, Mark and Pirtle (1975). (Rakich, 1973 anticipated the latter article.) Each explicitly stated that wind tunnels would not be replaced by computers, invited participation by experimentalists in the CFD conferences, and called for collaboration between computational, experimental, and analytical fluid dynamicists. Implicitly, they took stands against the even more aggressively optimistic salesmanship of others during this time.[2]

At the round table discussion of the 1988 AGARD conference on Validation in CFD (AGARD, 1988), H. Hornung noted approvingly that numerical performance had improved over previous years, yet the claims were more modest, and that "reduction of the extreme claims is a very healthy sign." In the same round table discussion, P. Sacher noted that "Nobody talks anymore of replacement of wind tunnels by the computer. I am really happy to find that."

The salesmanship persists today in the arena of CFD commercial codes. There exists a deleterious Darwinistic effect here. Most potential customers want to hear only good news. The more conscientious salespeople will point out limitations of their products, and often lose sales to less scrupulous salespeople. The best support one can give to the conscientious salespeople is to encourage enlightened customers, which is a thrust of this book.

[2] An anonymously authored parody of the situation appeared in the mid 1980's, and was reproduced in Appendix D of V&V1 with additional comments.

The division of opinion persisted but not so much in a spirit of debate. Unfortunately, as is often the case in human affairs, the opinions solidified and there arose separate camps that did not much talk to each other. The effects were very significant, with the optimists largely winning the day in the management struggles. While arguably good for computing power and computors like myself, the result for fluid dynamics as a whole was deleterious. Aeschliman et al (1995) also attribute to the Chapman, Mark and Pirtle (1975) article the widely recognized historical trend (especially in the U.S.A.) of reduced funding for experimental work, to the point where irreplaceable facilities were scrapped, graduate students went unfunded, and cultural skills were almost lost. As they noted, it became possible at some colleges for one to obtain a Bachelors Degree in engineering or physics and "never witness, let alone actively participate in an actual experiment," the genuine physical laboratory courses having been totally "supplanted" with "CFD Labs." We obviously believe in CFD education and CFD Labs, but We agree with Aeschliman, Oberkampf, and Blottner (1995): "Modelers who have no laboratory experience are unlikely to appreciate the true complexity of the real world."

Aeschliman, Oberkampf, and Blottner (1995) also noted that the optimistic predictions of Chapman, Mark and Pirtle (1975) failed to materialize, even though the actual increase in computing power exceeded their requirements. Likewise, Rizzi and Vos (1996), noted "the fundamental question of how well available RANS codes and turbulence models predict the maximum lift of a simple transport aircraft remains largely unanswered." Also, Aeschliman et al (1995) noted that the continued closing of experimental facilities would result in "increasing dependence on new and unvalidated CFD codes for solutions to the most difficult remaining flow problems. The National Aerospace Plane, which was to be designed and developed with a very heavy reliance on CFD, and the ensuing programmatic fiasco surrounding NASP after the truth became known, was not necessarily a unique event."

1.2 THE NEW IMPETUS TOWARD HIGHER QUALITY SOLUTIONS

The Stanford Turbulence "Olympics" (Kline et al, 1981) made a tremendous effort to establish high standards of turbulence simulations, and the organizational aspects still stand as a paradigm for this type of workshop or community endeavor (e.g., see Rizzi and Vos, 1996). Unfortunately, the first conclusion of the Review Committee (chaired by Howard Emmons) was that the numerical quality of the solutions was so poor that it was impossible to draw meaningful conclusions about the relative merits of various turbulence models. This same story has been repeated many times in conferences, symposia and workshops since then.

Needless to say, there have always been some individuals and groups who produced high quality, reliable CFD calculations. Two decades before V&V1 was written, Blottner (1977) stated the present issue with clarity. After distinguishing the two distinct sources of error from "either inappropriate governing equations or inaccurate numerical solution procedure" (i.e., the issues now named Validation and Verification; see Chapter 2), Blottner introduced the term "credibility" and noted its necessary requirement: "Computational fluid dynamicists must perform grid refinement studies whenever they present numerical studies as experimentalists are expected to put error bounds on their results. Only if this is done can the needed credibility of computational fluid dynamics be established."

However, individuals like Blottner were the exception, and the general level of quality even as late as the mid-1980's was unacceptable, in our opinion. This evaluation is based not only on reading of final published articles, but on my experience on the Review Committee of the Stanford Turbulence "Olympics" (Kline et al, 1981) and as an editor and reviewer. In 1985, Frank White, the Editor of the American Society of Mechanical Engineers (ASME) *Journal of Fluids Engineering* created the position of Associate Editor for Numerical Methods, which formally recognized the special needs of this discipline. In 1986, we published (Roache, Ghia and White, 1986) in *JFE* an "Editorial Policy Statement on the Control of Numerical Accuracy" which instituted "higher standards on the control of numerical accuracy." Since the

rationale and needs (in 1986) for the policy statement are explained in the announcement, it is repeated below.

1.2.1 ASME Journal of Fluids Engineering 1986 - "Editorial Policy Statement on the Control of Numerical Accuracy"

A professional problem exists in the computational fluid dynamics community and in the broader area of computational physics. Namely, there is a need for higher standards on the control of numerical accuracy.

The numerical fluid dynamics community is aware of this problem but, although individual researchers strive to control accuracy, the issue has not to our knowledge been addressed collectively and formally by any professional society or journal editorial board. The problem is certainly not unique to the JFE and came into even sharper focus at the 1980–81 AFOSR HTTM-Stanford Conference on Complex Turbulent Flows. It was a conclusion of that conference's Evaluation Committee that, in most of the submissions to the conference, it was impossible to evaluate and compare the accuracy of different turbulence models, since one could not distinguish physical modeling errors from numerical errors related to the algorithm and grid. This is especially the case for 1st-order accurate methods and hybrid methods.

The practice of publishing comparisons based on coarse grid solutions, without systematic truncation error testing, may have been acceptable in the past. Certainly 10–15 years ago any calculation was of interest, and much of the exploratory work deserved publication, as many researchers lacked the computational power or funds to do a thorough and systematic error estimation. We are of the opinion that this practice, however understandable in the past, is outmoded and that, with powerful computers becoming more common, standards should be raised. Consequently, this journal hereby announces the following policy:

The Journal of Fluids Engineering will not accept for publication any paper reporting the numerical solution of a fluids engineering problem that fails to address the task of systematic truncation error testing and accuracy estimation.

Although the formal announcement of this journal policy is new, it has been the practice of many of our conscientious reviewers. Thus the present announcement is not a change in policy so much as a clarification and standardization.

Methods are available to accomplish this task, such as Richardson Extrapolation (when applicable), calculations with a high- and low-order method on the same grid, and straightforward repeat calculations with finer or coarser grids. As in the case of experimental uncertainty analysis, "…any appropriate analysis is far better than none as long as the procedure is explained." Whatever the authors use will be considered in the review process, but we must make it clear that *a single calculation in a fixed grid will not be acceptable,* since it is impossible to infer an accuracy estimate from such a calculation. Also, the editors will not consider a reasonable agreement with experimental data to be sufficient proof of accuracy, especially if any adjustable parameters are involved, as in turbulence modeling.

We recognize that it can be costly to do a thorough study, and that many practical engineering calculations will continue to be performed on a single fixed grid. However, this practice is insufficient for publication in an archival journal.

1.2.2 Δ Later Policy Statements and Other Initiatives

Although apparently innocuous, this policy was not universally welcomed nor was it implemented without difficulty. See Appendix A, a reprint of Roache (1990), for my experiences with the implementation of this policy, including the objections put forth on its practicality. The *ASME Journal of Heat Transfer* implemented a similar policy in 1994 (see Appendix B). In 1988 and 1989, there were special sessions at the ASME Winter Annual Meetings on numerical accuracy. In June 1989, Clark Lewis organized a session of invited papers at the AIAA Thermophysics Conference on CFD "code validation/verification/certification." The papers from this well attended session were published in an issue of the *AIAA Journal of Spacecraft and Rockets* (Vol. 27, No. 2, March-April 1990). In his introductory comments as *JSR* editor-in-chief, Lewis (1990) noted that "computational fluid dynamics has reached the age where those who practice it and publish the results of code development and applications must demonstrate the accuracy and value of their work." In October 1989, Tinsley Oden organized a workshop at the Texas Institute for Computational Mechanics on the somewhat broader issue of reliability. In the early 1990's, the ASME Coordinating Group on Computational Fluid Dynamics sponsored several Symposia on the general subjects of "Quantification of Uncertainty in CFD" and benchmarking, with motivation principally provided by Ismail Celik and Christopher Freitas; e.g., see Celik and Freitas (1990), Celik et al (1993), Freitas (1993b, 1995b), Johnson and Hughes (1995). In 1994, the *International Journal of Numerical Methods in Fluids* published its statement (Gresho and Taylor, 1994). In 1993 the AIAA Fluid Dynamics Technical Committee, with the encouragement of the Chairman David Walker, passed (almost unanimously) its recommendation for a similar policy, which was adopted for *all* AIAA publications and first published in January 1994. (See also Appendix B of V&V1[3] for reprints.)

The AIAA Policy Statement was admirable on at least four accounts. First, it included requirements for both computational and experimental uncertainty in a single statement, thus indicating that nothing special is being required of the computational work beyond simple good engineering practice. Second, the policy statement is to be re-published yearly. Third, the statement is included with instructions to journal manuscript reviewers. Fourth, the core statement is now included in the "Information for Contributors to Journals of the AIAA" published in every issue. An expanded version was published in AIAA (2006).

In 1994, the impetus for quality work reached a new plateau. The ASME *Journal of Fluids Engineering* editor, D. P. Telionis, accepted the recommendation of the ASME Coordinating Group for CFD and its Chairman, C. J. Freitas, and published a much expanded and more specific Policy Statement (supplanting Roache et al, 1986) which included the categorical rejection from publication of 2-point first-order upstream methods. That policy statement and the introduction by Freitas were reproduced in Appendix B of V&V1[1]. Once again, and more understandably, the new policy caused considerable discussion; see Leonard (1995a,b), Leonard and Drummond (1995), Freitas (1995a), Roache (1995). In 1998, the AIAA CFD Committee on Standards published the *Guide for the Verification and Validation of Computational Fluid Dynamics Simulations* (AIAA, 1998), and in 2006 the ASME Committee PTC-60 published *ASME V&V 10-2006. Guide on Verification and Validation in Computational Solid Mechanics*, (V&V10), inspired by the AIAA Guide. These are both valuable over-views of the V&V processes and philosophies (differing sometimes significantly from those of this book) and are probably most valuable for V&V management of large engineering projects, but they do not present any procedures (e.g., they contain

[3] Shorthand references will be used for four publications: V&V1 for the first edition of this V&V book, Roache (1998a); V&V10 for ASME Committee PTC-60 (2006), now known as ASME Committee V&V 10; V&V20 for ASME Committee PTC-61 (2009), now known as ASME V&V 20; and AIAA Guide for (AIAA, 1998).

no equations). The book by Oberkampf and Roy (2010) is also recommended especially for current views on management issues (although my opinion differs from theirs on some points).

In 2005 the journal *Clinical Biomechanics* published its first policy statement on V&V (Burton et al, 2005). Although this initial policy statement confused some V&V terminology, it was a significant contribution to setting quality standards for CSM (Computational Solid Mechanics) in difficult biomechanical problems. For example, Validation is complicated by the aging of animal tissues during the experiments.

In 2009, the ASCE/EWRI Task Committee published *3D Free Surface Flow Model Verification / Validation*, (ASCE/EWRI, 2009) that contains detailed presentations of the procedures for verifications of codes and calculations, and extensive experimental data for benchmark exercises. Finally, in 2009 the ASME Committee PTC-61 published *ASME-ANSI Standard V&V 20, Guide on Verification and Validation in Computational Fluid Dynamics and Heat Transfer* (V&V20). This document covers all the details of the entire V&V process, including the refinement of the Validation Uncertainty concepts initiated by H. Coleman based upon internationally accepted terminology and concepts used in experimental work, including ASME PTC-19.1 (ASME 2005). (I had the pleasure of working on the ASME and ASCE/EWRI committees that produced these last three publications.) The new Chapter 11 of this book presents a short version of this V&V20 Validation Uncertainty concept, which is a significant addition to the first edition of this book (V&V1).

The Grid Convergence Index or GCI (see Chapter 5) presented in the first edition now has been exercised on many hundreds of cases and has proven to be reliable. An important single paper is that of Cadafalch et al. (2002); see also Roache (2003b). Because of this extensive experience, the GCI has been recognized as a standard approach for reporting numerical uncertainty by the *ASME Journal of Fluids Engineering* (Celik et al., 2008) and in the ASME-ANSI Standard V&V 20. A significant addition is the Least-Squares version of GCI developed by Eça and Hoekstra, as well as the other V&V methods evaluated in their exemplary three Lisbon Workshops of V&V (see Section 5.11 for references).

In a visionary editorial in the ASME *Journal of Fluids Engineering*, Telionis (1995) described how the revolution in Internet information technology can be used to favorably affect all these issues and broader ones of the review process and publication lag, critical comments, author revisions, etc. For example, gory details of Verification (see the succeeding chapters) could be relegated to a longer electronic version of a paper, with only a short summary statement in the printed version. Besides shortening the printed version and removing clutter for readers interested primarily in the results, the electronic mode provides another tremendous advantage. No longer will authors have to struggle to reduce the electronic paper to 6 pages, so no longer will they have that excuse for not providing convincing evidence of grid convergence. Reviewers can, in good conscience, be more demanding of numerical quality. Likewise for the expository aspects of the papers. Electronic versions can provide a multi-level publication, and the longer, more leisurely versions can avoid what G. H. Douglas has called the "cobblestone writing" style, in which the requirement for brevity fosters staccato statements of fact – subject, verb, object; subject, verb, object; ...– one after another, like an old wagon bouncing along on a cobblestone street from one hard unyielding fact to the next. Wouldn't it be wonderfully ironic if electronic on-line publishing became responsible for a return to Victorian prose style?

1.2.3 Δ Commercial Codes and Users

New aspects of the problem are caused by the now extensive use of general purpose commercial codes. In 1972, when my first CFD book was published, there *were no* general purpose CFD commercial codes. A third of a century later, Hutton (2006) cited estimates of the worldwide number of commercial CFD code users of 25-30,000, including 2/3 of the Fortune 500 companies, and licensing revenues growing at 15-20%

annually. From a V&V viewpoint, this situation is intimidating. Industry colleagues tell me that the Code Verification exercises provided to users by vendors are neither as convincing for the individual cases as the examples given here in Chapter 3, nor as extensive in option coverage as users would like. With recent corporate consolidation of commercial CFD vendors, the Code Verification quality is not likely to improve.

In 1972, a few researchers had distributed their codes (including source code) to a select audience. Usually, the authors had negative experiences with misuse and mostly became discouraged with the idea of wide use of their codes. The codes were used without sufficient knowledge of either the sensitivity to iteration convergence parameters or the fluid dynamics, were applied without conscientious grid resolution studies, and were applied outside the regime of physical parameters intended. In the worst cases, the users modified the source code, reducing the accuracy, and then published inaccurate results attributed to the original code author. In fairness to the users, it must be noted that the code documentation (both internal and external, and both theory and users manuals) was usually atrocious; only slowly did the CFD community develop an appreciation for the significant work required to produce adequate documentation. In the groundwater flow discipline, in 1988 the U. S. Geological Survey published and released into the public domain its code which became known universally as MODFLOW (McDonald and Harbaugh, 1988). Its clear documentation, both internal and external, became the standard of performance and contributed to the code's immense popularity and use.

The difficulties change and compound as one considers in turn

(a) research codes for a limited class of problems, developed and used only by the authors and their close colleagues (described as the "cottage industry" phase by Kleb and Wood, 1988),
(b) public domain codes for a limited class of problems and for which source code is released,
(c) commercial codes for a limited class of problems and for which source code is not accessible, and,
(d) the toughest case, general purpose commercial codes for which source code is not accessible.

The methods for Verification and Validation (V&V) to be described in this book are applicable to general purpose commercial codes, but the user must have confidence that the numerical methods as described in the manuals are actually those implemented in the code, with no undocumented special case "switches," "fixes," etc. This is especially true in the difficult area of turbulence modeling. Even for thoroughly documented codes, the most conservative Verification techniques may be required because of the unfortunate prevalence in general purpose commercial codes of using "hybrid" methods. The details will be discussed in later chapters; for now, suffice it to say that CFD algorithm developers have long known that there is a trade-off between code robustness and accuracy, and CFD code marketers know that there is little market for numerical accuracy but much demand for bullet-proof code robustness. General purpose CFD codes must be treated with skepticism in any new application by any conscientious user. See the results of early "CFD Triathlons" organized by C. J. Freitas (Freitas, 1993c, 1995b) for an indication of use of CFD general purpose commercial codes on *laminar* problems. This skepticism is supported by the experiences of the three Lisbon V&V Workshops (Eça and Hoekstra, 2004, 2006, 2008; Eça et al, 2005, 2007, 2009). Also, note that it is very difficult to devise test cases for non-trivial industrial problems and to obtain quality data for Validation comparisons; e.g., see the CFD World Users Association publications (Muller, 1994; de Vahl Davis, 1995; Muller and Loffler, 1996). See also Chapter 9.

In the same vein, the issue of user education must be addressed - not education in code use, but in whatever scientific discipline is involved (e.g. fluid dynamics). It is asking too much of a code vendor to make up for lack of technical training (or plain sloppy thinking) of a user. A CFD code is not an "aerodynamicist on a chip" or a "groundwater analyst on a chip." The folklore of both communities abound with horror stories of technically unqualified users who care little for the accuracy of the results, and who obtain from CFD codes answers that are either patently absurd, or that are correct but could have been obtained with elementary calculations by hand or handbook.

It is apparently the case that user education is more of a practical concern in fluid dynamics than in some other disciplines, notably structural analysis. It is obvious that there is no analogue of fluid turbulence in structures, but the more significant distinction may be that the simplest level of analysis for structures (involving linear stress-strain, small deformations, steady state, etc.) happens to be very useful, whereas the analogous level in fluid dynamics (potential flow) by itself has few practical applications. However, Prof. G. Sinclair (2006) has pointed out how common it is to ignore the major effects of corner singularities in FEM solid mechanics stress calculations.

Likewise, user (and management) expectations are often unrealistic, and should be corrected whenever possible. In this regard, let me state the bad news right now.

It is totally unrealistic to expect that an analyst can approach a new problem (not close to a previously calculated problem) that is difficult (not potential flow, etc.) and expect to achieve reliably accurate answers with a single calculation.

1.3 PERSONAL ANECDOTE ILLUSTRATING THE IMPORTANCE OF SYSTEMATIC CODE VERIFICATION

I would like to introduce and hopefully motivate the subject of code Verification (and more generally, reliability) with a personal anecdotal example (Roache, 1982). The subject area is not fluid dynamics, but electrostatics.

In the mid 1980's, working with colleagues at the Tetra Corporation, I developed the ELF (for **EL**ectric **F**ield) codes for the design of laser electrodes and high power switches (Roache et al, 1984). Later (after the experience described below) commercial versions known as the TetraELF codes were marketed. The problems involved boundary fitted coordinate generation in two-dimensional and three-dimensional geometries using elliptic grid generation methods (Thompson et al, 1974), and variants by Steger and Sorenson (1979) and Thomas and Middlecoff (1980), solved by semidirect elliptic marching methods (Roache, 1995). The codes ultimately included time-dependent strongly nonlinear source terms, solution adaptive grid generation, interior boundaries of dielectric materials, and other complications (Roache et al, 1984).

An early production version of the two-dimensional ELF code certainly produced plausible, intuitively appealing solutions, both for the nonorthogonal boundary-fitted grid generation and for the solution of the "hosted" equations for the electric potential ϕ. The code had been exercised over a variety of problems for several years. It had failed to converge in the boundary-fitted grid generation task only on one class of electrodes with a slit-like geometry, which failure was inexplicable. Other than this failure to achieve a solution (which is not the same, nor as dangerous, as achieving a false solution) the codes were used with much confidence; the answers looked good, even compared to an exact solution.

I had ostensibly benchmarked the code by comparison to the classical solution for the Rogowski family electrode case (Lorrain and Corson, 1962) based on traditional complex variable theory, and restricted to linear steady two-dimensional problems (vacuum solution). The code solved the (transformed) Laplacian equation for potential ϕ with an error of 0.4% using only a 13×13 cell grid. This I considered excellent performance for a relatively coarse grid, and I believe that most practitioners would be confident in such a code. Certainly, many codes are in use that have not been benchmarked as well and are used with confidence.

My colleague at Tetra Corporation, Henry J. Happ, was not so confident. I patiently explained how the code could not be expected to produce the exact results, that 0.4% was to be expected for this coarse level of discretization, and was certainly adequate for engineering design and analysis. Ignoring my explanation and greater experience, Happ proceeded to perform a systematic grid convergence study of the old

fashioned kind, described for one-dimensional problems in any old numerical analysis textbook. This required considerable work for a two-dimensional problem using the computers of the day. Starting with a more coarse grid of 5×5 cells, as the grid was refined, the solution converged *at the expected rate* for a 2nd-order numerical method. Unfortunately, it did not converge to the correct answer. The error from the Rogowski solution was reduced, as expected, but the error did not approach zero. Rather, the asymptote for the error appeared to be about 0.2%. The code accuracy was zeroth order, i.e., wrong.

Chastened by this numerical experiment, I investigated the source code thoroughly, looking for some subtle error, perhaps in a boundary condition implementation near the logical corners of the grid, that might cause this small error in the results. What I found instead was a *gross* error, namely, a factor of 2 error in the stencil coefficient for the cross derivative term in the transformed equations. The error appeared in both the equations for the grid generation, and in the discretization of the hosted equation. (At least I had been consistent!)

When the code was corrected, the solution error did asymptote to zero. The factor of 2 error in cross-derivatives affected the grid generation by less than 0.01% in the location of any x and y of the nodes, and the electric field by 0.016%. Note that the electric field, being the gradient of the solution for the potential ϕ, is a difficult quantity to predict since the maximum occurs at the curved boundary.

This insensitivity of the solution and its gradient to a gross error in coding was an affront to intuition. However, intuition was salved by re-running the previously failed grid generation for slit-like geometries. With the error, iteration convergence of this nonlinear problem (the nonlinearities are introduced by the coordinate transformation) was obtained only with extreme measures of 20 continuation steps plus the use of extensive under-relaxation. The resulting "mesh" was a "mess," with coordinate lines that crossed and extended outside the physical domain. When the coding error was corrected, the code converged to a good grid with only 2 continuation steps.

How could such a gross error have escaped detection in so many previous exercises? The answer is, the same way that such a gross error could make so little difference (maximum 0.016% in the gradient) in the solution; *the test problems had not sufficiently exercised the terms in the equations*.

The insensitivity originated in the mathematics of the grid generation procedure, which was based on the homogeneous Thompson-Thames-Mastin (Thompson et al, 1974) or the original Winslow (1967) method. For this generator, the continuum interior equations are identical to those of an orthogonal and indeed *conformal*, angle-preserving transformation. The discrete coordinate transformation generated is close to the conformal transformation, differing only because of truncation error and the distribution of boundary values of the grid. (The boundary nodes were located by equidistribution in arc length, but this distribution is not far from that of a conformal transformation grid.) *For a conformal transformation, the cross derivative terms in the transformed Laplacian operator are identically zero*. Thus, the gross error of a factor of 2 occurred in terms that were very small for the problem studied, and thus escaped detection when only absolute size of the error for a single grid solution was examined. However, when a *systematic grid convergence study* was performed, the presence of an error was quickly indicated. My experience has been that such systematic grid convergence testing is remarkably sensitive; other examples will be given in Chapter 3.

Why is not this kind of systematic grid convergence study used more often? Certainly, it has been used since the early 20th century in ordinary differential equations (ODEs). Virtually anyone who has taught numerical methods for ODEs has had the experience of students obtaining a good answer with their own Runge-Kutta integration code, but observing that the method does not converge with 4th-order rate on their problem, but only at 2nd-order rate. (Their are half a dozen errors in understanding and coding the RK4 algorithm that yield a consistent 2nd-order method but are still in error.) I believe that the bad habit of not performing systematic grid convergence studies to Verify the convergence rate of a multidimensional code developed historically because of limited computer resources. We certainly now have adequate resources

available to do the job, but practitioners have lowered the standards from the old days of ODE solutions and are often content with superficial Verification exercises, or none at all. (Another reason was the lack of general exact solutions for difficult PDEs like the Navier-Stokes equations. This reason is now obsolete; see the Method of Manufactured Solutions in Chapter 3.)

What was frequently done was to appeal directly to agreement with physical experiments. (In the terms to be described more specifically in the next chapter, practitioners passed over Verification and went directly to Validation.) The general inadequacy of this practice is clear, considering the previous example on an important class of problems with only a 0.4% error arising from a gross coding mistake. In many experiments (certainly most aerodynamics and all groundwater flow experiments) this level of error would not be detectable with confidence. It is now the dominant opinion (see citations in Chapter 10) that there is a "continuing need for high quality experiments that are designed specifically for CFD code [Validation]" (Melnick et al, 1996).

It is clear to all who have thought and wrote seriously about V&V that

- Verification and Validation are separate activities,
- Verification should precede Validation, and
- These two terms had best be defined with some specificity in a technical context.

Verification and Validation must be defined, distinguished, and contrasted to other activities in the arena generally described as "confidence building" or "Quantification of Uncertainty" These necessary semantic distinctions are the subject of the next chapter.

1.4 § V&V CREDIBILITY CHECKLIST FOR NON-SPECIALISTS

Conscientious scientists and engineers will want to do credible V&V for their computational work but are not usually V&V specialists. Understandably and properly, they are more concerned with their own technical interests. The following (partial) checklist of Table 1.4.1 is provided both as a reminder to check before publications or presentations, and as motivation for topics to be covered in this book. Being careful in terminology will make you more credible and will be easier on your audience.

- Do not equate V&V with Quality Assurance (QA).
- Do not equate V&V or QA with organization charts or QA forms.
- Distinguish between Verification and Validation.
- Distinguish between Verification of Codes vs. Verification of Calculations (Solutions).
- (Note from the two items above that V&V comprises three subjects, not two.)
- Do not claim impossibility of Code Verification because there are no nonlinear exact solutions.
- Do not confuse Calibration with Validation.
- Distinguish between *numerical uncertainty* and *numerical error estimate*.
- Do not say just *uncertainty* when you mean *parametric uncertainty*.
- Do not confuse *parametric uncertainty* in Validation with that in design studies.
- Do not just say *convergence* -
- distinguish between *iteration convergence* and *grid convergence* (or *discretization convergence*)
- Do not assume that *grid convergence* means *grid doubling*.
- Do not assume that *grid convergence* means *grid refinement* rather than *grid coarsening*.
- Do not assume that *cost* of grid convergence scales linearly with number of parameters.
- Do not assume that trend predictions from coarse grid studies will be qualitatively correct.
- To really impress your audience, distinguish between *strong* and *weak* senses of *model*.
- Experimentalists: always present estimates, however rough, of experimental uncertainties (error bars).
- Experimentalists: strive to mostly run controlled and measured experiments.

Table 1.4.1. V&V Credibility Checklist for Non-Specialists.

CHAPTER 2

SEMANTICS: TERMINOLOGY, TAXONOMIES, DEFINITIONS

The purpose of a fish trap is to catch fish, and when the fish are caught, the trap is forgotten.
The purpose of a rabbit snare is to catch rabbits. When the rabbits are caught, the snare is forgotten.
The purpose of words is to convey ideas. When the ideas are grasped, the words are forgotten.
Where can I find a man who has forgotten words? He is the one I would like to talk to.

"Means and Ends," *The Way of Chuang Tzu*, Thomas Merton
(New Directions Publishing Corporation, New York, 1965)

2.1 INTRODUCTION[4]

Background discussion, definitions and descriptions will be given for some terms related to confidence building in computational PDEs. Examples will be given of worthwhile semantics vs. worthless semantics, and practical definitions vs. effete philosophizing. The *critical* distinction is made between Verification vs. Validation. Distinctions will be made between numerical errors vs. conceptual modeling errors; iteration convergence vs. grid convergence (or residual "accuracy" vs. discretization accuracy); adequate and inadequate error taxonomies; Confirmation, Calibration, Tuning, and Certification; Verification of numerical accuracy of codes vs. Verification of individual calculations; truncation error vs. discretization error; customer illusions vs. customer care; and Quality Assurance vs. quality work.

[4] This chapter is based on Roache (1995), "Verification of Codes and Calculations" and Roache (1997), "Quantification of Uncertainty in CFD" with significant new material in Section 2.3.

The editorial "we" or "our opinion" will be used herein to refer to an opinion held not just by myself but by a significant number of practitioners in the field, although not necessarily a majority. The more restricted "I" will be used to indicate more individual opinion, although not necessarily unique. The definitions and distinctions have developed over years, with less standardization in the earlier years. The definitions did not flow inexorably from the etymology or the common use (as some philosophers would have us believe) but rather developed as the need arose to make the distinctions. None of the terms has enjoyed a pristine use in the literature, i.e., all have been used by authors (including myself) in ways inconsistent with the present definitions, and some continue to be so. Nevertheless, the definitions and usage proposed herein are becoming standardized, enabling communications to become more efficient and precise. Also, the ideas behind the words are important.

"Verification and Validation" and the broader area of confidence building in computational PDEs is a curious subject. The growing recognition of its importance is attested to by the policy statements given by professional journals and societies, as noted in Chapter 1. See Roache (1990) reproduced herein as Appendix A for my experience in implementing the policy in Roache et al (1986). The tone of articles can be fairly legalistic, yet the area is quite subjective, dependent on opinions, world-view, philosophy of science, philosophy of engineering, and appeals based on common sense as much as mathematics. This is partly due to the semantics involved.

2.2 SEMANTICS

I fear that many readers will not be interested in this discussion, but will dismiss it as "mere semantics." But it is important, and in my experience can have major consequences to projects. "Semantics," after all, is "the study of meaning, especially in language," and would seem to be a worthwhile activity if people want to know *what it is* they are arguing about. The negative connotation of "mere semantics" arises when people argue uselessly about words without looking beyond, to the ideas behind the words. For example, the choice of "Verification" or "Validation" was originally arbitrary (in our opinion), and is now recommended solely because of common developing use. (I have published articles using the opposite definition.) In a common English thesaurus, "verify," "validate," and "confirm" are all synonyms, but the words are used herein, and generally in code Quality Assurance (QA), as *technical terms* with more context-specific meaning.

Such technical terms are preferably related to common use, although regrettable exceptions occur. (For example, I have always found the term "moment" as used in mechanics to be annoying because its technical meaning is orthogonal to common usage, whereas "torque" is adequate and commonly understood.) But the term's *technical meaning* is defined independent of common use, and in a *specific technical context*. The same word can have different technical meanings in different technical contexts. Biologists use the word "model" for a laboratory rat with a human-like disease, dentists speak of a "calculus" that Leibnitz would not claim, and "Confirmation" is a religious rite. Even the word "error," which a non-technical person would never suspect of ambiguity, has multiple meanings depending on context. Programming errors are *mistakes* but discretization errors are not; in common terminology, "error" and "mistake" are virtually synonymous.

This is not a universally accepted attitude toward semantics. In a widely quoted paper described as "brilliant" in an otherwise excellent *Scientific American* article (Horgan, 1995), Oreskes et al (1994) think that we can find the real meaning of a technical term by inquiring about its common meaning. They make much of supposed *intrinsic* meaning in "verify" and "validate" and, like a Greek morality play, agonize over "truth." They come to the remarkable conclusion that it is impossible to verify or validate a numerical model of a natural system.

Most of their concern is with groundwater flow codes, and indeed, in geophysics problems, Validation is very difficult. But the authors extended this to all physical sciences. They even stated the following: "We are left with the conclusion [from their flawed syllogism] that we can never verify a scientific hypothesis of any kind." They clearly have no intuitive concept of error tolerance, or of range of applicability, or of common sense. My impression is that they, like most lay readers, actually think Newton's Law of gravity was proven wrong by Einstein, rather than that Einstein defined the limits of applicability of Newton. But Oreskes et al (1994) go much further, quoting with approval (in their endnote 36) various modern philosophers (especially Popper, 1980) who question not only whether we can prove any hypothesis true, but also "whether we can in fact prove a hypothesis false." They are talking about physical laws (not just codes, but any physical law). Specifically, we can neither validate nor invalidate Newton's Law of Gravity. One might expect that even a philosopher who would not accept "proof" of Newton's inverse-square law could at least accept that one could *disprove* a "*direct*-square law" hypothesis, i.e. that gravitational attraction *increases* with the square of distance. But not so; to these philosophers, neither Newton nor anti-Newton can be "proven." See also Konikow and Bredehoeft (1992) for a similar outrageous claim. (What shall we do? No hazardous waste disposals, no bridges, no airplanes, no construction codes for earthquake areas, no analysis or predictions of atmospheric pollution, no...)

If one insisted on reserving an overly rarefied concept of "proof" only for mathematics, one could still reasonably use the word "validate" with a common-sense (or more precise) concept of specified accuracy uncertainty. One could say with Hull (1997) (in the context of creationism arguments against the "unproved theory" of evolution) that "Newton's celestial mechanics...is a well-validated theory (in the sense of explaining or predicting the result of an observation) that is unproved and incomplete. We even know its successor–general relativity. Nevertheless, we launch Mars probes, among other space voyagers, confidently using Newton's mechanics. Unproved (but validated) theories are quite useful! The electric power industry is based on Maxwell's theory of electricity and magnetism, unproved and incomplete as it is."

As Rykiel (1996) noted of this Popperian philosophical approach, "In effect, Validation is equated with certainty rather than a degree of belief." It is important to recognize that these controversies are not just the stuff of academic debates, but have serious consequences, especially on public policy. For example, it can (and has) happened that ill-conceived legalistic (regulatory) definitions of Validation could, if applied rigidly and without the leaven of common sense, categorically eliminate a geologic repository as a possibility for hazardous or radioactive waste disposal. It is not necessary to argue here the relative merits of different system categories, but we can at least allow the possibility that such a geologic repository *could*, at least in principal, be the best (safest and cheapest) method of disposal. The problem here with regard to overly legalistic definitions of Validation is *time* and *spatial data*. The time scale legislated for low-level radioactive waste containment is 10,000 years and clearly no tests can be run to "Validate" a model at this time scale. Likewise, the spatial data regime for one site (WIPP, now operational) covers hundreds of square kilometers to a depth of at least 700 meters, and pumping tests at length scales of 10's of meters to completely cover the area are clearly not possible (WIPP PA, 1992; Helton et al, 1995,1996). Unless common sense and good engineering practice prevail, the legalistic definition of "Validation" could *categorically* eliminate this potentially best category of waste disposal. For the benefit of those who would prefer to eliminate it, with a feeling of moral superiority, I note two additional facts. (1) The pseudo-intellectual agonizing over "truth" and impossibly rarefied definitions of "Validation" will ultimately eliminate *all* conceivable disposal schemes, not just geologic systems. (2) The dangerous waste already exists, it will not go away, and in fact is presently stored in systems which are fragile and dangerous even by common-sense standards. Also, this sophomoric attitude towards semantics likewise would eliminate bridges (built, after all, on ground, with unknown data) and aircraft; the atmosphere provides the "open system" which Oreskes et al (1994) claimed eliminates any possibility of "Validation."

A legitimate and well-meaning concern of Popper and the related philosophical community including Oreskes et al is to avoid misleading the public with overwhelming claims of specialized knowledge. Oreskes et al think "verification" etc. are too strong, implying *certainty* or *truth* to the public. But their suggested replacement (from Popper) of "corroboration" is no better in this regard. They call for a neutral language to avoid such implications, but as Rykiel (1996) observed, "It is hard to see how this translation to a neutral language could be accomplished. ... the better choice may be to educate the audience."

For a mild but worthwhile criticism of Oreskes et al (1994), see Rykiel (1994). However, the same author (Rykiel, 1996) has given a penetrating discussion of these issues and others related to Validation. The focus is on ecological modeling, but is highly recommended reading for those in any discipline. (See also Section 9.2.3 for excerpts from Rykiel.) For my complete criticism of Oreskes et al (1994), see Appendix C of V&V1. For criticisms of Konikow and Bredehoeft (1992), see de Marsily et al (1992) and Leijnse and Hassanizadeh (1994). This latter paper is important for unraveling the semantics associated with the word "model," in particular, whether "model" includes parameter values. They make the worthwhile distinction of "strong" and "weak" definitions of "model" (based on whether or not it includes parameters or is only a conceptual model), and corresponding narrow and broad senses of Validation. See also related discussion of modeling philosophy in Tsang (1991) and in the entire Special Issue of *Advances in Water Resources* (Gray, 1993) devoted to "Research Perspectives in Hydrology." Further criticism of the science philosophy of Popper will be presented in Sections 9.2.2 and 9.2.3.

The Konikow and Bredehoeft (1992) paper, entitled "Groundwater Models Cannot be Validated," asserted that "In the end, action concerning waste disposal will be a judgment; a professional judgment by the scientific community and a judgment by society." We assert that questions like "Can groundwater models be Validated?" and "Do we or don't we model?" are phony questions. The fundamental and meaningful question is, "Do we make decisions qualitatively or quantitatively?" All the geological information and insight possible (age of rocks, chemical composition, color, ...) cannot provide the quantitative information on groundwater flows that are necessary for decisions on important subjects like hazardous and/or radioactive waste disposal. Once we decide to make decisions quantitatively (and we must), then we *do* model. The next question is one of scale of the model; a "back of the envelope" calculation, based on geological insight etc., *is* a model. It is a model using the limit of crude resolution, a single cell or control volume, or what chemical engineers refer to as a "batch model." Whether such a batch model is adequate, or if high-resolution computer solutions of partial differential equations are required, depends on the accuracy required. In any case, the models must be Validated in some sense, at some level of accuracy, no matter how crude, in order to be quantitatively useful.

Clearly, we are interested in normal speech and practical definitions, applied in the context of engineering and science accuracy, not in such worthless semantics and effete philosophizing.

Below, descriptions and definitions are given for some terms related to confidence building in computational PDEs. (I will not try to define "confidence building" with any precision.) To reiterate, these are technical terms described or defined in a technical context, not just common language.

2.3 Δ VERIFICATION AND VALIDATION: NUMERICAL VS. CONCEPTUAL MODELING

First and foremost, we must make the essential distinction between Verification and Validation. Following Boehm (1981) and Blottner (1990), we adopt the succinct *description* (not definition) of "Verification" as "solving the equations right," and of "Validation" as "solving the right equations."

"*Verification*" ~ *solving the equations right*.
"*Validation*" ~ *solving the right equations*.

The code author defines precisely what continuum partial differential equations and continuum boundary conditions are being solved, and convincingly demonstrates that they are solved correctly, i.e. usually with some order of accuracy, and always consistently, so that as some measure of discretization (e.g., the mesh increments) $\Delta \rightarrow 0$, the code produces *a* solution to the continuum equations; this is Verification. Whether or not those equations and that solution bear any relation to a physical problem of interest to the code user is the subject of Validation.[5]

In a meaningful though perhaps overly scrupulous sense, a "code" cannot be Validated, but only a calculation (or range of calculations with a code, for a specific class of problems) can be validated. In my experience, dealing with other than algorithm developers, this is a difficult concept and requires continual reiteration.

Another way to make the distinction (i.e., to get to the idea behind the words, beyond "mere" semantics) is to speak of numerical errors vs. conceptual modeling errors. An example of conceptual modeling vs. numerical modeling is the assumption of incompressibility. This is clearly a conceptual modeling assumption. Is it the code builder's fault, or any criticism of a commercial code itself, if the user incorrectly applies it? For example, dynamic stall of helicopter rotor blades involves compressibility at a surprisingly low free-stream Mach number. Results from an incompressible code may not agree with experiment very well, but we cannot say that the code fails Validation because it was applied to compressible flow, although we may have some sympathy for the user who is fooled by dynamic stall. But no one would have sympathy for a user who applied an incompressible flow code to a reentry vehicle at Mach 20. In this example, and in many practical cases, the lack of agreement with experiment is not a *code* problem, but a conceptual *modeling* problem.

"Model" includes more than the code. "Model" includes conceptual modeling assumptions (e.g., incompressibility, symmetry, etc.). "Model" in this specific or strong sense also includes data input to the code, e.g., geometry data (which are not so easy to determine accurately as many people assume) and boundary conditions and initial conditions). These can lead to failure of Validation of a *model*, with possibly no criticism of the *code*.

Another way to make the distinction between Verification and Validation is to follow the classical distinction between mathematics and science. Mathematics is a tool of science, often the predominant language of science. But mathematics exists by itself. It would be "true" regardless of any correspondence to the natural world. Verification is seen to be essentially and strictly an activity in mathematics, the mathematics of numerical analysis. Validation is essentially and strictly an activity in science and engineering science: physics, chemistry, fluid dynamics, even the "soft" sciences of economics, sociology, etc.

"*Verification*" ~ *mathematics*
"*Validation*" ~ *science/engineering*

The typical Computer Science view of "code Verification" is not that of engineering science. "Verification" in the present view does not include all aspects of code QA. For example, it does not include the important and nagging concerns of version control, or archiving of input data, or documentation (external and internal). Less obviously, it does not include *reading* of source code. Blottner (1990) described his verification of a Navier-Stokes code and included the fact that the Fortran source code was examined. Although perhaps useful, and contributes to confidence building, we do not consider it to be part of Code Verification *per se* (nor of "Confirmation"; see below). Even as part of Code QA, the practice is of less value than it might appear at first glance. Consider the ridiculousness of regulatory agency personnel

[5] Possibly the first such distinction in modern terminology was made by Fishman and Kiviat (1968), who also used descriptions rather than definitions.

reading source code when they cannot comprehend the algorithms. Or consider how difficult it is, even if one comprehends the algorithm, to read someone else's source code for a three-dimensional semicoarsening multigrid method, or a full spectral element code, or a block-7 implicit multi-equation turbulence model. We can read source code, either our own or someone else's, interminably and still not be able to claim Verification. On the other hand, if the grid convergence tests prove that the code is (say) 2nd-order accurate, it would take some strange coding to nullify this test. (Perhaps some in-line limit on the parameter range that was not "hit" in the grid convergence test is possible.) Except for these unusual cases, reading of the code appears to be neither necessary nor sufficient for Verification.

Surely few would claim that reading of code is any substitute for Verification via grid convergence testing (although some of my clients have tried). If we included reading of the source code as part of Code Verification, then the code author's reading could be claimed as a "partial Verification." We know it does not prove a thing. In our view, "Verification" (in this restricted, technical meaning) should treat the code as a "black box," without reference to its internal organization. Note, then, that the present "Verification" has nothing in common with the type of software systems that perform internal consistency checks, look for "dead code" (i.e., un-exercised code segments, which might be left-over artifacts from early stages of code development), and otherwise perform computer-science-type of code "Verification" by consideration of the internal code.

As reading source code does not constitute partial Verification, neither does extensive use of a code, nor publication, as some authors have claimed. Especially in groundwater flow codes, where the physical parameters and initial conditions are so poorly known, extensive experience with so-called "real world" calculations (even hundreds of cases calculated by many different users) would be an inefficient and, more importantly, *inconclusive* way to approach Verification. Likewise, journal publication and wide code distribution do not constitute Verification. Code Verification is not some kind of exercise in democracy, anymore than proving a theorem is. *First* one verifies a code's accuracy, *then* one publishes. (Note that, when codes with many options combinations are considered, no journal would publish a complete set of Verifications, nor would most journal referees consider the test in sufficient detail.) Extensive code use and journal publication certainly have value. They add to confidence building, and are worthwhile for general code QA, and are helpful to improve code documentation, garner suggestions for input/output improvements, obtain data on performance measures for a wide range of problem parameters, demonstrate robustness and portability, etc. They are just not part of Verification as technically defined herein.

On the issue of robustness, note we do not consider a code bomb (i.e., a divide by zero, or instability) to be a failure of Code Verification in the present context. As long as the code does not lie to the user, Code Verification has not been breached. Code robustness is certainly desirable, and is properly part of QA and overall code evaluation. From a QA "customer" point of view, robustness is often paramount, unfortunately sometimes to the point that it trumps all other considerations. (Frankly, users often do not care about accuracy, so long as the computational results are smooth, and are obtained with no intelligent interaction required.) But in our use of the terms, robustness is not an issue in Verification. (Besides, it is well known that robustness is often achieved at the expense of numerical accuracy, as noted in Chapter 1.)

2.3.1 § Definitions and Interpretations of Validation

We now go beyond mere *description* of Validation and begin consideration of some attempts at actual *definition* of Validation. This has proven to be a surprisingly contentious issue in the literature, as will be seen.

Since Validation involves comparisons with experiment, there is necessarily an error tolerance involved somewhere in the process, and what level is acceptable depends on the use intended. Whether or not the

acceptable level is to be included in Validation, or relegated elsewhere, is a major issue. Such considerations were incorporated into Mehta's (1995) more expanded definition.

"*Validation* is defined as the process of assessing the credibility of the simulation model, within its domain of applicability, by determining whether the right simulation model is developed and by estimating the degree to which this model is an accurate representation of reality from the perspective of its intended uses." (Mehta, 1995)

This definition used by Mehta in the aerospace industry is in fair agreement with that accepted by Tsang (1991) in the groundwater modeling community, as given originally by Schlesinger (1979).

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

Tsang (1991) also explicitly included the specification of performance metrics in the definition of (groundwater flow) model Validation, and noted that "almost by definition, one can never have a Validated computer model without further qualifying phrases." Tsang also noted the different aspects of Validation with respect to (a) a process (i.e., governing equations) and (b) a "site-specific system," which includes geometric structure, stratigraphy, fractures and other geologic characteristics, plus boundary and initial conditions (both of which are notoriously vague in groundwater modeling). Tsang also took the ostensibly precise approach of only speaking of Validation of computer *models* rather than *codes*. Tsang stated that the term "code Validation" is illogical, but close reading shows that his problem arises from not distinguishing Validation from Verification (a term he did not use at all). Likewise, some of his "other Validation methods" (other than the basic approach of comparing model predictions with laboratory or field data) would be classified as Verification or model sensitivity studies in present terminology, and the rest are simply variants of the basic approach. (Given this relatively minor caveat on semantics, Tsang's paper is highly recommended reading for modelers of groundwater flow and transport.)

The most commonly used definition of Validation is the following.

Validation: The process of determining the degree to which a model {and its associated data} is an accurate representation of the real world from the perspective of the intended uses of the model.

Unfortunately, considerable disagreement exists on what this definition *means*, or should mean. This definition of validation has been cited extensively in computational modeling fields, and is widely accepted, e.g. DoD (2003), AIAA Guide, V&V10, NASA (2008). (The last adds "or a simulation" in place of the bracketed words.) Despite the apparent clarity of this concise one-sentence definition using common terms, there are at least three contested issues: whether *degree* implies acceptability criteria (pass/fail); whether *real world* implies experimental data; and whether *intended use* is specific or general (even by those who think it is needed at all). This gives $2^3 = 8$ possible interpretations of the same definition, without even getting into arguments about what is meant by *model*, i.e. computational, conceptual, mathematical, strong, weak. The job of sorting out claims and arguments is further complicated by the fact that principals in the debates have sometimes switched sides on one or more of these three issues (myself included).

An extended discussion of the pros and cons of these issues, and the history of the controversy, was given in Roache (2009), reproduced herein as Appendix B. Those summary recommendations, consistent with V&V20, are as follows.

Issue #1. Criteria for *acceptability* of accuracy (adequacy, or pass/fail criteria, or accuracy tolerance) are not part of Validation[6], but analysts performing validation exercises should be wary of appearing to

[6] Oberkampf and Trucano (2007, 2008), amoung others, concur with this recommendation, and in fact W. Oberkampf was persuasive in changing my opinion from that expressed in V&V1. In some disciplines, this

bless a code as "validated" when it is clearly unsatisfactory for any reasonable application (e.g. it cannot even predict correct qualitative trends). In an engineering project, the acceptability of the agreement is part of the next project step, variously called accreditation, certification, or other. It is an engineering management decision, not a scientific evaluation.

Issue #2. Experimental data is necessary for Validation. Many have said unequivocally [5-8,11] that experimental data are the *sine qua non* of validation.

No experimental data => No validation

Issue #3. Intended use, at least in its specific sense, is not required for validation. For example, the well-known data on turbulent backstep flow of Driver and Seegmiller (1985) in the ERCOFTAC database can be used for code/model validation, with neither the experimenters in 1985 nor modelers in (say) 2008 having a specific use in mind. This is precisely the situation for the Third Lisbon V&V Workshop (Eça and Hoekstra, 2008).

However, it is also true and very important that experiments designed specifically for a validation exercise, and with a specific application in mind, and with collaboration between experimenters and modelers in the design of the experiments, are much more likely to produce data on the relevant metrics with relevant precisions than are experiments designed without applications in mind.

In general terms, validation involves comparison of modeling results with experimental results. This has been used as a working definition in the past, but we agree with Oberkampf et al. (2004) that it is too soft, the trouble being that the difference between model result and experiment is too easily taken to be the accuracy when in fact the story is more difficult. It is time to improve standards somewhat on even the minimal requirements for the term *validation*. The minimal required improvement is contained in one word: *uncertainty*. We can describe validation (legitimate, minimal validation) as the comparison of model results *and their associated uncertainties* with experimental results *and their associated uncertainties*.

2.3.2 § Definitions and Interpretations of Error and Uncertainty

2.3.2.1 § Numerical Error

The error of a numerical value (experimental or computational) is intuitively defined as the {numerical value - true value}. Some care is required, as always, in making some distinctions.

The "true value" of an experimental result f (say the lift coefficient C_L for a wing) could be interpreted to mean the true value in flight, or the true value in the wind tunnel. Note that the wind tunnel measurements will have experimental errors from sources other than those inherent in using a wind tunnel to model atmospheric flight, e.g. instrument errors, physical model alignment errors, etc. For code/model Validations, it is best to take the experimental "true value" of f to mean that of the experiment "as run" (V&V20). The applicability of either that experiment or the simulation of that experiment to the free flight case is a separate issue from computational code/model Validation.

---

concept is isolated by the term "simulation fidelity" (Pace, 2003) so the present interpretation of "Validation" could be replaced by "fidelity assessment." It is a good descriptive term that carries no connotation of either acceptability or unacceptability and therefore little chance of misinterpretation. It would be preferable, but Validation is a term that will continue to be used so we must deal with its ambiguities.

There is more than one "true value"! The "true value" of the computation is *not* the true physical value, but the true *mathematical* value, i.e. the mathematical solution that would be produced by the code/model using an idealized computer with infinite resolution and word length: in other words, the continuum solution of the governing PDEs (plus boundary and initial conditions, of course). The "numerical accuracy" relevant here is just the accuracy of the (finite) code result relative to the continuum solution of the same governing equations. For example, consider three codes for the freshman physics problem of calculating point mass trajectories in vacuum. One code is based on a flat earth model with constant *g*, the second uses a spherical earth with an inverse square *g*, and the third uses higher order harmonics and local mass concentration for mountain ranges (and relativity corrections, if you like.) Each code will have its own "true value" for the trajectory, for purposes of evaluating the numerical errors. The difference between the "true values" of each code and the true *physical* value is simply the *modeling error* of each code. However, it is very common and understandable to mix these terms and to refer to the "numerical error" of a code prediction when "modeling error" (really a combination of numerical error and modeling error) would be appropriate. Often the fuzziness causes no difficulty in context, but sometimes it is necessary to insist on maintaining the distinction.

2.3.2.2 § Uncertainty

The concept of uncertainty is easily and frequently muddled with the concept of error. (I speak from personal experience here.) Uncertainty and error certainly are related, and confusion is abetted by the traditional use of the term "error bars" or "error bands" which are really not errors but uncertainties, and by the fact that there are two generic types of uncertainties, which are closely related but not identical. Uncertainty has sometimes been described as an estimate of error (AIAA Guide, page 10) which is not quite true (AIAA Guide, page 5) but is suggestive.[7] We will get into more details and variations in Chapter 11. For now, we consider the most common type of error bar or "expanded uncertainty" (V&V20) denoted by $U_{xx\%}$, often $U_{95\%}$ in engineering and physical sciences. We want to put a ± error bar around our calculated value that we expect to contain the true (mathematical) value in about 95% of cases, or with about ~20:1 odds.[8]

Consider a simulation for a wing that produces $f_{sim} = C_{L(sim)} = 1.2$. Using techniques to be described in Chapter 5, we might estimate the (signed) error to be $\delta = -0.1$. This means we are estimating that the simulation value 1.2 is in error by -0.1, i.e. the simulation value is lower than the true (mathematical) value by 0.1. So we think the true (mathematical) value is better estimated by $f_{corrected} = f_{sim} - \delta = 1.2 + 0.1 = 1.3$. How does this relate to an uncertainty like $U_{95\%}$? We want to calculate an interval such that $\langle f_{sim}-U_{95\%} , f_{sim}+U_{95\%} \rangle$ will contain the true (mathematical) value of f with ~20:1 odds. First, note that the error estimate δ is signed, whereas the uncertainty is applied as ±. [9] Second, if the error estimate is applied

[7] The AIAA Guide and V&V10 also define *Error* as "A recognizable deficiency in any phase or activity of modeling and simulation that is not due to lack of knowlege" and *Uncertainty* as "A potential deficiency in any phase or activity of the modeling process that is due to lack of knowlege." It is not evident how one could make use of these definitions in calculating any estimate of error or uncertainty.

[8] Intuitively, and in normal conversation, one could also speak of ~95% confidence, or ~95% probability. But beware; these are loaded technical terms to statisticians, and the associated debates strike at the heart of science philosophy. Likewise, the word "cases" needs some consideration. See Section 5.14.

[9] The usual unsigned uncertainty wastes information if one has confidence in the sign of δ, and a signed uncertainty can be developed (Romero, 2008) but it is difficult to combine with experimental uncertainties (see V&V20 and Chapter 11) which are traditionally given as ±. It is worthwhile to recognize that this waste does not affect the % coverage. If the true error has the same sign as δ and if a signed uncertainty

as a ± interval, as $<f_{sim}-\delta, f_{sim}+\delta>$, we would not expect this interval to contain the true value with 20:1 odds. In fact, we would usually expect (for a "best estimate" and certainly for any ordered estimate) that the odds would be ~1:1, i.e. equal chances for the true value to be inside as outside this interval. Thus $|\delta| \sim U_{50\%}$. (Roache, 2003a). Clearly, the intended $U_{95\%}$ is > $|\delta|$. The factor will be determined empirically in Chapter 5, which also contains summary distinctions between error estimators and uncertainty estimators.

2.3.2.3 § Coding Errors

Coding errors do not fit this description of "error" and in fact are a different sort of topic altogether than an ordered numerical error (discretization error or truncation error). An ordered numerical error is a *beautiful* thing; it is the foundational basis of all computational physics, and can be made arbitrarily small with application of sufficient computer power (grid resolution and word length). We can estimate the numerical error and, with empirical experience, state quantified error bars or uncertainties for it. On the other hand, a coding error is a screw-up. Or as Sinclair et al (1997) denoted it, e_b, for error due to blunder. The identification, elimination, and engineering "proof" or demonstration of the absence of coding errors are the concern of Code Verification.

2.3.3 Definitions and Interpretations of Code Verification

It is not so difficult to find agreement on Code Verification, which is defined rather tersely by the IEEE (Jay, 1984): "Formal proof of program correctness." We agree with Oberkampf (1994) in his evaluation. "Although very brief, this definition brings unprecedented clarity to the meaning of the term, and it adds a new perspective to the issue. Specifically, this definition bluntly requires correctness or veracity of the prediction, without bringing in supporting topics such as what is being predicted or how it is done." Although more general than the definition used herein, this IEEE definition is actually compatible with our descriptions, is more general, and is still compatible with the distinction between Verification and Validation. That is, "program correctness" for a PDE code would naturally include "solving the equations right," and of course a definition of what those (continuum and discretized) equations are, without getting into the question of whether certain problems are appropriate for those equations and that code, i.e. Validation.

The definition of Verification given by Golub and Ortega (1992), quoted approvingly by Jameson and Martinelli (1996), is somewhat muddled, since it uses the word "verification" (apparently in a common, non-technical sense) within the definition of Validation (in a technical sense). Likewise, Jameson and Martinelli used "validation" as we do here for the physical model correctness, but only for the" overall approach" or the "final validation." (I find it difficult to distinguish the "overall approach" from the submodels.) They also stated "verification is needed for every single claim made in constructing the model, or submodels (both physical and numerical)," i.e. they used "verification" in regard to the physical model as well as the numerical process. Likewise, they also used "Validation" for the "numerical solution process" requiring that it be "separately verified and validated." Other attempted distinctions made are also unclear, e.g. whether they intended "numerical scheme" to include the grid resolution. Nevertheless, the paper contains excellent practice and results, and is recommended reading. (See also excerpts in the present Chapter 8.)

$U_{S95\%}$ contains the true value within the interval $<f_{sim}+U_{S95\%}>$ in 95% of the cases, then an unsigned uncertainty $U_{95\%} = |U_{S95\%}|$ also contains the true value within the interval $<f_{sim}-U_{95\%}, f_{sim}+U_{95\%}>$ in 95% of the cases.

2.3.3.1 Δ *Solution Uniqueness*

The question of "solution uniqueness" always arises with nonlinear equations, and its position in the "Verification" or "Validation" distinction is important. If, as stated carefully above, we take "Verification" to mean simply that "*a* solution" to the continuum PDEs (plus boundary and initial conditions) is obtained, then the problem of PDE non-uniqueness is avoided. This is probably being too easy on the code developer in some cases. Physically inadmissible solutions should be eliminated by the code, e.g., a shock-tube simulator should be required to eliminate expansion shock solutions. But nature abounds with physically non-unique solutions: in fluid dynamics, e.g., hysteresis of airfoil stall and recovery, and bi-stable fluid amplifiers. If nature cannot decide which solution to produce, we cannot expect more of a code. Note that obtaining "*a* solution to the continuum PDEs" may involve sorting through multiple numerical solutions for nonlinear problems; e.g., see discussion in Chapter 8. See also especially Stephens and Shubin (1981) whose study of Euler solutions indicates that the multiple numerical solutions converge to the same solution as the grid is refined. Note that numerical artifacts may select one multiple solution over another; for example, Oberkampf et al (1995) cited the example of Levy et al (1995) showing that the manner in which the left-hand (iteration driver) side of each difference equation is formulated impacts whether the symmetric or asymmetric solution of steady state vortex flows over slender bodies at angle of attack is obtained. Oberkampf et al (1995) labeled the asymmetric solutions for this class of flows as "spurious," but in general we cannot be categorically dismissive. For example, computations of both laminar and turbulent flows through symmetric expansion channels can produce either symmetric or asymmetric solutions (provided that the entire channel flow is calculated, rather than having symmetry enforced through the *fiat* of computing only half the flow). Contrary to wide-spread intuition, nature prefers the asymmetric solution; this is the basis of bi-stable fluid amplifiers (e.g., see IAHR, 1982). Likewise, physical flow over a geometrically symmetric cavity (rectangular or spherical) does not produce a symmetric flow, but one in which the axis of rotation of the primary recirculating vortex is skewed. In general, it is asking too much of simulations to always select preferred solutions (either preferred by nature, or by intuition of the modeler). Sometimes, the "spurious" (or at least, unanticipated) solutions are demonstrably artifacts of discretization, as when the asymmetry tends to disappear as the grid is refined (Sengupta et al, 1995). Other times, the unanticipated solution persists as the grid is refined, and is therefore "Verified."

Non-uniqueness can also be attributable to mathematical modeling of constitutive equations. The original RANS k-ε turbulence model dates back to the 1970's, but it was not until 2006 (Rumsey et al, 2006; Pettersson et al, 2006) that fundamental problems with uniqueness were revealed for some low-Re versions. The solutions can depend on extraneous factors such as initial conditions and iterative algorithms and paths. (Better RANS models have not exhibited problems of non-uniqueness.) It is unlikely that this non-uniqueness would have been discovered routinely during Code Verification or Calculation Verification.[10] Spurious non-uniqueness is probably best treated as a Validation issue rather than as a Verification issue.

2.3.3.2 Δ *Solution Instability and Over-Stability*

A closely related question is the *stability* of a solution. Although people usually think of artificial (i.e., numerical artifact) *instability* as a code difficulty, a common situation is artificial *stability*, e.g., a 1st-order

[10] At most, non-uniqueness could result in a false-negative conclusion (the false conclusion that the code has an error when in fact the coding is correct) because the code converged to a different solution than the one built into the MMS (see Chapter 3). See also discussion in Roache (2006).

time-dependent code whose artificial viscosity damps out disturbances, so that accurate calculations of physical fluid instability would require impractically high resolution. It is asking too much of a code to exactly mimic stability boundaries except in the limit of $\Delta \rightarrow 0$. For example, it is not a failure of Verification for a full Navier-Stokes code to produce steady shear layer solutions at Reynolds numbers known to be unstable; an unstable solution is still a genuine solution. (It is certainly not necessary, as claimed in a SIAM article by Johnson et al [1995] to analytically and *a priori* solve the hydrodynamics stability problem in order to estimate numerical errors!)

Stable or not, observable in nature or not, and unique or not, such Verified simulations are mathematical solutions. Sorting out the other aspects is not an aspect of Verification; it is an aspect of Validation, but it should be approached with great respect for the subtlety of nature.

2.3.3.3 § No Physical Experiments in Verifications

Also, we believe strongly that Code Verification can and *should* be completed without appeal to physical experiments. The first part of this statement claims that Code Verification should be *completed*, i.e., Code Verification is *not* an ongoing exercise. Verification, as we have said, is an exercise in mathematics, not science. When one proves a theorem, the work is completed. Proving the formula for solution of a quadratic equation is not ongoing work. This is not to say that one could not have made an error in the proof of a theorem, nor that Confirmation exercises (see below) are not valuable in confidence building. It is to say that Code Verification is a mathematical activity that in principle comes to a conclusion, e.g. a code is or is not 2nd-order accurate for benign problems. The second part of the statement claims that Verification can and should be achieved without using physical experiments. That it *can* be achieved is unarguable - very general methods are available (see the Method of Manufactured Solutions in Chapter 3). That it *should* be achieved is perhaps debatable; Oberkampf (1994) and Aeschliman et al (1995) allowed Verification for carefully performed experiments in parameter regimes wherein confidence is high between the correspondence of the governing equations and the physical science. I suppose the physical experiment could be regarded as a sort of continuum analog computer that produces "solutions" to the governing equations, but at best such a definition would be strained and unnecessary. Experimental agreement (at least, comparison) is part of Validation, not Verification, and the concepts are distinct enough to be worth preserving. Also, as Roberts (1988) noted, justification of a purely mathematical approximation does not need physical experiments, e.g. if you want to know if the irrotational approximation is good, then compare potential code results with Euler results. Not only are physical experiments not needed, they may just be confusing, because one now has to sort out two sources of error, the rational approximation error and the physical experimental error. The other extreme is to consider high-order DNS (Direct Numerical Simulations) of turbulence to be of such high quality that simpler turbulence models can be "Validated" by comparison with this mathematical realization. Again, I would prefer to restrict the terms Verification and Validation. (Alternately, one can use the somewhat less precise terms of "Confirmation" and/or "Benchmark". See below.)

Validation has highest priority to engineers and scientists because "nature" is the final jury. But any experience with laboratory experiments will quickly disavow the absolute quality of experimental data. (It is asking too much of a CFD code to agree with wind tunnel test data when these data do not agree with other wind tunnel data, nor even from one run to another in the same wind tunnel.) Thus, we strongly believe that complete Verification of a code (and a calculation - see below) should *precede* any comparisons with experimental data, i.e., Verification first, then Validation. This is not so much a matter of definition as it is recommended engineering practice, but its wisdom is attested by the voice of experience (often using differing semantics), e.g., Bradley (1988), Marvin (1995), Aeschliman et al (1995), Porter (1996), Melnik et al (1995, p. 23), Sindir et al (1996), Jameson (1992), Jameson and Martinelli (1996). To

quote Jameson (1992): "Simply comparing experimental data with numerical results provides no way to distinguish the source of the discrepancies, whether they are due to faulty numerical approximation or programming, or to deviations between the math model and the true physics." The aerospace consortium experience in Verification, Validation and Certification (see definitions below) reported in Melnik et al (1995) was forcefully clear, as follows.

"In the past constraints on computer resources have led to shortcuts using inadequate coarse grids leading to inadequate Verification. In this project, because we were able to use grids that were sufficiently fine, we were able to observe first-hand the pitfalls of working with inadequately Verified codes [calculations]. From our experience in the project it is clear that proper Verification [of calculations] is an absolutely necessary first step in the code Certification process. This basically involves the determination of the grid required to achieve a specified level of numerical accuracy." (Melnik et al, 1995)

Note another contrast between Verification and Validation: Verification is *completed* (at least in principle, first for the code, then for a particular calculation) whereas Validation is *ongoing* (as experiments are improved and/or parameter ranges are extended).

This brings up another aspect of experiments. One sometimes hears complaints that a "CFD code needs additional data" that experimenters typically do not measure. For example, data are published on turbulent flow in channel expansions which do not characterize the details of the boundary layer ahead of the expansion (nor sometimes elementary quantities like boundary layer thickness). This complaint neglects a fundamental fact of life:

CFD codes require no more information than the physics.

In fact, a CFD code may be expected to require *less* information than the full physics, due to simplifying approximations in the turbulence theory. If experimenters have not measured these quantities, then they have run an uncontrolled and/or unmeasured experiment, regardless of whether CFD codes will be used. The question, of course, is whether these unmeasured quantities are important to the physics of interest. Marvin (1995) used the apt term "well-posed validation experiment." This seems to be a major opportunity for CFD to contribute to experimental work; CFD can be used to predict the importance of difficult-to-measure quantities like wind tunnel flow angularity, non-uniform stagnation enthalpy, surface waviness, etc. as discussed lucidly by Aeschliman et al (1995). A premier example given in the study by Haynes et al (1996) is the sensitivity of boundary layer transition to free stream vorticity, which is "an unusually difficult experiment" but which can be modeled by CFD. (See further discussion in Chapter 10.)

When we say that Code Verification is completed, rather than ongoing, we are addressing only a completed code, or code version, and perhaps only a limited set of option combinations. Large scientific and commercial codes develop over years, and have multitudinous combinations of options. In this sense, we can say that code Verification is an ongoing process, but only because the word "code" as used is amorphous, really referring to many codes (all with the same name, but hopefully different version numbers), each of which in turn must be Verified.

For guidelines on Verification and Validation in the nuclear industry, see American Nuclear Society (1987).

2.4 CODE CONFIRMATION

Some Computer Science or QA people would have Code Verification necessarily performed by someone other than the code developer, sort of an "arm's length transaction" philosophy. In our view, it is ridiculous to not demand that code builders to Verify their codes! Verification is a necessary part of code development. Code authors would be remiss in their duty if they released a code without Verification. We

would trust a code builder's Verification results if presented in full; fraud is not usually the issue. But if it is, and if further tests are required (or repeats of the original Verifications to check for fraud or oversight) then we would suggest the use of the term "Confirmation" for calculations independently run by other than the code builder.

Also, the suite of problems can be used for re-Verification (e.g. after porting to a new computer or a new compiler, or to be run after addition of new options) and for user training. Code use independent of the code builder is probably more of a genuine issue in Validation than in Verification, especially when judicious "tweaking" or Tuning of code/model parameters can be involved; e.g., see recommendations in Porter (1996), and the distinction between "prediction" and "postdiction" made in the 1980–81 Stanford Turbulence "Olympics" (Kline et al, 1981).

As in Roache et al (1990), we recognize five distinct regimes where errors can be made in Verifying (or "Benchmarking") a computational PDE code, even without considering the Validation question of whether the right equations are being solved for the target problem:

1. in code generation (either by hand or using computer Symbolic Manipulation; see below);
2. in code instructions (e.g., in a user manual or comment cards);
3. in problem set-up;
4. in defining and coding a test case (analytical solutions are often more difficult to code than numerical solutions); and
5. in the interpretation of code results.

The first two are errors of the code author. The last three are errors of the code user, although ambiguous or scant code documentation can put some of the responsibility back onto the code author. "Verification" of a code removes (1) and, if done thoroughly, (2), but (3–5) still contain the potential for errors in any new application. We reluctantly conclude that there will be a continuing need for users to construct and exercise test cases *even when using Verified codes*.

Obviously, it is good common sense to build more confidence with more problems, even if the code is Verified in the sense of a theorem. In the same way, a high school student will "confirm" the general solution of the quadratic equation for a particular case by back-substitution of the solutions. This exercise does not mean that the student does not "believe" the formula, but it is prudent and pragmatic to do so, and it builds confidence, not only in the theorem but in one's understanding of the theorem. Thus, although we consider that Code Verification should be completed and therefore *is not* on ongoing activity, Code Confirmation *is* naturally an ongoing activity (as is Validation), even for well-exercised commercial codes.

Also, as a practical political consideration, not everyone can appreciate the generality of a mathematical proof or a thoroughly executed Code Verification. This is particularly true when a physically unrealistic solution is used to Verify a code (see Method of Manufactured Solutions in Chapter 3). Although mathematically sound, the exercise will not inspire confidence in the mathematically unsophisticated. For such an audience, an extensive suite of realistic-*looking* Confirmation problems, even if they alone fail to constitute a rigorous Code Verification, may build more user confidence than a single rigorous but unreal-appearing Code Verification problem.

Indeed, this skeptical attitude is not only forgivable but often wise, prudent, and pragmatic. I have the same attitude in regard to "proofs" of grid generation algorithms, single-grid error estimators, iteration convergence rate proofs, efficiency comparisons based on operation counts, gains from parallelization, and other topics. [For example, in Roache and Steinberg (1984), I gave a counter-example to a widely-accepted and important theorem on grid generation; see also Knupp and Luczak (1995).] Still, the skeptic should keep in mind the possibility, in principle and often in practice, of rigorous Code Verification in the sense of a mathematical theorem.

2.5 BENCHMARKS AND INTER-CODE COMPARISONS

"Benchmarking" of codes is a less specific term than Verification or Validation. Williams and Baker (1996) preferred to reserve the term "benchmarking" for code-to-code comparisons; this has the advantage of clearly distinguishing it from Verification and Validation, which they defined as we do herein, i.e. comparisons with closed form mathematical solutions and with experiments, respectively. Unfortunately, the term is commonly used more inclusively. It commonly refers to comparisons of results of the code in question to some other standard or "benchmark," which might be an analytical solution, a physical experiment [usually on simple configurations involving a single dominant physical phenomenon (Melnik et al, 1995; Rizzi and Vos, 1996)], a rigorously performed numerical solution, or just another code solution at comparable grid resolution, which has been termed "inter-code comparison." Often, the term "benchmarking" has been used in meeting papers wherein rigorous grid convergence testing has not been performed, and comparisons have not been precise; such Benchmarking exercises could be classified with Confirmation exercises above. [See Hutchings and Iannuzzelli (1987) for popular level examples of such non-rigorous benchmarking in fluid dynamics, and Rizzo (1991) for a popular level description of convergence studies for Finite Element structures calculations.] The more rigorous Benchmarking exercises could be classified with Verification or Validation exercises above, depending on whether the Benchmark is a rigorous numerical solution [e.g., see Ghia et al (1982), de Vahl Davis (1983), (Celik and Freitas, 1990), Celik et al (1993)]or a physical experiment. The "CFD Triathlons" of Freitas (1993c,1995b) were based on Benchmarks that are experimental data; thus, these exercises could all be described as Validation.

The term Benchmarking is also widely used for the distinct activity of comparisons not of accuracy, but execution speed, comparing different codes on the same computer, or different computers running the same code. On old computers (until the 1980's) such comparisons were relatively straightforward, dependable, scaleable, and repeatable. This is no longer the case with modern computer architectures, flexible system configurations and memory management, and compiler options. Particularly for purchasers of Workstations and PC's, speed Benchmarking on non-computational PDE codes from different manufacturers do not correlate well with computational PDE code performance (Dudebout and Fahs, 1996).

Precise inter-code comparisons could be used in principle to rigorously Verify a code. If Code A has been rigorously Verified against analytical solutions, it could be used to approximately (in an ordered sense) reproduce another analytical solution that could be used legitimately as the basis of comparison for Code B, a kind of indirect Verification of Code B. This would be straight-forward if both codes produced essentially the same discrete solution at all grid refinements, e.g. both used simple linear FEM and differed only in the solver, so that the answers from each agreed to within the iteration convergence tolerance. Less straight-forward would be the Verification of a 4th-order accurate Code B by comparison with a previously Verified 2nd-order accurate Code A, though it could be done convincingly.

Unfortunately, in practice, inter-code comparisons are usually rather loose; e.g., see the PSACOIN exercises (Nies et al, 1990) and the international projects INTRACOIN, HYDROCOIN and INTRAVAL (Larsson, 1992). In the HYDROCOIN exercise (OECD, 1988), "Level 1" was titled "Code Verification," yet *no* participating group Verified the theoretical rate of convergence. Surely some agreement of (say) a new code with an old and widely used code could legitimately build some confidence. Note, however, that there is nothing to be gained by demonstrating agreement with an inaccurate code! This occurs surprisingly often. Also note that a good code (say a 2nd- or higher order accuracy code) may appear to the naive to be inaccurate when its results are compared with results of four or five 1st-order or hybrid codes, all of which are in rough agreement. (Furthermore, the 1st-order solutions are smoother, and often appeal more to the naive user's intuition.) In such a case, the statistical "outlier" is more credible. Applied mathematics is not an exercise in democracy! Also, note that group exercises in inter-code comparisons invariably are not only

comparisons of code accuracy *per se* but also exercises in conceptual modeling and code user skills as well. [See e.g., Nies et al (1990), (Larsson, 1992), (Freitas, 1993b, 1995), de Vahl Davis (1994), (Muller, 1994).] Furthermore, the exercises challenge the ability of organizers to create meaningful comparison problems.

2.6 CODE CERTIFICATION, QUALITY ASSURANCE, AND ACCREDITATION

"Code Certification" goes beyond Code Verification and Validation, and is used with less uniformity; see definitions in Mehta (1989, 1991, 1992a,b, 1995, 1996), Melnik et al (1995), Rizzi and Vos 1996). Following Mehta (1988), Aeschliman et al (1995) defined Code Certification as "the entire process of establishing the credibility of a code, i.e., a Certified code has been Verified, Calibrated, and Validated." (See Section 2.12 below on "Calibration and Tuning".) In a later publication, Mehta (1996) described code Certification as consisting of Verification and Validation of a code, plus rules for its use (on a project) and complete documentation. He also spoke of certifying the simulations as well as the code. In the field of groundwater modeling, Tsang (1991) simply equated Certification of a code with proper Verification plus proper documentation. The aerospace consortium experience in Verification, Validation and Certification reported in Melnik et al (1995) used the term Certification similarly but with an additional distinction. They associated Certification with a different set of variables than Validation. While both involve experimental data comparisons, typically (at least in the aerospace community) Validation has focused on simple geometries usually involving a single dominant physical phenomenon with data comparisons to fundamental flow quantities (velocities, pressures, etc.). In contrast, Certification must focus on the needs of the aerospace design process, which involve more complex geometries or complete configurations involving multiple physical phenomena, with data comparisons to performance quantities like lift and drag coefficients, nozzle efficiencies, etc. In our opinion, this distinction seems to be non-essential. Surely one could compare to experiment and Validate or invalidate any quantity. Our position agrees with Porter (1996), who calls for the Validation process to "address at least three levels of flow problems: fundamental physics, unit or component flow problems, [and] overall systems."

Many authors in aerospace (e.g., Melnik et al, 1995) have spoken of three levels of codes, namely research, pilot, and production codes, and associate Certification only with the production level. However, Validation (in the general sense of comparison with experiments, perhaps for a more limited range of parameters than Certification) is not so restricted. That research and pilot codes should also be compared to experiments was the opinion of Marvin (1995), Melnik et al (1995) and Rizzi and Vos (1996), all of whom also adopted the same three-level description code development.

Rizzi and Vos (1996) also described Certification somewhat differently than Melnik et al (1995), linking it more with Verification and mathematics rather than with Validation and science. "Verification and Certification address the accuracy of the computational model, and...Calibration and Validation are concerned with the suitability of the physical model." Also, "Certification is concerned with programming issues, e.g. logic checks, programming style, documentation, and Quality Assurance issues, e.g. rerunning the suite of Certification test cases after each new version release in order to be certain that no new errors have been introduced into the previously Certified version." They also included in Certification the issues of model options and robustness. They quoted the definition of Certification (above) from Mehta; however, Mehta included Validation in Certification, whereas Rizzi and Vos did not. (They did not note the inconsistency.) Thus we see that "Certification" can and has been taken to refer only to Verification-type activities, or to include project-oriented Validation activities as well, by different respected authorities in the same sector of computational engineering (aerodynamics). Sindir et al (1996) described their four-phase "Validation" procedure (outlined in Chapter 10) which would perhaps fit the description of "Certification" as used by the previously cited authors.

Generally, I find Certification and Code Quality Assurance (QA) to be practically indistinguishable, with the different terms being preferred in different engineering activities. As contrasted to Code Verification and Validation, Code Certification or QA is usually not associated with mathematics or science so much as engineering and engineering management. Certification is more of a *programmatic* concept than a scientific concept. "Code Certification" appears to be the preferred term in the Aerospace field, (Mehta, 1989, 1991, 1995; Melnik et al, 1995; Barber, 1996; Porter, 1996) while "Code QA" is preferred by regulatory agencies involved with groundwater remediation (WIPP PA, 1992). "Code Accreditation" appears to be simply the process of some authority (perhaps legal or regulatory) officially declaring a code to be useable for a specific project (e.g., see Mehta, 1996); no general guidelines are discernible. In general categories,

"*Verification*" ~ *mathematics*
"*Validation*" ~ *science/engineering*
"*Certification*" ~ *engineering management*

Contributing to the lack of uniformity in usage is the fact that there are two extremes of Code Certification, the smallest at the scale of the isolated code, and the largest at the scale of the engineering project. Suppose we have a code that is rigorously Verified, and then acceptably Validated for a limited set of physical problems and parameters. One could still not claim Certification or QA unless the code were documented, etc. And then, even if the code were Certified by some set of criteria, one might not be able to claim Certification for the project or *programmatic* needs.

My inclination would be to include in Code Certification the activity of Code Verification independent of the code authors, i.e., Code Confirmation, the adequacy of Validation tolerance (i.e. the pass/fail evaluation which is project-determined), other aspects of code QA including robustness, documentation (internal, external, users manuals), version control, the QA system itself including such clearly non-mathematical but practical considerations as the selection process for review committees, signature authority for change orders, forms for bug reports, conflict resolution procedures, etc.. For project-level Certification, one would also need Code Validation and pass/fail evaluation for a project-oriented range of parameters. For several authors, a principal distinction is that Validation involves comparisons to experiments primarily on model problems while Certification involves comparisons to experiments on more realistic geometries and range of operating parameters (e.g., Melnik et al, 1995; Barber, 1996). Note that "code Certification as a design tool requires specific criteria/metrics" (Melnik et al, 1995). Barber (1996) also noted that a research group (associated more with Validation) is typically interested in fundamental flow variables (streamlines, velocity profiles, etc.) whereas a design group (associated with Certification) is typically interested in performance oriented variables (lift and drag coefficients, system efficiency, etc.). Barber (1996) included Risk Assessment and Reduction in the Certification process.

For project-oriented engineers, the activities of Code Verification/Validation/Confirmation almost form a continuum, and the three terms are often used as a suite and even an acronym, "VVC." (See Melnik et al, 1995; Porter, 1996; Aeschliman et al, 1995.) For them, it is natural to see a progression from "research codes" through "pilot codes" to "production codes" (Melnik et al, 1995). However, the qualities associated with each category by Melnik et al, while perhaps representative of aerospace practice, are sometimes arbitrary and unnecessary. For example, there would seem to be no inherent reason why a code with a "limited range of application" could not become a "production code," nor why a "research code" or a "pilot code" could not have thorough documentation.

Likewise, Melnik et al (1995) and Rizzi and Vos (1996) described Validation as progressing through higher stages, from Benchmarking for simple flows exhibiting one single dominant flow feature and simple geometry to multi-feature flows and complex geometries. However, we would prefer to allow Validation to

be considered thorough, i.e. at a high stage, for even a limited class of problems. The progression they described would then not be defined as higher stages of Validation, but simply as Validation for a more general class of computational PDE codes.

2.7 VERIFICATION OF CALCULATIONS (SOLUTIONS)

Just as it is critical to distinguish Code Verification (correctness of the mathematics) from Code Validation (correctness of the science), so we must distinguish between Verification of Codes vs. Verification of (individual) Calculations or Solutions.

A code may be rigorously verified to be (say) 2nd-order accurate (at least for a benign problem), but when applied to a new problem, this fact provides no estimate of accuracy or confidence interval. It is still necessary to band the numerical error for the individual calculation, usually (and most reliably) by performing grid convergence tests. It would be preferable to have different words for these two "Verification" activities, but I am at a loss for a clarifying term. The very important point, independent of the semantics, is that *the use of a Verified Code is not enough*.

This point is possibly well recognized by present readers, but is not universally so. Especially in the commercial code arena, user expectations are often that the purchase and use of a "really good code" will remove from the user the obligation of "doing his homework," i.e. the tedious work of Verification of Calculations via systematic grid convergence studies. This unrealistic hope is sometimes encouraged by salespeople and by advertising.

The methodology for Verification of Calculations is a major theme of this book; see Chapters 5, 6, 8.

2.8 QUANTIFICATION OF UNCERTAINTY

The term "Quantification of Uncertainty," as used in the early ASME Workshops (e.g., Freitas, 1993b; Celik et al, 1993; Johnson and Hughes, 1995) and elsewhere, can refer both to Verification of Codes and to Verification of (individual) Calculations as described above. The term also allows for inclusion of both error "estimation" and the more conservative error "banding," which includes a "factor of safety" as in the Grid Convergence Index (see Chapter 5) and related methods. Quantification of Uncertainty is distinguished from the more amorphous "Confidence Building" by the key word "Quantification," and is less project-oriented than Code Certification or Quality Assurance (see above). Etymologically, it could include Validation, of course, but whether or not this is intended must be inferred by its context. One authority (Oberkampf, 1998) prefers to restrict the word Uncertainty to technical definitions (as in Section 2.3.2.2). This usage would in fact be preferable, but unfortunately the broader use of Uncertainty is already widespread.

2.9 GRID CONVERGENCE VS. ITERATION CONVERGENCE

The literature commonly uses the term "convergence" in two completely different ways. Readers probably will know the distinction between iteration convergence vs. grid convergence (or residual accuracy vs. discretization accuracy). Usually, the meaning is clear from the context, but sometimes confusion occurs, e.g. when some new variant of the SIMPLE algorithm (Patankar, 1980) is presented as being "more accurate." The accuracy claimed here often is residual accuracy, i.e. what is better called iteration convergence accuracy or iterative speed, and has nothing to do with the order of accuracy of the discretization.

For the present subject, we note that iteration convergence can muddy the distinction between Code Verification and Calculation Verification, since iteration tuning parameters (e.g., multigrid cycles, relaxation factors, etc.) can be problem dependent. Most importantly, incomplete iteration convergence will corrupt Verification.

2.10 ERROR TAXONOMIES

2.10.1 Inadequate Error Taxonomies

Several taxonomies of errors given in the earlier literature were inadequate and misleading, in our opinion. Not all lists are taxonomies. For example, the list "flora, fauna, mammals, dogs" is not an adequate biological taxonomy. All items are certainly part of biology, but mammals are not separate from fauna, rather they are part of it, as are dogs. In a true taxonomic classification, each element of the taxonomy is counted once and only once. In the list "flora, fauna, mammals, dogs," consisting of four categories, the family pet is counted three times! Biologists (and logicians) are well aware of difficulties and sometimes arbitrariness in taxonomic classifications. There can be gray areas and elements that do not fit the supposedly complete classification system, e.g. the platypus, hermaphrodites. Also, the same set of individuals can be classified by more than one valid taxonomy, e.g. automobiles might be classified according to year of manufacture, or country of manufacture, or color, etc. But there are also clearly false taxonomies, and inappropriate or inadequate taxonomies, and we see these sometimes in computational PDEs.

Specifically, a listing of "sources of error" as seen often in the literature is not a taxonomy appropriate for Verification of codes or calculations. The "grid generation errors" (Ferziger, 1993; Celik and Zhang, 1995 or "grid resolution/distortion" errors (Bobbitt, 1988; Barber, 1996) or "factorization errors" (Rumsey, 1988) are not separate from discretization errors for purposes of error estimation. For the Verification of a code or a calculation, there are no such things as separate "grid generation errors," nor are there separate "errors associated with coordinate transformations" (Ferziger, 1993). Indeed, bad grids do add to discretization error *size*, but do not add new terms other than discretization errors. This does not mean that one grid is as good as another, or that a really bad grid cannot magnify errors, but only that these so-called grid generation errors do not have to be considered separately from other discretization errors in a grid convergence test. If the grid convergence test is performed, and the errors are shown to reduce as $O(\Delta^2)$, for example, then all discretization errors are verified. One does not need to separately estimate or band the grid generation errors.

Such listings *are* useful for analysis of algorithms. For example, one can study the effect on (formal) accuracy of grid angle θ. We know that as two families of coordinate lines approach parallelism ($\theta = 0$) the transformed PDEs (continuum and discretizations) degenerate. For θ small, the effect is to increase the size of the coefficients in the error expansion, but not the *order*. It has often been stated in the grid generation literature that strong coordinate stretching reduces the order of centered differencing in logical space to 1st order. In fact, it can be demonstrated (confirmed by numerical experiments) that severe stretching like $\chi = a \cdot \exp(x)$ or even the ridiculous $\chi = a \cdot \exp[\exp(x)]$ greatly increase the size of the error (with 6-figure coefficients) but that the formal and observed orders remain 2nd-order (Steinberg and Roache, 1985).

Likewise for the proposed numerical error bar (Karniadakis, 1995; see also Vanka, 1995) shown in Table 2.10.1. The contributions consist of separately estimated numerical errors from boundary conditions, computational domain size, temporal errors, and spatial errors. This is an inadequate taxonomy, since the

category boundary errors can include ordered terms like wall vorticity (when vorticity is a dependent variable) standard forms for which (Roache, 1998b) can be $O(\Delta)$, $O(\Delta^2)$, $O(\Delta^4)$, etc. Such ordered errors will tend to zero as the discretization improves, so that a boundary error from wall vorticity evaluation need not be considered separately from the grid convergence study. The category of boundary errors would also include far-field boundary conditions, which are not ordered in Δ, i.e. the error persists even in the limit of $\Delta \rightarrow 0$. This does not mean that all far-field boundary conditions are equally bad; some are better than others, but *none* vanish as $\Delta \rightarrow 0$. On the other hand, *all* vanish as distance to the far-field boundary $L_b \rightarrow \infty$. This boundary error will improve as the computational domain size increases. The taxonomy already includes temporal errors and spatial errors (which are evaluated in the grid convergence study) and computational domain size errors, so that both the ordered boundary errors and the non-ordered (outflow, or "free") boundary errors are already counted elsewhere in the taxonomy. Since the intention is to provide a quantitative breakdown in the sources of numerical error in an error band, the taxonomy is inappropriate for the Verification of Codes and Calculations.

- boundary conditions
- computational domain size
- temporal errors
- spatial errors

Table 2.10.1 Example of an Inadequate Taxonomy for Error Estimation

Note, however, that "free" outflow boundary errors may prove to be ordered not in Δ but in $1 / L_b$ where L_b is the distance from the region of interest (e.g., an airfoil) to the outflow boundary. [See the data of Zingg (1992), shown in Chapter 6 to be 1st-order in $1/ L_b$.] Thus, this source of error can be evaluated by purely numerical tests, by varying the position of the outflow boundary.

In contrast, consider the commonly used approximation for near-wall density ρ of $\partial\rho / \partial n = 0$ for compressible flow, a modeling approximation that (although commonly used and accurate in some sense) cannot be evaluated easily, i.e. its error does not vanish as $\Delta \rightarrow 0$ nor with any other limit of a modeling parameter such as L_b. Thus, this error of the wall density condition is more resistant to modeling accuracy than the commonly acknowledged free outflow or far-field computational boundary. (In fact, it will get worse as Mach Number increases and $(T_w - T_{adiabatic})$ increases.) This is really a conceptual modeling error. Like incompressibility, or two-dimensionality, it cannot be evaluated numerically using only the code being Verified, but must be evaluated by comparison with some external benchmark - numerical, experimental, or theoretical.

2.10.2 A Meaningful Error Taxonomy

By contrast, the list of error sources provided by Oberkampf et al (1995) is an appropriate and meaningful taxonomy. They gave four categories of errors, shown in Table 2.10.2. A slightly expanded (though not necessarily exhaustive) taxonomy will be suggested shortly.

- physical modeling errors
- discretization errors
- programming errors [i.e., mistakes]
- computer round-off errors

Table 2.10.2 A Meaningful Error Taxonomy (from Oberkampf et al, 1995)

2.10.3 Gray Areas: "Justification"

The subject of outflow (or open) boundary conditions does produce some fuzziness in categorization of Verification vs. Validation, in my opinion. The error can be ordered in $1 / L_b$, as above, and therefore can be part of Verification. That is, it is up to the user (who is doing the conceptual modeling) to estimate or band the error caused by the position of the outflow boundary. But if the code has some sophisticated free outflow condition, e.g. a simple vortex condition for Euler equations (Thomas and Salas, 1985), or the "fringe method" in DNS turbulence calculations (Marvin, 1995), then the distinction is not as immediate. Certainly the equations used are clear, and the code may "solve the equations right" (i.e., Verification), yet there exists another benchmark solely from the mathematics (the case with infinite boundary distance) which could be used to justify the outflow condition without recourse to physical experiment (which would clearly be Validation).

In any case, we agree with Oberkampf et al (1995) that there is no free outflow or open boundary condition that is *true*, *correct* or *exact* in any general sense. ("What are the exact boundary conditions for the partial differential equations on an open boundary?" is a meaningless question.) And while improvements in free outflow boundary conditions are possible, they are only capable of improving efficiency by allowing computations on a smaller numerical domain for the same error tolerance as present methods. But in no way are improvements a necessity for obtaining Verified solutions. Methods presently exist that will give the correct answers, to within an arbitrary tolerance, if the position of the free outflow boundary is varied and its influence assessed, i.e. if the effect is Verified quantitatively. On the other hand, consider the use of "fuzzy boundary conditions," a term introduced by Sani and Gresho (1994) to "suggest the existence of numerical BC's that produce good numerical solutions, but if one tries to take the limit as $\Delta \rightarrow 0$ of these BC's, one obtains unacceptable BC's for the PDEs" (Oberkampf et al, 1995). In my opinion, these fuzzy boundary conditions should not be used, since they make convincing Verification by rigorous grid convergence testing impossible.

Another example of semantic failure or fuzzy taxonomy arises when we consider Benchmarking a code that is based on simplified equations against a code using more complete equations. Examples include a turbulent boundary layer code or Parabolized Navier-Stokes (PNS) code benchmarked against a Reynolds-stress averaged Full Navier-Stokes (FNS) code, and a groundwater flow code using the Dupuit approximation against a code using the full nonlinear free surface (phreatic) equations [Serrano, (1995), (Knupp, 1996), (Knupp et al, 1996)]. Presume that both codes are convincingly Verified, i.e. they correctly solve their respective equations. Suppose that the PNS code results agree well with the FNS code results for some range of parameters (e.g., including angle of attack). This agreement is not strictly included in the term Verification, since the Verification of the PNS code has already been completed prior to the FNS benchmarking. Then we could say that the agreement has demonstrated that the PNS code is "solving the right equations" in one sense, i.e., it justifies the use of parabolic marching. Yet to claim Validation would be over-reaching, since we have not demonstrated the adequacy of the turbulence model by comparison

with experiment. We have "solved the right equations" only in an intermediate sense of demonstrating that the PNS equations adequately represent the FNS equations, but not in the ultimate sense of "solving the right *physical* equations."

Appealing to the other distinction between Verification and Validation based on mathematics vs. science, it is clear that such a comparison exercise should be categorized with Verification rather than Validation, but that the categorization is somewhat inadequate because the PNS code can (should) have been fully Verified in the more usual sense before the comparison exercise was started. But if the only choice is Verification or Validation, it is safer to reserve the term Validation for experimental comparisons.

In some situation where nit-picking would be excusable, the solution is to introduce a finer-scale of categorization, perhaps using the word "Justification," technically defined in this specific context. The term is in common use in theoretical papers, as when one "justifies" the assumption of some simplification to the accepted full theory, e.g. a small perturbation. In situations dictating a legalistic distinction, (e.g., code review by a regulatory agency) one could claim Justification of the simplification of mathematical models (Full Navier-Stokes to boundary layer, or compressible flow to incompressible flow, or variable properties to constant properties, or small-scale isotropy of turbulence, or Dupuit approximation, etc.) while making it clear that the physical Validation remains to be accomplished.

Certainly one can understand the issues in the above examples, and do good work, without introducing new terminology like Justification. For example, in discussing the "fringe method" for inflow and outflow boundary conditions for DNS turbulence calculations, Marvin (1995) used the overall context of Validation, and says the comparisons "confirmed the appropriateness," obviously using "confirm" in a general sense rather than the more limited technical sense proposed herein. Fortunately, anyone with some technical knowledge of the subject area and some common sense understands the meaning behind the words without needing an overly precise, tiresome, legalistic, semantic distinction between Validation and Confirmation as the terms are used by Marvin and others, and the terms Validation, Verification, Confirmation, and now Justification as proposed here. Unfortunately, such "mere semantics" may become of vital interest when dealing with regulatory agencies such as the EPA, or with legal definitions in a NASA contract, etc.

2.10.4 An Expanded Error Taxonomy

The taxonomy of errors shown in Table 2.10.4.1 is an expanded version of that given by Oberkampf et al (1995). It still has some gray areas, but is appropriate for purposes of Verification of Codes and Calculations.

- Errors ordered in discretization measures Δ; these errors can be evaluated by grid convergence studies.
- Errors ordered in some numerical (rather than physical) parameter not associated with discretization (like L_b); these errors can be tested numerically in the code being Verified.
- Errors ordered in some physical parameter.
- Non-ordered approximations (like $\partial\rho / \partial n = 0$ or $\partial P / \partial n = 0$) that are conceptual modeling errors.
- Programming errors (unlike the other errors, these errors are *mistakes*); these can be detected by grid convergence studies for a problem with an exact solution (see "Method of Manufactured Solutions" in Chapter 3).
- Computer round-off errors; these errors can be identified by grid convergence studies or *ad hoc* approaches, but often they are simply demonstrated to not be significant.

Table 2.10.4.1. An Error Taxonomy Appropriate for Verification of Codes and Calculations.

Each category in the Taxonomy of Table 2.10.4.1 will now be elaborated upon.

- Errors ordered in discretization measures Δ; these errors can be evaluated by grid convergence studies.

  The discretization measures Δ can include each space coordinate, time, or N vortices, N Fourier modes, etc. The category includes ordered artificial viscosity (e.g., Roache 1998b) but this can be tricky to categorize and estimate if the parameters of the artificial viscosity (or better, artificial diffusion) change with the discretization, and with hybrid methods.

- Errors ordered in some numerical (rather than physical) parameter not associated with discretization (like L_b); these errors can be tested numerically in the code being Verified.

  This category notably includes the location of idealized far-field boundaries, such as distance from a wing to the downstream (outflow) boundary L_b, or the cross-stream distance from a wing to the free-stream conditions L_∞. Note that these could be argued to be physical conditions rather than numerical parameters, since the same kind of consideration exists for wind tunnel tests (Roache, 1998b). That is, one could think of numerically modeling the wind tunnel walls with no numerical error associated with the location of the walls, and relegate the conceptual modeling error (compared to the desired simulation of flight in the atmosphere) to the wind tunnel concept. In practice, the assignment to non-discretization numerical errors is probably preferable, since the effect may be evaluated by purely numerical tests in the code being Verified (i.e., it may be considered to be a mathematics problem rather than a physical science problem, and therefore Verification rather than Validation) and since the numerical simulation at the walls can be *better* than the physical wind tunnel simulation. For example, why model a physical no-slip wall with a boundary layer, when a slip condition better approximates the condition of free flight? Conditions at the quasi-free stream location (upper boundary) that are even less restrictive are available; the simple expansion wave approximation (Allen and Cheng, 1970; Roache, 1998b) is not just a slip wall, but is a better model of the free-flight condition. Note that the actual order (in $1 / L_\infty$) may be unknown *a priori*, but at least we know the limit is good and the error can be investigated by straight-forward numerical experimentation (Roache and Mueller, 1970; Roache, 1998b). Also, the order may be determined by numerical experiments and sometimes by theoretical considerations, e.g. of potential flow (Roache, 1998b). Similar considerations would apply to the modeling in groundwater flow and transport calculations, in which the far-field condition may be conceptualized as a distant drainage divide and simplified in several ways.

- Errors ordered in some physical parameter.

  These are physical or conceptual modeling errors, the order of which might be determinable using perturbation methods (Van Dyke, 1975; Wilcox, 1995). Examples are $Mach \rightarrow 0$, the Boussinesq approximation for natural convection, number of fractures or fracture width in a dual porosity formulation, skin resistance, etc.

- Non-ordered approximations (like $\partial\rho / \partial n = 0$ or $\partial P / \partial n = 0$) that are conceptual modeling errors.

  The condition $\partial\rho / \partial n = 0$ at a wall is not quite identical to the boundary layer approximation, and is a better (less restricted) approximation than $\partial\rho / \partial y = 0$ throughout the boundary layer, but is commonly applied in situations where the boundary layer approximation is not valid (e.g., stagnation

regions, shock-boundary layer regions). Another example is non-ordered artificial viscosity. The inherent artificial viscosity of nonlinear flux limiters may be difficult to classify. Errors in this non-ordered category might become graduated to the previous ordered category by later analyses. Discrete vortex methods may be difficult to classify, but the solution should get better in some sense as the number of vortices increases. The solution may "converge" in the sense of approaching a limit, but whether that limit approaches the full Navier-Stokes equations, or the inviscid Euler equations, or something else, may be open to question.

- Programming errors can be detected by grid convergence studies for a problem with an exact solution (see "Method of Manufactured Solutions" in Chapter 3).

- Computer round-off errors

Round-off error is sometimes confused with discretization error, and to emphasize the distinction people often speak of "computer round-off error" or "machine error." Round-off errors arise from the fact that computers do not work in the real number system, but only with a finite word-length subset of the real number system. They will be discussed further in Chapter 4.

2.11 TRUNCATION ERROR VS. DISCRETIZATION ERROR

"Truncation error" is an unfortunate term. Strictly speaking, it refers to the truncation at some finite value of a series, which could be analytical (e.g., the Sudicky-Frind solution, 1982) or more commonly in the present context, the Taylor-series expansion of the solution, which is the basis of developing the finite difference equations. It is a worthwhile concept because it allows one to define the "order" of the finite difference method (or finite element, finite volume, etc.). Unfortunately, the term is often used loosely in the sense of "discretization error," i.e. the error that is caused by the fact that we can only use a finite number of grid points (or another measure of discretization)[11]. In a FDM, one cannot take the limit of infinite order (i.e., limit of zero truncation error) without also taking the limit of infinite number of grid points, since high order methods require higher support. This terminology makes the limit process somewhat misleading, in my opinion. Also, it confuses the issue of solution smoothness with discretization error, since the Taylor series expansion depends on smoothness.

In the context of grid convergence tests, it is preferable to not speak of evaluating the "truncation error" of a numerical approximate solution, but rather the "discretization error" that arises because of the finite discretization of the problem. This terminology applies to every consistent methodology: FDM, FVM, FEM, spectral, pseudo-spectral, vortex-in-cell, etc., regardless of solution smoothness. (By "consistent" we mean, of course, that the continuum equations are recovered in the limit of infinite discretization.) The term truncation error is then reserved just for the examination of the order of convergence rate of the discretization. Note again the point of a taxonomy; these two errors are not independent. For any finite grid calculation, we do not have a truncation error (arising from the use of finite-ordered FDM, say) that we *add* to the discretization error (arising from the use of a finite number of grid points). And it is not possible to approach the limit of zero truncation error by arbitrarily increasing the order of the FDM or FEM without increasing the discretization. (Note in FEM we could fix a finite number of elements but we would still have to increase the discretization, i.e. the support within the elements.) However, the alternate is true: we can in fact approach the limit of eliminating all the discretization error by arbitrarily increasing the

[11] Discretization error can be precisely associated with truncation error for smooth linear problems, but not so easily for nonlinear problems.

number of grid points without changing the order of the method. Thus, discretization error is the preferable term for speaking of the numerical error of a calculation, and truncation error is not separate in the taxonomy of an error estimate for a calculation. (However, the order of the truncation error is still verified in the code verification.)

Truncation error has also been defined (Ferziger, 1993) as the residual resulting from substitution of the continuum (exact) solution values into the discrete equations on some grid. This definition can be useful for analysis of discrete methods. But again, it is not distinguishable from discretization error in an accuracy estimate of a real calculation.

2.12 CALIBRATION AND TUNING

The term "Calibration" is used with more latitude than Verification, Validation, or Confirmation. We prefer to use the term "Code Calibration" to mean the adjustment of parameters needed to fit experimental data, e.g. the 6 closure coefficient values necessary for two-equation turbulence models. Many practitioners agree, e.g. Marvin (1995), who gave the following description.

"Code parameters such as turbulence models may need to be adjusted to accommodate applications for geometries and conditions outside the envelope of their original validation. Experiments intended to support this activity can be referred to as Calibration experiments." (Marvin, 1995)

As an example of Calibration, Marvin (1995) cited the work of Coakley and Huang (1992) in which basic k-ε and k-ω turbulence models were [rationally] corrected for compressibility and length scale, vastly improving the experimental agreement for surface pressures and heating rates on an ogive-cylinder-flare body at Mach number = 7. Importantly, the good agreement was maintained for other hypersonic experiments on shock interactions.

Likewise, Mehta (1996) was clear that "Calibration is not the process of determining the level of accuracy or credibility, but is the process of obtaining correction factors." Consistent with Marvin and with Mehta, Porter (1996) noted the dictionary definition of Calibration: "to standardize by determining the deviation from a standard so as to ascertain the proper correction factor." Rizzi and Vos (1996) were also consistent with this view. "Calibration is the process of Tuning or Calibrating a code with a particular fluid dynamics model to improve its prediction of global quantities for realistic geometries of design interest. This has to be done because there is no universal turbulence..."

However, other colleagues have assured me that the term Calibration has been used in experimental studies just as a means of ascertaining accuracy, or more properly of determining the inaccuracy, e.g. of a pressure probe or a wind tunnel test section flow. (Oberkampf, 1994 noted that the term Validated is never used for an experimental ground test facility because it would be a misnomer. Rather, facilities are Calibrated.) If extended to codes as in Aeschliman et al (1995), this definition would make Code Calibration almost indistinguishable from Validation, or perhaps Validation for a more restricted range of parameters. Indeed, Porter (1996) gave the definitions adopted by a NASA *ad hoc* committee.

- *Validation*: "comparison with experiment to verify [*sic*] the ability to accurately model over a range of parameters [*sic*].
- *Calibration*: comparison with experiment to provide a measure of the ability to predict specific parameters [*sic*]."

Even if we ignore the unfortunate use of term "verify" in a common- language (rather than technical context) sense, and overlook the use of "parameters" in two different senses (the first as an input parameter, the second as solution value), these NASA definitions still offer little distinction, with

Validation involving a *range* of parameters but Calibration involving *specific* parameters. See also Bradley (1988).

Aeschliman et al (1995) used (as have others) the triplet "Code Verification, Calibration, and Validation" or VCV and gave this description of Calibration. "We loosely interpret code 'Calibration' to mean a code's ability to reproduce valid data (not exclusively experimental) over a specified range of parameters, for some geometry, without necessarily assessing the overall correctness of all of the physical models employed. We consider Calibration to be a less-demanding element of Validation, and is addressable experimentally by the same methods." For CFD, we prefer Marvin's (1995) description above.

Bradley (1988) summarized the substance of a NASA study of a process for CFD Validation, and distinguished code Validation from Calibration as follows. (The quote is from Marvin, 1995.)

"CFD code Validation: Detailed surface and flowfield comparisons with experimental data to verify the code's ability to accurately model the critical physics of the flow. Validation can occur only when the accuracy and limitations of the experimental data are known and thoroughly understood and only when the accuracy and limitations of the code's numerical algorithms, grid density effects, and physical basis are equally known and understood over a range of specific parameters. CFD code Calibration: The comparison of CFD code results with experimental data for realistic geometries that are similar to the ones of interest, made to provide a measure of the code's ability to predict specific parameters that are of importance to the design objectives without necessarily verifying that all of the features of the flow are correctly modeled."

Note that the term Calibration also has traditionally been applied to the empirical adjustment of constants in a *theoretical* analysis, as in Bradley (1988). Marvin (1995) went on to state that "During the intervening years since these definitions were formulated [1988], it has been argued that the definition of Validation is too restrictive, especially for the complex applications associated with realistic geometries. Nevertheless, NASA has maintained the definition as a goal of its Validation process." [See also NASA (2008).]

In our opinion, definitions or descriptions should avoid enforcing levels of accuracy, which are inevitably vague anyway (e.g., "detailed," "thoroughly understood," etc.) and leave these qualifications simply to the evaluation of the *thoroughness* of the Validation (or invalidation). Also, as noted earlier, assessment of "grid density effects" is not a code property *per se*, but rather a property of the particular code *application*, i.e., part of Calculation Verification rather than Code Verification.

Many people simply equate Calibration with adjustment of parameters called "Tuning," and often "somewhat without scientific justification" (Mehta, 1996)." Rykiel (1996), addressing ecological modeling, gave this definition: "Calibration is the estimation and adjustment of model parameters and constants to improve the agreement between model output and a data set." This essentially agrees with NASA (2008)" "The process of adjusting numerical or modeling parameters in the model to improve agreement with a referent." In addition to such "model Tuning," one may tune, perhaps legitimately, to correct for under-resolution in engineering parametric studies, especially for trends (Oberkampf, 1998) but this is a dangerous practice. The faintly pejorative association of model Tuning is deserved if every new data set requires re-tuning, but not so if reasonable universality is obtained. Bradshaw (1992) noted that "a simple model which has been carefully calibrated" [for a particular problem] may out-perform more advanced models on its home ground. This may be the First Law of Turbulence Modeling."

Although these and other earlier uses often described calibration as just validation for a restricted range of physical parameters, it has become apparent that this definition is dangerous and has been abused. It is necessary to be clear that calibration, the adjustment or tuning of free parameters in a model to fit the model output with experimental data, is not validation. (See Appendix B for references.) Calibration is a sometimes necessary component of (strong sense) model development. But this calibration is not to be

considered as validation, which occurs only when the previously calibrated model predictions are evaluated against a set of data not used in the tuning. There is no value in tuning free parameters to obtain a drag coefficient to match an experimental value, and then claiming code/model validation because the "prediction" agrees with the same experiment. Historically, this has been a common failing of free-surface flow modeling projects (ASCE, 2009). Of course, if all point-values and functionals of interest are well matched using a small set of free parameters with physically realistic values, this will tend to be convincing in itself, but another data set not used in the tuning will be more so. The most important point is the following.

Calibration is not Validation.

It is important to recognize that all of these activities (Verification of calculations, Calibration, Validation when used in the sense of including pass/fail evaluation, and especially Certification) have associated with them error tolerances that cannot be arbitrarily defined universally but must be defined with reference to their intended uses (Marvin, 1988; Mehta, 1995; Melnik et al, 1996). Also, as noted above, Certification is a *programmatic* concept, rather than a scientific concept.

2.13 QUALITY ASSURANCE (QA) VS QUALITY WORK

Since we are making semantic distinctions in the context of confidence building for computational PDEs, it is worth making the distinction between the technical term "Quality Assurance" or QA vs. "quality work." QA boosters like to talk as though the two were equivalent, but formal QA is largely a system of paperwork (e.g., see ISO, 1991) run by managers. A project can meet all the formal QA requirements and still be low quality (or flatly erroneous). On the other hand, high quality work can and has been done without a modern and formal QA program. (The legendary Blackbird spy plane was test flown in 1962, long before formal QA programs.)

QA can, of course, be helpful for quality work. (If nothing else, it can be used to encourage management to support quality work.) Even the formality and paperwork are helpful in issues like version control. But if allowed to run amuck, formal QA can mire real work in forms and procedures and definitions, impeding real quality rather than fostering it. For example, formal QA procedures may require expensive re-running of an entire suite of confirmation problems whenever a code is modified. This can discourage minor code fixes and improvements. As another example, I have seen QA reviewers complain and require written justification (with committee approvals, documentation of resolution, archiving of correspondence on approved forms, etc.) for a change in input data that produced numerical value changes from 3.0 to 2.999998. And this, in a geophysical problem wherein the parameter ranges were sampled over three orders of magnitude!

2.14 CUSTOMER ILLUSIONS VS. CUSTOMER CARE

The QA dictum that "the customer is always right" is adequate for a candy store operation, but not for any professional service. Catering to customer illusions can be antithetical to true customer care. (Indeed, the prerequisite task of *identification* of the QA customer is not obvious in many cases, e.g., university teaching.) This is the great QA philosophy failure; the customer is *not* always right, and we should not always give the customers what they want. (I have not read the original QA manifestoes, so I speak only from experience and second-hand sources of what QA originally was intended to be.) I do not want my physician, lawyer, or surveyor to take orders from me.

This distinction is pertinent in code user training and education. The limits of applicability of codes must be made clear. The best commercial code cannot be an "aerodynamicist on a chip" or a "groundwater modeler on a chip," i.e., the code cannot substitute for user expertise in the technical area. Also, this distinction of true "customer care" is most important in the area of code robustness. We all know how to build a robust CFD code, following what has become a CFD joke (Roache, 1994). "The good news about first-order upstream differencing is that it always gives you an answer. The bad news is that it always gives you the same answer, no matter what the Reynolds number." It is remarkable and disheartening to see how many industrial CFD practitioners will freely admit to caring little about numerical accuracy. (A director of a very successful consulting firm, specializing in simulation of enhanced oil recovery processes, stated that in some 20 years of experience he had *never* had a request from a customer for enhanced numerical accuracy.) For those who do care, we must not gloss over the limitations of CFD and give them a false sense of security.

2.15 OTHER DISTINCTIONS: AUTHORS, USERS, MODELERS, CODES, AND SOFTWARE

Another way to make distinctions is between the "code author" or "code builder" (which may of course be a team, including algorithm developers and programmers) and the "code user" or analyst. In many past situations (e.g., a graduate dissertation project) the same person performed both functions and therefore took the blame or credit for all aspects, but it is still worthwhile to make the distinction in functionality, especially with the rise of general-purpose commercial codes and software.

Another distinction is "constitutive equation developer" most typified in fluid dynamics by the "turbulence modeler." Again, the same person could and has functioned in all three capacities (code builder, code user, and constitutive equation developer) but we agree with Blottner (1990) that it is too much burden on one person.

In regard to the distinction made earlier between numerical errors and modeling errors, I cannot agree with Mehta (1996) and others that these may be viewed as precision and bias errors; see Chapter 9 for definitions of these terms as used in experimental error assessment, and Chapter 10 for complete interpretations.

Another distinction often made in the software engineering community is between "software" and "code." Briefly, code is defined as computer instructions and data definitions, whereas software is a more comprehensive term, including "programs, procedures, rules, and any associated documentation pertaining to the operation of a computer system ..." (Mehta, 1996) This distinguishes system software from applications software (like PDE codes) but, as readers will know, physical modelers usually overlap the terms, to no practical detriment.

2.16 Δ SENSITIVITY, UNCERTAINTY, AND RISK

Significant practical detriment does occur in public policy discussions that fail to distinguish between terms like *sensitivity*, *uncertainty*, and *risk*. Sensitivity calculations (often called "perturbation" calculations) usually refer to determination of the change of code output (either directly calculated quantities or inferred quantities) caused by unit changes in input data (including parameters). Essentially, this is equivalent to a numerical approximation to a partial derivative, say $\partial Q / \partial d$ where Q is an output quantity of interest and d is input data or parameter. Such data cannot be transformed into uncertainty in the results until a conceptual model is developed for the uncertainty distribution for the input parameters, which provide a value for Δd and thence uncertainty in output values $\Delta Q = \Delta d \cdot \partial Q / \partial d$. Such a concept

from elementary, intuitive level calculus is simple but still confusing to many public stakeholders in public policy debates. The next jump, from uncertainty to risk, is even more difficult. The simplest definition is Risk = Uncertainty × Consequence. That is, a high-level uncertainty (say, a 1% chance of a contaminant release to the environment) may produce an acceptably low level of risk if its consequence is small, e.g. a minor and temporary health problem.

Mehta (1996) stated that "Sensitivity studies usually consider extreme values of the model parameter," but representative or mid-range values would seem to be equally significant, or more so. (Note, however, that mid-range values may not exist, or be permissible, for bi-modal distributions.) Sensitivity studies (and the consequent uncertainty and risk studies) can also be assessed not just for input data and parameters but also for differing conceptual models (also referred to as structural uncertainty or model form uncertainty). As Mehta (1996) noted, "The method for assessing output uncertainty is much more developed for parameter uncertainty than for structural uncertainty." Note, however, that it is always possible, in principle, to transform a structural uncertainty into an artificial parameter uncertainty by using a blending function between models. The blending function could be applied to the outputs of the different models, or in some cases to internal parameters which blend one model structure into another. Also, in our view, the assessment of model form uncertainty as distinct from parameter uncertainty is precisely the activity of Validation; see V&V20 and Chapter 11.

Such parameter sensitivity studies are, in these taxonomies, neither part of Verification nor of Validation; they are simply part of the *results* of the modeling. (However, Oberkampf and Trucano (2007, 2008) persuasively stated that sensitivity analysis is an important part of Validation comparisons, providing deeper understanding to both the computational analysts and the experimentalists.[12]) This distinction is often lost in the (necessarily brief) reporting in journals, as when an author investigates the sensitivity of answers to some uncertainly specified physical parameter; this physical *modeling* sensitivity is usually reported in conjunction with purely numerical grid convergence tests. For example, Zha and Knight (1996) reported sensitivity tests for their difficult problem of three-dimensional shock wave/turbulent boundary layer interaction. They reported grid convergence tests by independent coordinate refinement (see Chapter 5), first doubling the grid in x, then in y, then in z. These are purely numerical tests. They also reported (contiguously) sensitivity tests on the thickness of the upstream boundary-layer and on different isothermal wall temperatures. These are physical modeling tests. (Both categories were convincingly performed.) In most papers, this distinction will not be made, and the results may be reported under a heading like "Solution Convergence Tests" or "Verification." The distinction is perhaps worth noting, but the only important point is that all the work be done and reported clearly, as in Zha and Knight (1996).

Blackwell and Dowding (2006) gave a review of modern methods of sensitivity analysis and uncertainty propagation in computational models. Their descriptions are also given in V&V20, Sections 3 and 5. An ASME position paper on "The Role of Risk Analysis in Decision Making" is given in ASME (2002).

[12] However, the recommended rank-ordering of importance based on "unit change" of input parameters is meaningful only for input parameters of the same type. There is no comparison of a unit change in free-stream Mach number to a unit change in model angle of attack.

2.17 ETYMOLOGY AND NEAR-SYNONYMS

As noted earlier, validation, verification and confirmation in common (non-technical) use are given as synonyms in a common English thesaurus. More complete definitions indicate extensive overlap of these words in some contexts, and with other related words such as substantiation, sanction, approval, authoritative, cogent, convincing, truth, authenticity, accuracy, genuineness, definite assurance, even logical. The Middle English/Latin root of "verification" is the equivalent of "truth," whereas the Latin root of "validation" is the equivalent of "strong." This etymology contributes, at best, a shade of meaning in favor of the distinction used herein. Others, e.g. Jameson and Martinelli, 1996, have seen a stronger etymological distinction. In any case, there is absolutely no justification for the extreme position taken by Oreskes et al (1994), arguing as though the etymology and common use have a compelling (almost magical) power over our use as technical terms.

Many other instances of near-synonyms and fuzzy usage occur. Mehta (1996) used the words "conceptual model" and "simulation model" distinctly, yet the essence of the distinction was not stated nor is it apparent. Mehta stated that credibility is established through the processes of Verification, Validation, and Certification, and he offered guidelines for various aspects of establishing code credibility, including developing and assessing models, Verification, Validation, and experimental tests. (Some of these suggested guidelines are arguable or obvious, most are useful, and all reflect the voice of experience.) Credibility is an easily understood term and is obviously desirable; as Mehta (1996) noted, "The significance of computer simulations depends solely on their credibility." Yet I am somewhat uncomfortable with the word, because if credibility becomes the stated goal of a project, then the work can devolve into image building and public relations rather than science and engineering.

Naturally, writers will use overlapping and/or inconsistent descriptions, and make distinction in their terminology only adequate for their intentions, which probably do not include philosophy of science. As stated earlier, it is best to recognize the significant arbitrariness in the original choice of technical usage, but still to recognize the widespread acceptance of a standard technical distinction where it exists, e.g. in Verification vs. Validation.

2.18 ACCURACY VS. RELIABILITY

Another worthwhile semantic distinction arises with the terms "accuracy" and "reliability," which are often used interchangeably in the engineering literature but which are, in fact, distinct. Consider the following hypothetical situation. You are solving an expensive time-dependent three-dimensional problem. Based on estimates from experience on related problems, you have a computing budget adequate to solve the problem on a $100 \times 100 \times 100$ grid. This would produce the most *accurate* solution you can afford. However, for about the same budget, you could produce two solutions on $90 \times 90 \times 90$ and $80 \times 80 \times 80$ grids. (The economics of the scaling for three-dimensional problems with $\Delta t \propto \Delta x$ indicate that the fine grid solution will cost $C \cdot 100^4 = C \cdot 10^8$ while the two grid solutions will cost $C \cdot 90^4 + C \cdot 80^4 \cong 1.07C \cdot 10^8$, about 7% more[13].)

The question is, how shall you spend your budget? The answer is, if you want *accuracy*, you use the single 100^3 grid, because this will produce the most accurate solution. However, although you know it is the most accurate solution that you can afford, you do not know how accurate it is! If instead you want

[13] This estimate is based on the assumption of optimal numerical methods with computing time $\propto$ to resolution. If, as is commonly the case, sub-optimal methods are used, the comparison is more favorable for the two coarser grids.

reliability, the answer is you use the two 90^3 and 80^3 grid solutions. The difference between the two solutions gives you some idea of the accuracy. You have more *confidence* in the 90^3 solution; you can *rely* upon it. You have a somewhat *less accurate* solution but it is *more reliable*.

(Methods for quantifying the accuracy, and perhaps using both 90^3 and 80^3 grid solutions to achieve accuracy higher than the 100^3 grid solution, will be addressed in Chapter 5.)

2.19 ADDITIONAL REMARKS ON VERIFICATION

Finally, I would like to address some concerns raised in the literature that are not as significant as claimed, in my opinion.

Cheng (1970) and later Oberkampf et al (1995) put much emphasis on the importance of the Lax Equivalence Theorem (see Richtmyer and Morton, 1967) as being of fundamental importance to CFD. I believe the importance is exaggerated. The Equivalence Theorem may be briefly condensed to stating that *consistency + stability yields convergence*. Consistency simply means that as $\Delta \rightarrow 0$, the discrete (algebraic) equations approach the continuum partial differential equations, which is just the calculus limit. Stability is defined in the von Neumann sense (Roache, 1998b). Convergence here means convergence of the algebraic solution to the continuum solution. Certainly this theorem is good to know, and provides some confidence. Unfortunately, it applies only to *linear* (and properly posed) initial value problems. As noted by Oberkampf et al (1995), the analyses for consistency and especially stability are "predominantly developed for very simple model problems... The model equations are always linearized equations and uncoupled from any other equations."..."Additional, but related, simplifications of consistency and stability analyses are elimination of: mixed classification partial differential equations, non-uniform grids, and boundary conditions." In the matter of mixed PDEs, they noted that "in *every* supersonic flow problem modeled by the steady Navier-Stokes equations, hyperbolic and elliptic regions exist adjacent to one another." So as a practical matter, modelers cannot sit on their hands waiting for more general theorems. Also, note that numerical boundary layer solutions were obtained, Verified, and Validated many years before the Equivalence Theorem was published, so progress can be made without the rigor. The same applies to questions of existence and uniqueness.

Another concern in Code Verification is the number of user options in a code, especially general-purpose commercial codes. This is a genuine practical problem, but does not nullify claims of Verification; it just limits those claims. The exponentially expanding complexity of the option tree does not nullify the definition of Verification of Code; it simply qualifies the definition. "Code Verification" is restricted to that combination of options claimed to be Verified. There is a gray area here, as one might expect, in the judgment of the independence of options. Some knowledge of algorithm and code structure may be necessary to infer the reasonableness of simplifications of option interactions (essentially, partitioning of the full matrix of option interactions). The easy answer is the rigid one of categorical "no." This would usually appeal to the rigid SJ temperament type (see Chapter 12) that gives Quality Assurance a bad name. A more intelligent and economical approach is possible, bearing in mind that subtle and unanticipated option interactions have occurred, especially before the acceptance of structured coding and modularity.

In regard to "canned" or highly modular elliptic solvers, one can make a very strong case for independence of the options. The argument becomes unassailable if separate residual checks are made after the canned solver has obtained the solution. Essentially, the "solver" is treated as providing an initial guess for a simple point-iteration method. Different modules for the solver could be interchanged (e.g., a multigrid solver, elliptic marching method, PCG, etc.) without repeating all option-tree combinations (e.g., types of boundary conditions, time differencing schemes, well descriptions) since the algebraic solution provided by the solver is always verified (Roache et al, 1996).

Validation	– Supported by an accepted authority, such as an archival benchmark experiment that provides specified requirements
Verification	– Implies that accuracy has been ensured, for instance with regard to coding
Calibration	– Implies that deviations from some standard, such as an archival benchmark experiment, have been determined
Certification	– Implies official recognition that a standard has been met

Table 2.21.1. Connotations Associated with Key CFD [computational PDEs] Terms.
(From Figure 2, Porter, 1996).

2.20 § DOES *MODEL* INCLUDE THE GRID?

The word *model* itself is used with different definitions and connotations. Attempts are made to sort out different definitions with terms like mathematical model, computational model, conceptual model, etc. (E.g. see V&V10.) Unfortunately, the existing literature is very far from consistent or clear, and one must read carefully. A simple question is, does *model* include the discretization, i.e. the grid or mesh on which the equations are solved, or does it just refer to the continuum equations being solved? The usual first impulse is to include the grid but it quickly leads to contradictions, since a grid convergence verification test then involves changing the "model." If one verifies the calculation by performing grid convergence testing, one concludes that the model is different for each grid, so what model calculation is being verified? In geophysical modeling (weather, climate, ocean, groundwater flow and transport) a particular mesh will have long-term use, measured in years. The models are run and re-run over many simulations, and it is common to use *model* to include the discretization, as in "a 1/5 degree model of the Gulf of Mexico." Yet this use by inference does not include initial conditions or forcing functions in *model* since these change with every run, whereas the use of *model* in a strong sense would include these. In other areas, *model* (unqualified) usually does not include the grid.

Other people will include additional numerical features within the concept of *model*, e.g. *discretization algorithms*. In extreme cases, people will include not only grid and discretization algorithms but also iteration tolerances and solvers. This makes *some* sense, in that these all affect the answers, but it leads to unwanted results. A change in iteration tolerance means that one has a new model, which requires new Validation, which requires new experiments in some rigid interpretations (see V&V10). Inclusion of *mesh* and *discretization algorithms* within *model* reduces Validation, when using solution adaptive remeshing and discretization, to semantic nonsense.[14] To avoid logical problems, the recommended practice is to define *model* in reference to what the code that embodies the model[15] will produce when (near) grid discretization independence is reached ($\Delta \to 0$). Clearly there is a need for balance, for avoiding sophistry and legalism, and for valuing semantic distinctions yet recognizing their limits.

[14] Some climate models include parametric sampling over algorithms. See McWilliams (2007), Rehmeyer (2007). Semantics aside, I cannot imagine anything physically meaningful resulting from sampling of algorithms.

[15] This can be a virtual code, i.e. a thought experiment. Note that position of far-field boundaries is part of *model* in this definition, since its error does not vanish as $\Delta \to 0$. See Section 6.10.

2.21 CONCLUSION: LIMITATIONS OF SEMANTIC DISTINCTIONS

Porter (1996) gave a useful list of connotations rather than definitions, given here in Table 2.21.1. The semantic distinctions involved in the general area of confidence building in computational PDEs are important and worthwhile. While sometimes arbitrary, it is worthwhile to try to maintain uniformity of terminology, or at least to recognize the underlying conceptual distinctions and to define one's terms with appropriate precision. The most important distinction, and fortunately the most generally agreed upon, is the distinction between Verification (mathematics) and Validation (science).

The definitions used herein are sometimes at variance with the discussions in Oberkampf (1994), Sindir et al (1996), Jameson and Martinelli (1996), V&V10, and others, notably in the present title subject of Verification. Our position is that we *Verify a Code* by convincingly demonstrating (if not proving as in a mathematical theorem) that the code *can* "solve the equations right." When done properly, the exercise also Verifies the *order of convergence* of the code. Then, we *Verify a Calculation* by convincingly demonstrating that the code *has* "solved the equations right," to a rationally estimated accuracy or error band. These two exercises are purely mathematical; neither appeals to experimental data for their justification. Only in *Validation* do we demonstrate that we have "solved the right equations," with an understood context of engineering or scientific accuracy, by appealing to experimental data.

It is well to recognize the limitations of our attempts at semantic distinction. Although these efforts are worthwhile, it is clear that we (the scientific–mathematical–engineering community) are *not* going to achieve uniform, non-overlapping terminology. For example, in the classification framework of Sindir et al (1996), their "Validation" procedure included what is herein and elsewhere defined as aspects of Verification, so that their "Validation" was more like a total code Quality Assurance program (see Chapter 11). Likewise, Bogdonoff (1993), Grasso et al (1994), Marvin (1995), Jameson and Martinelli (1996) referred to "Validation" of both numerical simulations and experiments, i.e., what we here define as Verification and Validation. Also, these authors often used the word "verification" in a non-specific sense when discussing Validation, i.e., they did not make the distinction in terms as used herein. Beyond the "mere" semantics, Oberkampf (1994) included comparisons of calculations with carefully performed experiments as part of Verification of a code (a definition he has since refined). In an extreme case of semantic confusion, the otherwise informative general-audience article titled "Verifying Analytical Methods" (Puttre, 1994) should have been titled "Validating Numerical Codes"!

Nevertheless, following the advice of Chuang Tzu (see quote at the beginning of this chapter), it is worthwhile to keep in mind the ideas *behind* Verification and Validation. Porter (1996) made the limitations of languages clear by using the word connotations instead of definitions for the key terms, as shown in Table 2.21.1.

Porter (1996) also quoted Alice from *Through the Looking Glass* (an English fantasy written, appropriately enough, by a mathematician, Lewis Carroll): "Don't listen to what I say, listen to what I mean." This wise advice agrees with the Taoist spirit of Chuang Tzu. In reading journal papers and reports, it is not a good idea to try to enforce terminology or taxonomies. It is better to try to *detect* the authors' terminology and/or taxonomies (sometimes defined only implicitly, and perhaps used with some forgivable inconsistency) and then go on to learn from the authors' experience and perspective.

Finally, techniques are already available to convincingly Verify the numerical accuracy of Codes and Calculations without impossible stress on modern computer resources, as will be demonstrated in the succeeding chapters. These techniques are applicable to commercial codes as well as specialized scientific codes.

PART II

VERIFICATIONS

Part II will cover procedures for the rigorous accuracy Verification of Codes and for Verification of individual calculations. These two activities are distinct. The code must be convincingly Verified before Verification of an individual calculation can proceed with any confidence. Verification of a calculation involves error *estimation* and/or banding, whereas Verification of a code involves error *evaluation* from a known solution. Included in the Verification of calculations will be a taxonomy for sources of additional information needed for the error estimation.

Since this Part II concerns accuracy Verification only, we will not be concerned herein with the accuracy of physical laws but only with mathematics. Part III will cover Validation, i.e., the agreement of the mathematics with science.

CHAPTER 3

METHOD OF MANUFACTURED SOLUTIONS: A METHODOLOGY FOR VERIFICATION OF CODES

3.1 INTRODUCTION

This chapter describes the Method of Manufactured Solutions, which is now an undeniably mature methodology and is widely accepted as the gold standard for Code Verification.

This chapter is based primarily on the methodology I presented in Steinberg and Roache (1985), "Symbolic Manipulation and Computational Fluid Dynamics" and expanded in Roache et al (1990). The methodology provides for convincing, rigorous Verification of the numerical accuracy of a code via systematic grid convergence testing. This procedure is straightforward though somewhat tedious to apply, and verifies all accuracy aspects of the code: formulation of the discrete equations (interior and boundary conditions) and their order of accuracy, the accuracy of the solution procedure, and the user instructions.

3.2 WARNINGS: THE DIVISION OF LABOR IN CODE DEVELOPMENT AND USE

As complete as the claim above may sound, a rigorous Verification of the numerical accuracy of a code does *not* by any means imply that all numerical questions about a particular *simulation* are finished. Far from it. To keep the distinction in mind, it is helpful to think of the *division of labor* or division of responsibility, with a code builder Verifying the code with no knowledge of the application. If this sounds strange, realize that there are a multitude of applications possible for a code. I want to Verify my code, without waiting to see how many applications will develop.

I build a code: what can I claim about it, and communicate to you the possible user, without knowing anything about your intended application? I can tell you what equations my code solves. I cannot tell you what equations you need to solve for your problem. I can tell you a theoretical order of convergence for the algorithms in my code, and the observed order of convergence of my code for one or more well-behaved problems, and what grid refinement level was sufficient to attain asymptotic performance on those well-behaved problems. I cannot tell you what grid will be required for your problem, and (unfortunately) I cannot guarantee that the observed order of convergence for your problem will be the same as that observed on my well-behaved problems. You as modeler and code user will have to decide if the equations that I solve are appropriate for your problem. You will have to decide whether my Verified order of convergence on well-behaved problems can be trusted for your problem, or whether you will Verify the rate of convergence on your problem. In either case, you cannot simply perform a single grid calculation, but rather you must Verify the calculation for your problem, which involves error estimation and/or banding, as discussed in Chapters 4 through 8.

We are concerned in this Chapter 3 with accuracy Verification of the *code* only. Strictly speaking, this will involve error *evaluation* rather than error *estimation*. However, the Verification of the *order of convergence* (or the *rate of convergence*) of the code involves the same definition of order that is at the basis of error estimation. Thus, there are many connections, but as usual, it will be useful to maintain the distinctions, this time between error *evaluation* and error *estimation*.

3.3 ORDER OF CONVERGENCE

The principle definition of "order of convergence" that we use here is based on behavior of the error of the discrete solution. There are various measures of discretization error, but in some sense we are always referring to the difference between the discrete solution $f(\Delta)$ (or a functional of the solution) and the exact (continuum) solution,

$$E = f(\Delta) - f^{exact} \tag{3.3.1}$$

For an order p method, and for a well-behaved problem (exceptions to be discussed in Chapters 6 and 8), the error E in the solution asymptotically will be proportional to Δ^p, where Δ is the grid spacing or other measure of the discretization. This terminology applies to all consistent methodologies including finite difference methods (FDM), finite volume methods (FVM), finite element methods (FEM), block spectral[16] and pseudo-spectral, vortex-in-cell, etc., regardless of solution smoothness. By "consistent" we mean, of course, that the continuum equations are recovered in the limit of infinite discretization. (Lattice gas methods may or may not qualify as consistent, and therefore are highly suspect.) Thus,

$$E = f(\Delta) - f^{exact} = C \cdot \Delta^p + H.O.T. \tag{3.3.2}$$

where H.O.T. are higher order terms. For smooth problems, it may be possible in principle to evaluate the coefficient C and the H.O.T. from the continuum solution, but as a practical matter, we do not do this in the accuracy Verification procedure.

[16] Global spectral methods converge exponentially fast, i.e. faster than any finite-ordered polynomial-converging discretization method,, so Eq. 3.3.1 does not apply. However, these methods still require an exact solution for Code Verification, and the MMS described in Section 3.4 ff is applicable.

3.4 Δ METHOD OF MANUFACTURED SOLUTIONS

The Method of Manufactured Solutions (MMS) is a general procedure for generating an analytical solution for code accuracy Verification. The well-known procedure for grid convergence testing described earlier in Chapter 1 is applicable only when an exact solution is known. In the anecdote cited there, a known exact solution for electrode potential fields was very useful in uncovering a previously unsuspected code bug. However, the procedure Verified the code only for those features exercised for the analytical solution. Thus, the code accuracy Verification was limited to the vacuum electrode case; the solution was for linear steady state only and constant properties, whereas the code also treated the more general problem.

It has often been stated in research journal articles that general accuracy Verification of codes for difficult problems, e.g. the full Navier-Stokes equations of fluid dynamics, is not possible because exact solutions exist only for relatively simple problems that do not fully exercise the code. Many papers and reports approach accuracy Verification of codes in a haphazard way, comparing single-grid results for a few exact solutions on problems of reduced complexity. In fact, a very general procedure exists, first described in detail in Steinberg and Roache (1985) for generating analytical solutions for code accuracy Verification.

The basic idea of the procedure is to simply *manufacture* an exact solution, without being concerned about its physical realism. The "realism" or lack thereof has nothing to do with the mathematics, and Verification is a *purely mathematical* exercise. In the original, most straightforward and most universally applicable version of the method , one simply includes in the code a general source term, $Q(x, y, z, t)$ and uses it to generate a non-trivial but known solution structure. We follow the counsel of G. Polya (1957):

Only a fool starts at the beginning; the wise one starts at the end.

We first pick a continuum *solution*. Interestingly enough, we can pick a solution virtually independent of the code or the hosted equations. That is, we can pick a solution, then use it to verify an incompressible Navier-Stokes code, a Darcy flow in porous media code, a heat conduction code, an electrode design code, a materials code,...("Pick a card - any card!")[17]

We want a solution that is non-trivial but analytic, and that exercises all ordered derivatives in the error expansion and all terms, e.g., cross-derivative terms. For example, chose a solution involving the hyperbolic tangent function *tanh*. This solution also defines boundary conditions, to be applied in any (all) forms, i.e., Dirichlet, Neuman, Robin, etc. *Then the solution is passed through the governing PDEs to give the production term* $Q(x, y, z, t)$ *that produces this solution.* In Steinberg and Roache (1985) we used Symbolic Manipulation to generate Q; the detailed results will be given shortly.

This procedure is much easier and more general than looking for solutions to real problems (see additional references in Roache et al, 1990.) We then monitor the numerical error as the grid is systematically refined. Successive grid *halving* is not required, just *refinement*. However, it is essential in the grid refinement procedure to refine with the same factor in all coordinate directions; otherwise, erroneous convergence rates p will be observed (Salas, 2006). Thorough iteration convergence is required; inadequate iteration convergence will corrupt the results.[18] Theoretically [from Eq. (3.3.2)], values of C =

[17] Considerable lecture experience by myself and colleagues has shown that the basic concept can be difficult to grasp and indeed can appear like a card trick or a shell game. The reader may prefer the more tutorial presentation (Roache, 2002) reproduced in Appendix C that presents the method using three simple 1-D problems.

[18] See Section 5.10.10 for more guidance on iteration convergence.

error/Δ^p should become constant as the grid is refined for a uniformly *p-th* order method ("uniformly" implying at all points for all derivatives).

When this systematic grid convergence test is Verified (for all point-by-point values), we have verified

1. any equation transformations used (e.g., nonorthogonal boundary fitted coordinates),
2. the order of the discretization,
3. the encoding of the discretization, and
4. the matrix solution procedure.

This technique was originally applied in Steinberg and Roache (1985) to long Fortran code produced by Artificial Intelligence (Symbol Manipulation) methods developed by Prof. Steinberg. (See also Roache and Steinberg, 1984; Steinberg and Roache, 1986.) The first versions of the code produced extremely long subroutines because the Symbol Manipulation code Macsyma at that time did not know the rules for intermediate expressions in the chain rule expansions and for the derivative of the inverse of a matrix function. (Steinberg later installed these rules in Macsyma; see Steinberg and Roache, 1985.) The original three-dimensional nonorthogonal coordinate code contained about 1800 lines of dense Fortran. It would be impossible to check this by reading the source code, yet the procedure described Verified the code convincingly. (Surprisingly, roundoff error was not a problem.)

The arbitrary solution, produced inversely by the specification of the source term Q, was aptly described by Oberkampf et al (1995) and Haynes et al (1996) as a "Manufactured Solution." The fact that the Manufactured Solution may bear no relation to any physical problem does not affect the rigor of the accuracy Verification. The only important point is that the solution (manufactured or otherwise) be non-trivial, i.e., that it exercise all the terms in the error expansion. (Ethier and Steinman, 1994, followed a similar philosophy and procedure.) The algebraic complexity may be something of a difficulty, but is not insurmountable, and in practice has been easily handled using Symbolic Manipulation packages like Macsyma, Mathematica, Maple, etc. Using the source-code (Fortran) writing capability of Macsyma, it is not even necessary for the analyst to look at the form of Q; rather, the specification of the solution (e.g., *tanh* function) to the Symbolic Manipulation code results in some possibly complicated analytical expression that can be directly converted by the Symbolic Manipulation code to a Fortran (or Pascal, C, etc.) source code segment, which is then readily emplaced in a source code module (subroutine, function, etc.) that then is called in the accuracy Verification procedure. (This emplacement can be performed by hand by the analyst, without actually reading the complicated source code expressions, or can itself be automated in the Symbolic Manipulation code.)

The procedure may also be applied to systems of equations, with separate Q's generated for each equation. Both unsteady and steady solutions are possible. (It may be useful to avoid exponential solution growth in time so as to avoid confusion with instabilities; e.g., see the fully three-dimensional incompressible Navier-Stokes analytical solutions of Ethier and Steinman, 1994.) Nonlinearity is an issue only because of uniqueness questions; otherwise, the source term complexity may be worse, but that is the job of the Symbolic Manipulation code. Non-uniqueness could be an issue because the code could converge to another legitimate solution other than the Manufactured Solution, producing a false-negative accuracy Verification test for a correct code. However, it would be difficult to contrive a situation in which a false positive accuracy Verification was obtained. In Steinberg and Roache (1985), we applied the procedure to the nonlinear (quasi-linear) PDEs of the elliptic grid generation method for non-orthogonal coordinates (Thompson et al, 1974). The Manufactured Solution here was an analytical three-dimensional coordinate transformation; see examples below.

The only disadvantage of the procedure is the requirement that the code being Verified must include accurate treatment of a source term and that its boundary condition values not be hard-wired. Many codes

are built with source terms included, and many algorithms allow trivial extension to include Q's. However, in directionally-split algorithms such as Approximate Factorization (Briley and McDonald, 1977; Beam and Warming, 1976) the time-accurate treatment of $Q(x, y, z, t)$ involves subtleties and complexities at boundaries, especially for non-orthogonal coordinates (e.g. see Salari and Roache, 1990). Thus, significant code extensions may be required in order to apply this procedure involving Manufactured Solutions for code accuracy Verification. Likewise, some old groundwater flow codes were built with hard-wired homogeneous Neumann boundary conditions, $\partial f / \partial n = 0$. In order to use an arbitrary solution function, non-homogeneous boundary values like, $\partial f / \partial n = g$ would be required. (The Q could be modified to set g.) Alternately, one could circumvent the problem by restricting the choice of Manufactured Solution functions to fit the hard-wired values. Likewise, to test periodic boundary conditions, one must chose a periodic function for the MS. It is also possible to choose the MS to meet other constraints of the system, e.g. incompressibility, but this is not always necessary.

Note that it is not necessary to develop different Manufactured Solutions to test different boundary condition *types*. This has been often misunderstood, even by proponents of MMS. The same MS can be evaluated at boundaries to produce function values to test Dirichlet boundary condition options, normal gradient values to test Neumann boundary condition options, and both for mixed (Robin) boundary condition options, with an arbitrary non-zero weighting function β in

$$f + \beta \cdot \frac{\partial f}{\partial n} = g \tag{3.4.1}$$

Similarly for angle conditions encountered in linearized potential flow and magnetohydrodynamics,

$$\frac{\partial f}{\partial x} = g \frac{\partial f}{\partial y} \tag{3.4.2}$$

Note that the method will Verify convergence rate of the solution, but will not Verify that a code is algebraically conservative (i.e., "strongly conservative" so that mass is identically conserved at all grid resolutions).

3.5 EXAMPLE: 3-D POISSON EQUATION AND NONORTHOGONAL 3-D GRID GENERATION

The following examples are taken almost verbatim[19] from my original work in Steinberg and Roache (1985), pp. 277–283, with some repetition of material presented earlier in this chapter. The codes being tested had been generated by Prof. Steinberg using an early VAX computer version of the Symbolic Manipulation code MACSYMA.

3.5.1 Verification of Code Generated by Symbolic Manipulation

The potential for errors in either the problem formulation or the encoding procedure always exists in complex codes such as three dimensional boundary fitted coordinate codes. The need for Code Verification was emphasized in the present work because Symbolic Manipulation was used to generate the codes (e.g.,

[19] One required departure from "verbatim" is due to the embarrassing fact that I used "validation" instead of "verification" in 1985.

see Roache and Steinberg, 1984, Steinberg and Roache, 1985, 1986a,b). The resulting "psychological distance" from the work made it less likely to be satisfied with superficial plausibility exercises based on intuitive ideas of acceptable levels of absolute error.

We Verify the codes by performing numerical tests. The approach is to choose a (manufactured) continuum solution for the class of problems treated by the code, and to Verify the convergence of the results to this continuum solution by systematic discretization error convergence testing over a sequence of grid sizes.

3.5.2 Hosted Equation Convergence Testing Method

To test the equation $L(F) = Q$, we tested a modified equation whose manufactured solution was chosen[20] to be

$$F_m = (x_1^{m+1} x_2^{m+2} x_3^{m+3}) / F_{mn} \quad on\ R \tag{3.5.2.1}$$

where

$$m = nl + na - 1 \tag{3.5.2.2}$$

and nl = degree of L (e.g., nl = 2 for L = Laplacian) and na = order of accuracy of the finite-difference expression (e.g., na = 2 for centered differences). F_{mn} is the normalizing value for the manufactured solution F_m. The motivation in the selected form of the solution is of course to insure that all the derivative terms in L are exercised, and that there is non-zero truncation error for finite h_i, even without the transformation of coordinates. (For example, a parabolic solution will show no truncation error using 2nd-order accurate solutions with the identity transformation. Several published "Verifications" of the accuracy of upwind differencing are inadequate because the chosen solution structure does not exercise the meaningful terms in the truncation error, giving a false indication of accuracy.)

The modified problem to be tested was then

$$L(F) = Qs \quad on\ R \tag{3.5.2.3}$$

where

$$Qs = Q + L(F_m). \tag{3.5.2.4}$$

The boundary equations for the modified problem are

$$F = F_m \quad \text{on boundary of} \quad R. \tag{3.5.2.5}$$

Qs can be obtained in x by elementary operations as

$$Qs = Q + F_m \left[\frac{m(m+1)}{x_1^2} + \frac{(m+1)(m+2)}{x_2^2} + \frac{(m+2)(m+3)}{x_3^2} \right]. \tag{3.5.2.6}$$

[20] It is usually best to generate the manufactured solution in original ("physical space") coordinates (x,y,z,t). Then the same solution can be used directly with various non-orthogonal grids or coordinate transformations.

The solution was obtained in a stretched coordinate system $\xi = (\xi_1, \xi_2, \xi_3)$ where the ξ_i values are linear from 0 to 1, $\xi_1 = h1 * (i - 1)$, etc. The chosen coordinate transformation was given by

$$x_i = \xi_s + \xi_i + tanh(d_i \xi_1 \xi_2 \xi_3) \tag{3.5.2.7}$$

ξ_s is a shift, necessary to avoid a singularity in the chosen solution at the origin. The parameters d_i control the severity of the coordinate stretching; for $d_i = 0$, there is no stretch in x_i direction. The form of the coordinate transformation is chosen to exercise all terms in the transformed equations, notably the cross-derivative terms, which would be zero if the coordinate transformations were independent in each direction. For non-zero d_i the *tanh* term in the transformation assures non-zero values for all derivatives.

The above solution and transformation generate non-trivial (non-zero and non-repeating or canceling) derivatives of all orders, including cross-derivatives.

The discrete solution F_d is then obtained on a family of grids, using any standard method. We used the spatial marching methods in the GEM codes (Roache, 1995) for 2-D, and hopscotch SOR in 3-D. It is important that iteration convergence criteria be stringent, so as not to confuse the incomplete iteration (residual) error with the discretization error $te = F_m - F_d$. We required iteration convergence to essentially the single precision accuracy of the 32-bit VAX computer used, about 7–8 decimal figures. We also forced a minimum number of iterations to be performed, somewhat greater than the maximum grid index, to be sure that a false indicator of iteration convergence was not obtained due to good initial conditions available at the fine grid spacing.

The discretization error convergence is then monitored as the grid is systematically refined. Theoretically, the local values of $p = te/h^2$ should become constant as the grid size is refined. For a uniformly *na*-th order finite difference discretization ("uniformly" implying at all points for all derivatives) the truncation error can be written as

$$te = C_1 h_1^{na} + C_2 h_2^{na} + C_3 h_3^{na} + \text{H.O.T.} \tag{3.5.2.8}$$

We define the index I of predicted order of discretization error

$$I = N^{na}. \tag{3.5.2.9}$$

With $h_1 = \xi_{1\max} / N = 1/N$, etc. we have

$$I = C_1 \xi_{1\max}^{na} + C_2 \xi_{1\max}^{na} + C_3 \xi_{1\max}^{na} + \text{H.O.T.} \tag{3.5.2.10}$$

which is constant to order *na* approximation.

We then monitor I as N increases, using I at specified points (e.g., *Icr* at the center of the grid) and *Imx* = $\max_{i,j,k} I$. If all I = *constant* as N increases, we have Verified (1) the equation transformation, (2) the order of finite difference expressions, and (incidentally) (3) the solution procedure.

This procedure does not require a grid halving sequence, $N = 2^m$, although this sequence does make the judgment easier. Also, we note from experience, even on trivial 1-D problems, that I = "constant" as the grid is halved can be significantly relaxed; a 5% variation is very safe. (The results presented herein are more convincing, as will be seen.)

For a coarse-mesh solution, it would be advantageous to eliminate the iteration residual from the calculation of *te* by using a direct method, i.e., Gaussian elimination. However, in 3-D this will fail for large N by round-off error and/or CPU time demands; likewise for the marching methods. Iterative methods

are dangerous in that their convergence depends on L, and on the coordinate transformation. Note, however, that it is not necessary, nor even advisable, to test the small-parameter (high Reynolds number) problem. Errors will show up more readily when all coefficients in L are of the same order. Once the above procedure has Verified the algebra and coding accuracy, one can proceed with alternate solutions techniques for the small-parameter problem.

Note that this procedure can be followed for mixed-order finite difference equations (e.g., upwind differencing on first derivative terms and 2nd-order differencing on other terms) with the modification that we look only for convergence in the sense of zero error as N increases. This modified procedure will *not* Verify the mixed orders of accuracy separately. Ultimately the convergence rate will be limited by the lowest order discretization used, i.e., 1st-order, which will be Verified by the procedure. However the procedure will not Verify that the non-advection terms (e.g. boundary conditions intended to be 2nd-order) were correctly discretized to 2nd-order. Also, mixed-order codes will probably require finer grids since the convergence of these is difficult to judge.

3.5.3 Hosted Equation Convergence Results in 3-D

Table 3.5.3.1 presents the results of testing for the constant-coefficient Laplacian equation in 3-D, using moderate stretching parameters $d_i = 0.1$. The mesh size was successively halved three times, from 5^3 to 3^{33}. The numerically calculated Jacobian for the 5^3 grid ranged from 1.0187 to 1.1684, and for the 3^{33} grid ranged from 1.0003 to 1.2792. The table presents the maximum discretization error over the grid, *temax*, and the grid location at which it occurred; the discretization error at the center of the grid, *tectr*; and the corresponding values *Imx* = *temax*/h^2 and *Ier* = *tectr*/h^2.

grid	*temax*	*at*	*tectr*	*Imx*	*Ier*
5^3	0.402E–03	4,4,4	0.832E–05	0.646E–02	0.133E–03
9^3	0.172E–03	8,8,8	0.650E–06	0.111E–01	0.416E–04
17^3	0.492E–04	15,15,15	0.340E–07	0.126E–01	0.870E–05
33^3	0.134E–04	29,29,29	–0.169E–06	0.137E–01	–0.173E–03

Table 3.5.3.1. 3-D Hosted Equation Convergence Testing for $\nabla^2 F = 0$.
Stretching parameter $d_i = 0.1$. (From Table I, Steinberg and Roache, 1985.)

The near-constancy of the coefficient *Imx* = *temax*/h^2 indicates that the entire solution, including the coordinate transformation and the finite difference expressions, is 2nd-order accurate. The erratic behavior of *Icr* at the finest grid indicates round-off error problems with the short word-length computer used.

Table 3.5.3.2 presents results of testing for the variable-coefficient Laplacian equation in 3-D, using both moderate stretching $d_i = 0.1$ and strong stretching $d_i = 10$. The hosted equation solved was

$$\nabla\sigma \cdot \nabla F = 0 \tag{3.5.3.1}$$

$$\sigma = \sigma_0 + \sigma_m \sin(b) \tag{3.5.3.2}$$

$$b = x_1^n x_2^n x_3^n \tag{3.5.3.3}$$

	grid =	5^3	9^3	17^3	33^3
di = 0.1	*Imx* =	0.390E–02	0.877E–02	l.00E–02	–
di= 10.0	*Imx* =	.433	.691	1.21	1.53

Table 3.5.3.2 3-D Hosted Equation Convergence Testing for $\nabla\cdot\sigma\nabla F = 0$.
(From Table II, Steinberg and Roache, 1985.)

$$x_{1n} = \frac{x_1 - x_{1\min}}{x_{1\max} - x_{1\min}}, \text{ etc.} \tag{3.5.3.4}$$

The code, of necessity, used the expanded (non-conservation form),

$$\sigma\nabla^2 F + \nabla\sigma\cdot\nabla F = 0. \tag{3.5.3.5}$$

(In 1985, Prof. Steinberg had not yet developed symbolic manipulation codes with the ability to retain the unexpanded or conservation form of the equation.)

The manufactured solution chosen was

$$F_m = \frac{1}{F_{mn}}(x_1^{m+1}x_2^{m+2}x_3^{m+3}) \tag{3.5.3.6}$$

giving the forcing term

$$\begin{aligned} Qs = \frac{1}{F_{mn}}[&\sigma m(m+1)x_1^{m-1}x_2^{m+2}x_3^{m+3} + \sigma(m+1)(m+2)x_1^{m+1}x_2^{m}x_3^{m+3} \\ &+ \sigma(m+2)(m+3)x_1^{m+1}x_2^{m+2}x_3^{m+1} + \sigma_x(m+1)x_1^{m}x_2^{m+2}x_3^{m+3} \\ &+ \sigma_y(m+2)x_1^{m+1}x_2^{m+1}x_3^{m+3} + \sigma_z(m+3)x_1^{m+1}x_2^{m+2}x_3^{m+2} \end{aligned} \tag{3.5.3.7}$$

with

$$\sigma_x = \frac{\sigma_m \cos(b)x_2^n x_3^n}{bd} \tag{3.5.3.8}$$

$$\sigma_y = \frac{\sigma_m \cos(b)x_1^n x_3^n}{bd} \tag{3.5.3.9}$$

$$\sigma_z = \frac{\sigma_m \cos(b)x_1^n x_2^n}{bd} \tag{3.5.3.10}$$

$$bd = (x_{1\max} - x_{1\min})(x_{2\max} - x_{2\min})(x_{3\max} - x_{3\min}) \tag{3.5.3.11}$$

The results for a Verification test using $\sigma_m / \sigma_0 = 0.1$ are shown in Table 3.5.3.2. The results for *Imx* show that the Code Verification occurs even at the grid 17^3 for moderate stretching with $d_i = 0.1$, but that there is some variation in *Imx* even at the grid 33^3 for strong stretching with $d_i = 10$. The convergence is clear, however.

3.5.4 Hosted Equation Convergence Results for Strong Stretching

The behavior of the discretization error for the strong stretching is of interest. The values of *temax* for just the coarse grid are plotted in Table 3.5.4.1 for a range of stretching parameters d_i. For this coarse grid, $d_i = 0.1$ minimizes the maximum discretization error. The table shows that large and inappropriate stretching values do indeed increase the discretization error by almost two orders of magnitude compared to the best transformation used. However, the table shows that the results are still $O(h^2)$ accurate. By definition, we mean that the method is 2nd-order accurate if a reduction in h by a factor of 1/2 will (asymptotically) reduce the discretization error by a factor of 1/4.

dj =	0.	0.1	0.5	1.0	10.0
temax	0.830E–03	0.244E–03	0.489E–02	0.737E–02	0.271E–01

Table 3.5.4.1 3-D Hosted Equation Convergence Testing for $\nabla\cdot\sigma\nabla F = 0$ in Coarse Grid = 5^3.
(From Table III, Steinberg and Roache, 1985.)

We also Verified this retention of 2nd-order accuracy for *very* strong 1-D coordinate stretching. Using double precision calculations, we experimented with a range of stretching parameters for reasonable coordinate transformations based on hyperbolic tangent and exponential functions, and even for *un*reasonable transformations based on exponentials of exponentials. In the extreme case, we used a transformation of

$$x = \xi + \frac{exp(d1\times(f(x_i)-1)) - exp(-d1)}{1 - exp(d1)} \tag{3.5.4.1}$$

where

$$f(x_i) = \frac{exp(d1\times(x_i-1)) - exp(-d1)}{1 - exp(d1)} \tag{3.5.4.2}$$

with $d1$ ranging from 0 to 100. For $d1 = 10$, *Imx* = *temax/n*∗∗2 reaches its asymptotic value of 225 to two significant figures at $n = 257$. For $d1 = 100$, the truncation error is 6 orders of magnitude higher than the no transformation case; nevertheless, *Imx* is constant to at least the first figure at $n = 8193$ and 16385.

These results are at variance with analyses which focus on the coefficients of the Taylor series terms in the original ("physical") independent variables. The present results also Verify the claim (Roache, 1972b, 1998b) that an analytical transformation of the equations does not change the order of the accuracy. If the coefficients of the transformation are also evaluated to $O(h^2)$ accuracy, as well as the hosted equation, then the overall results will be $O(h^2)$ accurate.

3.5.5 Grid Generation Results in 3-D

A similar procedure was followed to Verify the grid generation codes. The grid generation method (Thompson et al, 1974) involves solving a set of three elliptic equations for the new coordinates $\xi(\mathbf{x})$, written in the original ("physical") coordinates as

$$L(\xi_i) = P_i \tag{3.5.5.1}$$

where L is the Laplacian operator in $\mathbf{x}$, and the P_i may be chosen to give some control of grid position in the interior. Since the system is transformed to ξ space, resulting in a set of three coupled nonlinear (quasilinear) equations, the Code Verification procedure is more complicated. We chose the manufactured solution grid to be the transformation

$$\xi_i x_i + tanh(d_i x_1 x_2 x_3) \tag{3.5.5.2}$$

which is the obverse of the transformation used for the hosted equation testing above. (Note: when discrete grid generation equations are taken to the limit of $\Delta \to 0$, the result is a "continuum grid," i.e. a transformation or parameterization.) Since the ξ_i are known at each grid point, being selected as just linear variables in the grid indexes, the above equation must be solved for the inverse values of x_1, x_2, x_3 at each grid point. This nonlinear 3×3 system was solved by coupled 3×3 Newton-Raphson iteration at each grid point to establish the continuum grid solution. Again, tight convergence criteria were used, assuring accuracy to essentially the single precision of the machine.

The P_i necessary to produce this transformation are then obtained by operating on the solution ξ with the Laplacian L, giving

$$P_i = -2d_i^2\left[x_2^2(x_3^2 + x_1^2) + x_1^2 x_3^2\right]sech^2(d_i x_1 x_2 x_3)tanh(d_i x_1 x_2 x_3) \tag{3.5.5.3}$$

The numerical solution was obtained using hopscotch SOR to solve linearized equations sequentially for x_1, x_2, and x_3, followed by Picard outer iterations to update the linearized coefficients. A relaxation factor of 1/2 was used in the outer (nonlinear) iterations, and loose iteration convergence criteria (actually, a limit on maximum number of inner iterations) were used in the early nonlinear stages. Again, overall iteration convergence was tight, essentially to the single precision of the machine, to clarify the discretization error behavior. The cost of solving the nonlinear 3×3 system for the grid test was more expensive than solving the scalar linear equation for the hosted equation test, but was still reasonably obtained because the exact continuum solution was available for use as initial conditions. However, this is no longer the recommended procedure, since Knupp and Salari (2003) have pointed out that it can result in false positive verification in subtle cases (coding errors resulting in failure to update).

The result for an early test in two grids is given in Table 3.5.5.1 for strong stretching values $d_i = 3$. The values of *temax* seem small, and in fact decreased by almost a factor of 2 with the mesh size halving. This test could likely pass for a Code Verification. The test was actually inconclusive and, in fact, the test driver code was in error.

The false result is included here to emphasize that systematic convergence testing over a *sequence* of grids is required for Code Verification. The error was in fact discovered with such systematic testing, the results of which are shown in Table 3.5.5.2, for both the original and corrected test drivers.

The complete, systematic convergence test in Table 3.5.5.2 clearly indicate a persistent error in the original test, and clearly indicate 2nd-order convergence in the corrected test as *Imx* becomes virtually constant.

3.5.6 Discussion of the Code Verification Procedure Using Manufactured Solutions

We note that the Code Verification procedure can be complicated by the use of 1st-order differencing, since this slows convergence and makes it difficult to judge. Also, the use of mixed 1st- and 2nd-order differencing (boundary conditions, advection terms at high Reynolds numbers, etc.) and/or any sort of conditional differencing (e.g., upstream differencing) will complicate the Code Verification procedure. The technique is applicable to Verifying the hosted equation codes as long as the method of grid generation can systematically refine the grid, which is the case for elliptic grid generating systems themselves. Particularly, the technique is applicable to Verifying code for other grid generation techniques that provide control over the grid properties of smoothness and orthogonality (Knupp and Steinberg, 1993).

The cost of computer time to do the systematic convergence testing can be significant, especially in 3-D problems. However, it should be noted that the kind of Code Verification described here does not address the small parameter (high Reynolds number) problem, so the costs and difficulties associated with it do not contribute to the expense. In fact, in designing the test problems, one specifically avoids large or small parameters, since these could mask errors in the treatment of the negligible terms. Thus, the computer CPU time to perform these Code Verification computations in the finest grid will be much less than that required to obtain a realistic (unknown) problem solution in that grid, when iterative solution methods are used.

grid =	5^3	9^3
temax =	0.394E–02	0.205E–02

Table 3.5.5.1 Grid Generation Tests, $d_i = 3$, with an Error in Coding in the Test Driver. (From Table IV, Steinberg and Roache, 1985.)

	grid =	5^3	9^3	17^3	33^3
original	*temax* ∗ 100=	0.394	0.205	0.169	0.160
corrected	*temax* ∗ 100 =	0.304	0.0821	0.0208	0.00533
original	*Imx* =	0.0630	0.131	0.431	1.646
corrected	*Imx* =	0.0486	0.0525	0.0533	0.0545

Table 3.5.5.2 Complete Grid Generation Tests, $d_i = 3$. (From Table V, Steinberg and Roache, 1985.)

The results obtained confirm that strong and inappropriate coordinate stretching can indeed increase the size of the discretization error, as is well known, but that the asymptotic order of the accuracy, as indicated by the reduction in discretization error resulting from systematic reduction in grid size, remains $O(\Delta^2)$ if centered differences are used in the transformed equations. Thus, the Method of Manufactured Solutions is not disabled by strong coordinate stretching.

3.5.7 Debugging with Manufactured Solutions

As described above, the Code Verification procedure simply gives a "yes" or "no" answer to the accuracy of the entire code: the transformation of the hosted equation(s), the substitution of the finite difference forms, the ensuing algebraic grouping, the Fortran encoding, the solution procedure for the discretized equations, and the correct formulation of the test problem. If the Code Verification procedure produces a negative result, it is conceivable that the only information obtained is that there is an error, *somewhere*. For example, an error in the formulation could destroy the diagonal dominance of the matrix equation, resulting in an unstable or extremely slowly converging numerical procedure.

In actual experience, the procedure has been readily modified, on an *ad hoc* basis, and has successfully helped in isolating and identifying coding mistakes. By judiciously building up or modifying the test problem, one can selectively turn off the cross-derivative terms, perform the trivial identity transformation, etc. Direct solvers for the hosted equation testing may also be used if iteration convergence is not attainable. We have found that printing and inspection of all values of the stencil at one internal grid point aided debugging and Code Verification. Likewise, Ethier and Steinman (1994) also found the use of their manufactured incompressible solutions to be very helpful for debugging purposes by analytically computing individual terms in the Navier-Stokes equations and comparing them with their numerical counterparts.

3.6 ANOTHER PATH TO MANUFACTURED SOLUTIONS

Another path to generating Manufactured Solutions is to chose a functional form of variable coefficients in the PDEs instead of a source term $Q(x, y, z, t)$ to generate a non-trivial solution. This approach may not be as general as that utilizing $Q(x, y, z, t)$, but it may also produce more realistic-looking solutions. Although this is not a mathematical advantage, it may provide a political advantage and inspire more confidence, depending on the mathematical sophistication of the audience (e.g., regulatory agencies, code QA committees, etc.).

Further examples of Accuracy Verification of Codes will be given in Chapter 6.

3.7 CODE VERIFICATION INCLUDING SHOCK WAVES

Without special considerations, the Method of Manufactured Solutions can be used for supersonic flow codes, or other PDEs in the smooth solution regime, but treatments for shocks (e.g., TVD, FCT, explicit dissipation methods) and other shock-like near-discontinuities may not be fully exercised. It appears to be possible to extend the method to include arbitrary shocks by including jumps in Q, but to my knowledge this has not been accomplished. A straightforward application is simply to pick a solution with sufficiently steep gradients/curvatures to activate the shock algorithm at coarse grids, but the behavior of grid convergence may be difficult to judge.

However, a very similar approach, independently developed by J. M. Powers and colleagues (Powers and Stewart, 1992; Powers and Gonthier, 1992; Grismer and Powers, 1992, 1996), has been applied to detonation shocked flows thoroughly and successfully, although special considerations apply.

The simplest multidimensional shocked solution for accuracy Verification of codes is obtained by simply translating a one-dimensional normal shock solution tangentially to obtain the oblique shock solutions. For perfect gas dynamics, this solution exercises the shock-capturing algorithm, but little else, since the flow before and after the shock is uniform. One could then use a non-shocked flow to exercise and Verify treatment of other terms, and could perhaps claim code accuracy Verification. I believe this is the simplest and effective way to Verify supersonic flow codes in a two-step procedure: (1) verify the coding for M < 1 everywhere with MMS as described earlier, and (2) verify the shock capturing (or other shock treatment algorithms) with inviscid benchmark problems solved by other methods (e.g. Taylor-Macoll solutions for supersonic sharp cones).

However, it is also value to have a single exact solution that exercises shocks and non-uniform flow in the same problem, and this has been accomplished. Grismer and Powers (1996) obtain more structure in their solution to a detonation shock problem, since the solution is not simply uniform before and after the shock. In that problem, the one-dimensional solution is obtained numerically by integration of ordinary differential equations (ODEs). Since ODE solvers are so highly developed and reliable, this is considered to be an "exact solution," in the same way that the Blasius boundary layer solution, although numerical, has been considered to be a primitive or closed-form solution in fluid dynamics. (For example, if a solution to a compressible flow problem can be expressed in terms of the incompressible Blasius solution, it is considered to be a closed-form analytical solution. After all, solutions in terms of exponential and trigonometric functions also require numerical evaluations; these are closed form solutions, though not algebraic solutions.) This solution, translated parallel to the shock, gives a two-dimensional solution with shock structure before and after the shock. However, cross-derivative terms are identically zero.

A two-dimensional detonation solution with more structure in the shock-free regions, but still containing a straight shock wave, was obtained by Powers and Gonthier (1992). Beginning with a straight shock, they solved backwards for the final shape of a wall that would generate the straight shock. This solution structure is more general and does contain cross derivatives in the non-shock regions. However, the shock itself is straight; curved shocks generate entropy-vorticity gradients downstream, and are a more demanding exercise of shock-capturing and shock-fitting algorithms.

The additional generality of weakly curved shocks is included in the asymptotic solutions of premixed reactive ideal-gas oblique detonations developed by Powers and Stewart (1992) and used as a numerical benchmark by Grismer and Powers (1992). Although strictly applicable only in the asymptotic limits of weak shock curvature and high free-stream Mach number M [the perturbation parameter is $\varepsilon = 1/M^2$ and the asymptotic error is $O(\varepsilon^2)$], they provided a practical, reliable benchmark solution for attainable M. The solution for curved shocks and straight walls is *not* restricted to small wedge angles. (They also presented an elegant solution for the somewhat simpler problem of straight shocks and curved walls.) The solution method follows the spirit of the pioneering work of Van Dyke (1958) on the supersonic blunt body problem. The solutions for the pressure and velocity fields are first parameterized by the shock position, and then boundary conditions at the surface of a wedge are chosen to fix the shock position. The solution procedure is non-trivial and the answer involves infinite series, but the work already has been done, and the series converges absolutely and uniformly. The solution provides a good benchmark problem for curved shock flows with reactions. (The flow may properly be described as both a shock-induced combustion and an oblique detonation.)

Any remaining doubts about the applicability of an asymptotic solution at a finite M could also be addressed systematically, by obtaining code numerical solutions for a sequence of increasing M and extrapolating to the high M limit. Note again the important distinction between Verification and Validation in regard to the high M limit. The facts that the user may be interested in lower M, and that ideal gas assumptions are not *physically* valid at $M = 20$, or that the solutions may not be physically stable, are irrelevant to the issue of Verification of the code, just as we noted previously that the requirement for non-

physically-realizable source terms in the Method of Manufactured Solutions is irrelevant. Verification is a *purely mathematical* exercise, so questions like applicability of perfect gas assumptions should be postponed until Validation exercises, which will presumably be performed at more reasonable *M* and will demarcate the limits of applicability.

If complete generality is deemed necessary, with curved shocks but without the asymptotic limitation on the solution, one could pursue the generation of a Manufactured Solution with general source terms, as noted earlier. But the work of Powers and colleagues provides reliable benchmark solutions that allow convincing accuracy Verification of codes including the phenomena of shock waves.

3.8 NEED FOR A THEOREM

There seems to be no chance for a sweeping theorem proving correctness of computational PDE codes by this Method of Manufactured Solutions in any general sense. However, like any analysis, it seems that there could be a useful theorem for a properly defined and limited scope. For codes like the one described above in Section 3.5, treating only the well behaved Poisson equation in general nonorthogonal coordinates, the exercise is compelling. I claim that this technique applied to such PDE codes (which class of codes I unfortunately cannot define with sufficient mathematical theorem-like precision) is correct, i.e., the numerical accuracy of the code is Verified, beyond a reasonable doubt.

3.9 SPECIFIC ANALYTICAL SOLUTIONS

Manufactured solutions often have an advantage over other analytical solutions because the boundary conditions are often more general. It is not necessary, from the purely mathematical considerations, that the flows be realistic or physically realizable. However, as noted, realistic looking analytical solutions have a psychological and political advantage. Fortunately, there are many manufactured or otherwise contrived analytical solutions in the literature. Also, as noted above in Section 3.7, approximate but highly accurate solutions (often obtained by perturbation methods) can also be utilized in Code Verification. Each discipline will have its own literature, which I certainly cannot cover, but the following list, though far from complete, may be useful to some workers. (See also Chapter 7 of Roache, 1998b.)

- Rogowski electrode, as used in Roache et al (1984); see Lorrain and Corson (1962).
- Darcy flow in 2-D, variable properties, as used in Roache et al (1990); see Chapter 6, Section 6.2.2, Darcy Flow with Tensor Conductivity in Non-Orthogonal Coordinates.
- Particle tracking in 2-D and 3-D, as used in Roache et al (1990); see Chapter 8, Section 8.6.
- Colloid Transport in saturated fractures; see Abdel-Salam and Chrysikopoulos (1994).
- Driven Cavity Flow. Huang and Li (1997) presented two simple analytic solutions for streamline flows that are suggestive of their problem of classical incompressible flow in a driven cavity, including a secondary (recirculation) vortex. They noted that one of their methods with apparently 1st-order boundary conditions produced 2nd-order performance on one of the test solutions, but correctly explain it as a "coincidence" [or artifact] of the accident that the solution produced homogeneous boundary conditions. As noted earlier, it is important for the manufactured (or other analytic) solution to exercise all the terms in the equations, in order to avoid misleading and over-optimistic results.
- Nonlinear Free-Surface Boundary Condition. Serrano (1995) obtained new analytical solutions of the nonlinear Boussinesq flow equation and of the exact two-dimensional groundwater flow equation subject to a nonlinear free-surface boundary condition. The generality of the solution allowed testing (or "Justification" as in Chapter 2) of the simpler linearized equation with the Dupuit assumptions,

showing that these commonly used approximations produce discrepancies in the presence of high regional hydraulic gradients, unusually high recharge rates, or regions of low conductivity.

- Analytic Elements. The "analytic element modeling" of Haitjema (1995) is a method of producing simulations that involve analytical solutions applied element-wise, and could also be used for exact solutions in accuracy Verification exercises.
- For fluid dynamics solutions, see also the compilation of exact solutions of the steady-state Navier-Stokes equations by Wang (1991) and the use of analytical solutions by Grenda et al (1996) for Verifying (in the present terminology) unsteady CFD codes.

3.10 Δ MANUFACTURED SOLUTIONS VS. OTHER NUMERICAL BENCHMARKS

As discussed in Section 2.5, numerical benchmarks and inter-code comparisons can be useful if these are high quality solutions, but will be misleading otherwise. If the goal is not mere confidence building but convincing code verification as described here for MMS, the demands on quality are even higher. Following Oberkampf and Trucano (2007, 2008) we consider infinite series solutions, numerical solutions of ODEs (ordinary differential equations, i.e. 1-D) and numerical solutions of PDEs (partial differential equations, i.e. multi-dimensional).

3.10.1 Infinite Series Solutions

Infinite series solutions are traditionally described as "exact" but in reality require numerical evaluation. Sudicky and Frind (1982) presented analytical solutions for contaminant transport in a system of parallel fractures in porous media. Their solution is very well known and widely used for benchmarking groundwater transport codes. What is not so well known is that the general solution as originally presented is incorrect, in that it appears to be given for more general boundary conditions but in fact it only satisfies homogeneous boundary conditions, as pointed out by Davies and Johnston (1984). They also noted that the infinite series solution involves a summation that is only conditionally convergent, and presented a more strongly convergent (robust) summation. Van Gulick's (1994) experience was that several other numerical problems exist in the Sudicky - Frind solution, e.g. the series solutions for high diffusivity or early time converge slowly or not at all.

The situation with the Sudicky–Frind solution is not unusual. Even if correct mathematically, any such analytical solution that involves series needs to be evaluated numerically, and subtle problems often arise. Also, it is difficult and not confidence-inspiring to confirm the mathematical correctness of the derivation of a non-trivial analytical or infinite series solution. In my fairly extensive experience with colleagues in compressible and incompressible aerodynamics, heat transfer, groundwater flow, groundwater contaminant transport, and laser electrode design, I can say categorically that the numerical evaluation of these "analytical solutions" *invariably* has involved more frustration than the computational codes being Verified. In this regard, as well as testing of variable properties and in generality of initial and boundary conditions, Manufactured Solutions (which do not involve infinite series) have an advantage over these analytical solutions.

3.10.2 § ODE and PDE Solutions

ODE solutions usually involve reduced physics, so that comparison of a multi-dimensional code must be done carefully to avoid false-negative Code Verification. For example, the well known Blasius boundary layer numerical solution (an interesting nonlinear two-point boundary value problem with one boundary at infinite distance) is close to the solution for the full Navier-Stokes equations at high Re, but if a full

Navier-Stokes code is compared, the small discrepancy is zero-order and will therefore appear in a grid convergence test as would a coding error. Other examples do not necessarily have this problem. For example, the Taylor-Macoll solution for supersonic inviscid flow over a sharp slender cone (another interesting nonlinear two-point boundary problem with one boundary condition being compatibility with oblique shock wave solutions) is 1-D in polar coordinate θ but could be used to verify a code using other coordinates (Cartesian or boundary fitted) in which the problem is 2-D. (The multi-dimensional code would of course have applications beyond this problem which has a solution in reduced dimensionality, e.g. shapes other than sharp cones.) In this case, there is no loss of terms in the governing equations and the answers as $\Delta \rightarrow 0$ should be identical. The only concern is that the benchmark ODE solution should be more accurate than the level of accuracy required for the 2-D code verification. In fact, this is easy to accomplish, given the high level of accuracy and reliable error estimation of several readily accessible ODE solver packages. For a small category of such problems where reduced dimensionality can be achieved for some non-trivial problems, an ODE benchmark is very effective for Code Verification. The numerical accuracy of an ODE benchmark is *more* reliable than that of infinite series solutions.[21]

It is also possible to use highly accurate PDE solutions (e.g. high resolution spectral methods) for benchmarks in Code Verification, but this is much more difficult to achieve than for ODEs. It must be kept in mind that we are not looking only for engineering accuracy in the agreement between the output variables of the tested PDE code and the benchmark PDE code, because even small differences can corrupt the Verification of observed convergence rate for the tested code. However, the same is true for infinite series solutions used as benchmarks. For relatively simple infinite series solutions, the difficulties perhaps would be comparable (though the computer use would be less for infinite series). For a difficult problem like the Sudicky-Frind problem as described above in Section 3.10.1, numerical PDE solutions can be more reliable, as well as more flexible.

3.11 SENSITIVITY OF GRID CONVERGENCE TESTING

We have had considerable experience in using such systematic grid convergence tests, both with realistic and with more general Manufactured Solutions, for accuracy Verification of Codes as well as Verification of individual Calculations (involving error estimation). Many of these cases have utilized the Generalized Richardson Extrapolation and the Grid Convergence Index (GCI) to be described in Chapter 5. In our experience, this method of code accuracy Verification via systematic grid convergence testing (whether or not the GCI is used) is *remarkably sensitive* in revealing code problems, as indicated by the following examples. (Some of these examples will be covered in more detail in later chapters.)

[21] Oberkampf and Trucano (2007, 2008) rank numerical ODE and PDE benchmarks below infinite series solutions for reliability. They also recommend use of ODE solvers of higher order accuracy than used in the code to be Verified, and use of two separate ODE solvers (with detailed descriptions of the ODE algorithms) in order to qualify as a "Strong Sense Benchmark." I find these recommendations to be unnecessary burdens. The only requirement is that the *accuracy* of the ODE solution, not the order, be higher than that of the tested code. The accuracy can be achieved by the higher resolution enabled in 1-D (as well as higher order, typically). The reliability of the ODE error esimators is so high that multiple code solutions are not necessary. Also, contrary to claims, there is only small inaccuracy introduced by the presence of two-point boundary values, and the added complexity, while interesting, is easily treated and much simpler than infinite series evaluations. The authors do agree on the superiority of MMS.

1. In accuracy Verification tests of a groundwater flow commercial code, a 1st-order error in a *single corner cell* in a strongly elliptic problem caused the observed *global* convergence to be 1st-order accurate. (Roache et al, 1990.)
2. In accuracy Verification tests of our SECO_FLOW_2-D variable density (dissolved salt) groundwater flow code, 1st-order extrapolation for *ghost cell values of only one quantity* (aquifer thickness) along one boundary caused the observed convergence to be 1st-order accurate. (Roache, 1993.)
3. In groundwater contaminant transport calculations (advection-diffusion + decay, retardation, and matrix diffusion), use of a plausible single-grid-block representation for a point source as the grid is refined introduces error in a finite volume (or block-centered finite difference) formulation. In this cell configuration, the cell faces align with the boundaries of the computational domain, and doubling the number of cells requires the *location* of the single cell representing the source to *shift* by $\Delta/2$. It is to be expected that the solution accuracy in the neighborhood of the source would be affected. But surprisingly, the accuracy of time-integrated *discharge* across boundaries far from the source was also degraded to 1st-order accuracy. See Chapter 6, Section 6.12, and Salari et al (1995).
4. In tests of the two-dimensional SECO_FLOW groundwater flow code using the option for node (cell center) placement on the boundary (rather than cell edges on the boundary), the domain corner nodes enter into the calculation only when cross derivative terms are present. (These can arise from either non-orthogonal coordinate transformations or tensor properties.) When this value (say, at the lower left corner *ic, jc*) was set by plausible averaging, as in

$$f(ic, jc) = 1/2\ [\, f(ic+1, jc) + f(ic, jc+1)] \tag{3.11.1}$$

   the result was global 1st-order convergence rate. When the following form (recommended by S. Steinberg)

$$f(ic, jc) = f(ic+1, jc) + f(ic, jc+1) - f(ic+1, jc+1) \tag{3.11.2}$$

   was used, the result was global 2nd-order convergence rate. (Roache et al, 1990.)
5. The observed convergence rate of ostensibly 2nd-order accurate turbulent boundary layer codes (Wilcox, 1993) can be degraded, apparently by *conditional statements* limiting eddy viscosity and defining the boundary layer edge. (Wilcox, 1995.)
6. Airfoil codes can exhibit the expected 2nd-order convergence rates for lift and drag, but less for moment, possibly because of approximations involved in applying *quasi-periodicity* across cut-planes of a C-grid. (Salari, 1995.)

3.12 EXAMPLES OF UNANTICIPATED CONVERGENCE RATES DETERMINED BY SYSTEMATIC GRID CONVERGENCE TESTS

Note that these systematic grid convergence tests can Verify a code, not only in the sense of identifying presence of coding errors, but also in the sense of Verifying the order of convergence. Contrary to common belief, the theoretical order of convergence may not be so obvious for complicated methods. Jameson and Martinelli (1996) noted that "the nonlinearity of conservation laws, and the geometric complexity of the flow domain of interest in most applications, make it difficult to devise elegant analytical tools for the accuracy Verification of a numerical scheme."

Furthermore, "*the* formal analysis" may be a misnomer, since different levels of analysis may provide different approximate solutions. (Not all the world is governed by the Laplace equation in a square domain.) Here we give two examples.

3.12.1 Reduction to Periodicity Method: Unequal Orders of Accuracy for Derivatives

An excellent example of the difficulty of determining "*the* order of accuracy" for unusual methods is given by the following method, which is a combination of a very high accuracy base algorithm for periodic solutions combined with a heuristic extension to non-periodic solutions.

In Roache (1978), the "reduction to periodicity" (RTP) method was devised to use pseudo-spectral FFT (Fast Fourier Transform) methods for non-periodic problems by pre- and post-processing the solution with polynomials of various degrees. The accuracy was not confidently predictable *a priori*. The testing involved parametric combinations (4 methods of reduction-to-periodicity, 5 values of the reducing polynomial function N), 3 error indices of two derivatives, and comparison with 2nd-, 4th- and 6th-order finite difference methods (FDM). The results were given in rankings compared to the FDM, for two static functions and the transient Burgers equation. An example of the accuracy ranking obtained for the case of using exact values of boundary derivatives in the RTP method is shown in Table 3.12.1.1. The details of the technique are not required here; the point is the unusual and unanticipated grid convergence behavior.

Note the unusual convergence rates exhibited by this family of methods. The convergence rate for the second derivative $f''(x)$ is two orders lower than that of the first derivative $f'(x)$ for RTP with $N = 3$ or 5, but is equal for $N = 2$ or 4. For $f'(x)$, the accuracy jumps two orders when N is increased from 2 to 3, and from 4 to 5, but does not improve when N is increased from 3 to 4. This *modulo* 2 behavior is shifted for $f''(x)$, which jumps two orders when N increases from 3 to 4, but does not improve when N increases from 2 to 3, nor from 4 to 5.

Successful analysis of this technique is possible, but far from conventional. Lyness (1974) had analyzed the similar Lanczos representation of a function, and showed theoretically and in general what the

$N = 2$	O(2) < RTP < O(4)
$N = 3$	O(4) < RTP < O(6)
$N = 4$	O(4) < RTP < O(6)
$N = 5$	O(6) < RTP

a. Accuracy ranking for $f'(x)$ for the RTP method.

$N = 2$	O(2) < RTP < O(4)
$N = 3$	O(2) < RTP < O(4)
$N = 4$	O(4) < RTP < O(6)
$N = 5$	O(4) < RTP < O(6)

b. Accuracy ranking for $f''(x)$ for the RTP method.

Table 3.12.1.1. Order of accuracy ranking for the Reduction to Periodicity method (RTP) applied to a static function. The mesh spacing was varied from $\Delta = 1/8$ to 1/128. The notation O(2) < RTP < O(4) indicates that the RTP technique had an accuracy between 2nd-order and 4th-order FDM. From Table I of Roache (1978).

above tests showed experimentally for only specific functions: (1) that the degree of the reducing polynomial is important to the accuracy, (2) that the accuracy of $f'(x)$ would improve as N was increased from $N = 1$ by increments of 2. His theory also showed other results (not shown in the Table) found only experimentally in Roache (1978), that (3) when boundary derivatives for the RTP technique are evaluated by FDM (rather than using exact values as in the above example), the order of the FDM was not so important, and that (4) the use of a 5-th degree reducing polynomial with 2nd-order FDM at boundaries would give $f'(x)$ to overall 4th-order accuracy. At my request, Lyness also extended his previous work to prove another aspect found only experimentally, that (5) $f''(x)$ is two orders less accurate than $f'(x)$. (This is not a characteristic of pseudo-spectral or spectral methods themselves, but of the RTP method.) This imbalance in the order of accuracy of
$f'(x)$ and $f''(x)$ is in fact acceptable, giving the chance of a "balanced truncation error method" (Roache, 1978) in which the *size* rather than the *order* of truncation errors from advection and diffusion terms are roughly balanced for high Reynolds number (Peclet number) problems.

This complicated and unanticipated convergence behavior would not likely be determined without the kind of systematic grid convergence tests recommended herein. Note, however, that with the numerical grid convergence tests providing motivation, theoretical justification of this behavior was found using the independent results of Lyness (1974).

3.12.2 Completed Richardson Extrapolation: Higher Order Truncation Error Interaction

Richardson Extrapolation will be covered extensively (and generalized) in Chapter 5. For now, we note that the original method (Richardson, 1908) combines two 2nd-order solutions on two grids, the fine grid having twice the resolution of the coarse grid, to produce a 4th-order accurate solution on the coarse grid only. I devised a method (Roache and Knupp, 1993) for easily obtaining a 4th-order solution on the entire fine grid, i.e. obtaining a "completed" Richardson Extrapolation. The method consists simply in evaluating the solution at the "skipped" fine grid points by linear interpolation, not of the solution, but of the coarse grid *correction* between the 2nd- and 4th-order solutions. Richards (1997) extended the procedure systematically to mixed time and space PDEs with more general integer refinement.

Without going into the details here, the original formal analysis of the completed Richardson Extrapolation indicated that all fine grid nodes should converge at 4th-order rate, i.e., $p = 4$. Systematic grid convergence studies did show this experimental rate $p = 4$, but originally only for the Laplace and Poisson equations (with non-trivial source terms). For the advection-diffusion equation, the "skipped" fine grid points mysteriously showed roughly 3rd-order convergence, $p = 3$. It was not known if the source of discrepancy was an error in the formal analysis of the convergence rate, or a coding error.

The problem lay unexplained for a few years, until P. Knupp extended the formal analysis to show (Roache and Knupp, 1993) that the asymptotic rate $p = 4$ indicated by my original formal analysis was indeed correct, but would not emerge until the grid refinement had proceeded enough to reduce the cell Reynolds number $Rc << 3$. The cause is higher order truncation error interaction determined by the $\exp(x)$ form of the exact solution. This is demonstrated in Table 3.12.2.1 for $Re = 16$. For the coarse grid, from $\Delta = 1/N = 1/4$ to roughly 1/64, $E_3 = \text{max error}/\Delta^3$ is *roughly* constant, which might suggest $p = 3$. Only for finer grids is the true asymptotic behavior of $p = 4$ approached, with E_4 increasing only 3.2% from $N = 512$ to 1024. Note also the difficulty of judging the rate of convergence just from a presentation of max error, especially for $p > 2$.

N	Maximum Error	E_3	E_4
4	0.63299798320	40.512	162.05
8	0.09835113825	50.356	402.85
16	0.01602753711	65.649	1050.38
32	0.00208620873	68.361	2187.55
64	0.00019995963	52.429	3354.77
128	0.00001580555	33.135	4242.77
256	0.00000111805	18.758	4802.00
512	0.00000007447	9.995	5117.81
1024	0.00000000481	5.165	5285.92

Table 3.12.2.1. Grid Convergence Study for Completed Richardson Extrapolation. Linear advection-diffusion equation with Re = 16. E_3 = max error/Δ^3 and E_4 = max error/Δ^4. (From Table IVb, Roache and Knupp, 1993.)

The truncation error interaction is indicated by Knupp's analysis. The one-dimensional steady state equation solved is the following ODE for $f(x)$.

$$f'' - Re\, f' = 0,\ f(0) = 1,\ f(1) = 0 \tag{3.12.2.1}$$

The exact solution is

$$f(x) = (e^{-Rex} - e^{-Re}) / (1 - e^{-Re}) \tag{3.12.2.2}$$

From the exact solution, the form for any k-th derivative is

$$f^{(k)}(x) = \frac{(-1)^k Re^k e^{-Rex}}{1 - e^{-Re}} \tag{3.12.2.3}$$

In particular, the relation between any k-th derivative and the 2nd derivative is

$$\frac{f^{(k)}}{f^{(2)}} = (-1)^k Re^{k-2} \tag{3.12.2.4}$$

This is the aspect of the solution that causes the truncation error interaction. The Taylor Series for (any) f is

$$f(x + \Delta) = \sum_{k=0}^{\infty} \frac{\Delta^k}{k!} f^{(k)}(x) \tag{3.12.2.5}$$

Using the relation between any k-th derivative and the 2nd derivative, which is a feature only of this solution, we obtain

$$f(x+\Delta) = \left[f(x) + \Delta f'(x)\right] + f^{(2)}(x)\left\{\frac{\Delta^2}{2!} - \frac{\Delta^3}{3!}Re + \frac{\Delta^4}{4!}Re + \ldots\right\} \tag{3.12.2.6}$$

The significance of this expression is that the higher-order error terms contain not only the grid spacing Δ, but also the continuum parameter *Re*. Now,

$$F(x) = f(x) + \Delta f'(x) \tag{3.12.2.7}$$

acts like a 2nd-order approximation to $f(x+\Delta)$ only if

$$\Delta^2 / 2 >> \Delta^3 Re / 6, \cdots \tag{3.12.2.8}$$

This translates into the requirement that $1/\Delta = N >> Re/3$. Therefore, if *Re* is large, a large *N* (small Δ) is needed to make *F* behave in a 2nd-order accurate manner. Otherwise, the assumption necessary for the completed Richardson extrapolation is violated, and the fine-grid results are not 4th-order accurate.

As in the previous example of the Reduction To Periodicity method, this unanticipated convergence behavior would not have been determined without the kind of systematic grid convergence tests recommended herein. And again, with the numerical grid convergence tests providing motivation, theoretical justification of the behavior was determined using more detailed formal analysis. Note that the original formal analysis was not *wrong*, i.e. it applied asymptotically, but neither was it complete enough to adequately describe the experimental results.

These two examples illustrate an important and usually overlooked aspect of numerical analysis. Authors typically speak of "*the* formal convergence rate" as though only one analysis were possible. In fact, formal analyses of convergence rates can be at various levels, with various results, especially for nonlinear problems. The second example above shows unanticipated structure in the analysis, even for a linear, one-dimensional, steady state problem with uniform grid increments Δx.

3.12.3 Δ Three More Examples

For three other examples showing the non-uniqueness formal truncation error analysis, consider first that the *formal* accuracy of the QUICK schemes, first published in 1979, was still being debated in 1993, and "considerable confusion remains in the literature" (Leonard, 1993). In another paper (not related to this question), Leonard and MacVean (1995) provided a formal analysis that illuminates grid convergence effects due to nonlinearities (see Section 8.2). And Blackwell et al (2009) demonstrated second-order convergence for an enclosed radiation problem, verifying their own non-rigorous theoretical analysis but contrasting it with another that indicated $p = 3$. They noted the following. "In the present problem, as in many practical problems, the analysis for theoretical *p* is neither straight-forward nor unique. The absence of a theoretical *p* should not be used as an excuse for not performing a grid refinement study."

3.13 Δ MULTIPLE SCALES, MULTIPHYSICS, AND TURBULENCE MODELING

There are special considerations required for problems with multiple scales of solutions, notably turbulence modeling (which exhibits multiple scales even for single physics) and multiphysics problems. Here, the code theoretical performance can be Verified (within a tolerance) for a range of parameters, but could fail in another range.

It is necessary to get the grid resolution into (or close to) the asymptotic range in order to obtain good results for observed *p*. Virtually any grid is in the asymptotic range for a simple Laplace equation. For any boundary layer calculation, it is clear that the initial (coarse) grid must get some points into the boundary layer. For RANS (Reynolds Averaged Navier-Stokes) turbulence modeling without wall functions (Shirazi and Truman, 1989; Wilcox, 1993) the grid must get some points into the wall layer. For turbulence modeling with wall functions, it would seem that the grid preferably should not get into the wall layer (Celik and Zhang, 1995). In our interpretation, effectively the wall functions should be viewed as an elaborate nonlinear boundary condition, and the grid convergence exercise should be done from the edge of the wall layer out. This interpretation has now been confirmed in extensive MMS applications to RANS turbulence models; see Eça and Hoekstra (2006, 2008), Eça et al (2007, 2009). (However, see Celik and Karatekin, 1995 for a counter-example.)

Similarly, for large eddy simulations (LES) as used in aerodynamic turbulence research and in atmosphere and ocean modeling with sub-grid turbulence modeling, the grid convergence must not go to zero, or else the Reynolds stresses will be counted twice, once from the full Navier-Stokes terms and again modeled from the LES terms. (See also Section 6.17.2.) Also, the presence of any switching functions, such as length determinations for the Baldwin-Lomax turbulence model (Wilcox, 1993), can easily corrupt 2nd-order convergence rates. Also, the particular quirks of various RANS turbulence models (variants of k-ε, k-ω, Spalart-Allmaras, etc.) place subtle requirements of consistency and resolution on the Manufactured Solutions. Switching functions may need to be disabled for the MMS procedure. Although these obstacles can be overcome, it is not a trivial task to generate MMS for RANS turbulence models. The work of Eça et al (2007b,c) is highly recommended; the web sites given therein contain codes for the analytical source terms. See many applications in the Lisbon II and III V&V Workshops (Eça and Hoekstra, 2006, 2008; Eça et al, 2007, 2009). MMS results for RANS models are sometimes unexpected; for example, the observed asymptotic order of convergence for the Spalart-Allmaras method is less than the theoretical (Eça et al, 2007a).

The grid resolution requirements are much more demanding for turbulent boundary layers, just as laminar boundary layers are much more demanding than inviscid flows. For example, Claus and Vanka (1992) found that 2.4 million nodes ($256 \times 96 \times 96$) did not demonstrate grid independence of the computed velocity and turbulence fields of crossflow jets. In the Lisbon III V&V Workshop (Eça and Hoekstra, 2008) the 2-D turbulent flow over a backstep at moderate $Re = 5\times10^5$ required 401×401 nodes using stretched grids to resolve the boundary layers.[22]

Multiphysics codes can cause difficulty in judging convergence and estimating numerical error and uncertainty because different parts of the physics may converge differently. This can cause false-positive indications of attaining the asymptotic range, i.e. a grid may be in the asymptotic range for some physics and not for others.

3.14 Δ WARNINGS: WHAT THE METHOD DOES NOT "VERIFY"

The Method of Manufactured Solutions convincingly Verifies the *accuracy* of the solution produced by a code, and this is the technical definition of Verify adopted herein. As already noted, the method cannot Verify the coding accuracy of individual terms in mixed-order methods, nor can it Verify algebraic conservation.

Also, the method does not "verify" (in the more general sense) that everything about the coding is correct, since not all coding mistakes affect accuracy. In particular, the method does not verify that the *solution procedure* encoded actually follows the algorithm intended. It verifies that the procedure produces

[22] The problems also required Least-Squares GCI; see Chapter 5.

the correct numerical solution, but there can be coding mistakes that decrease the iteration convergence rate yet still produce the correct solution; the mistake causes a loss of *efficiency*, rather than a loss of *accuracy*. (Note that such coding mistakes should not be a concern to regulatory agencies or stakeholders in public policy projects like geologic waste disposal, since the numerical accuracy of the answers is not affected.)

Examples of such coding mistakes are plentiful. An SOR iteration scheme can have an mistake in the determination of the optimum relaxation factor without (significantly) changing the answer produced. If the mistake is gross, the slow iteration convergence will probably bring attention to the mistake and lead to successful debugging, but smaller mistakes may go unnoticed permanently. Multigrid methods (Brandt, 1977; McCormick, 1989) are known to be prone to subtle coding mistakes that diminish the iteration convergence rate from the theoretical rate. Verification of a theoretical *iteration* convergence rate is a good debugging tool, and builds high confidence for linear problems, but for nonlinear problems, the conclusions about correctness of the coding are not as firm as they are for Verification testing of numerical accuracy. Other esoteric mistakes that are difficult to detect will be described in Chapter 8.

Likewise, code verification of a solution adaptive algorithm is problematical, with or without MMS, because the coding could contain mistakes that reduce efficiency but still produce correct answers in the limit (as it would without solution adaptivity). However, the efficiency of solution adaptivity can be monitored with systematic grid convergence tests using MMS, resulting in high confidence; for excellent examples by Prof. D. Pelletier and colleagues, see Hay and Pelletier (2007, 2008) and references therein.[23]

3.15 ROBUSTNESS AND CONFIDENCE

Likewise, code/algorithm robustness is not "Verified" by the procedure. Robustness is necessarily a qualitative concept, so it does not make sense to ask for a definitive statement about it, especially for general purpose commercial codes.

One technique for building confidence (rather than a rigorous demonstration) in robustness and coding accuracy was given in Roache (1995). The questions were (1) robustness of direct elliptic marching methods (which are sensitive to computer round-off error) to various boundary conditions, and (2) correctness of the coding for mixed boundary conditions. A general boundary condition had been formulated and coded as

$$\alpha f + \beta \frac{\partial f}{\partial n} = \gamma \tag{3.15.1}$$

where f is the dependent variable and n is the direction normal to the boundary. For various combinations of α, β, and γ one obtains the usual homogeneous or non-homogeneous Dirichlet, Neuman, or Robin (mixed) boundary conditions. (Because of algorithmic quirks, the formulation is not so general as to include tangential derivatives at the boundary.) Robustness and coding were demonstrated by running many calculations with α, β, and γ arrays determined by a pseudo-random number generator. The values were not constant along a boundary, but changed randomly along the boundary. The range was limited so as to avoid indeterminacies in the discrete formulation. Since the elliptic marching methods are direct, i.e. non-iterative, the stencil evaluation for the error should be algebraically zero, except for round-off error, and the

[23] Hetu and Pelletier (1992), Ignat et al (1998), Ilinca et al (1995, 1997a,b), Pelletier and Ilinca (1994), Pelletier et al (1995), Pelletier and Ignat (1995), Pelletier and Ilinca (1997), Pelletier and Trepanier (1997), Pelletier and Roache (2002, 2006).

tests Verified this convincingly. (Note this interesting generality: solution methods that are sensitive to computer word length, i.e., to round-off error, are easy to debug or to demonstrate coding correctness, whereas solution methods that are not sensitive to word length are difficult.)

The approach of testing the general Robin boundary condition by pseudo-random number generation of coefficients can be combined with the Method of Manufactured Solutions. With the analytical solution evaluated for f and $\partial f / \partial n$ at every point, two of the three coefficients α, β, and γ can be determined by pseudo-random number generation, and the third solved algebraically. This test will provide a convincing demonstration of both accuracy Verification and robustness to boundary condition type.

3.16 § ULTIMATE RESPONSIBILITY FOR CODE VERIFICATION

The question of ultimate responsibility for Code Verification arises with the now widespread use commercial codes. Clearly, vendors have a responsibility not only to perform thorough Code Verification, but to document the Verifications and make the documents available. This is not typically the case, and colorful graphics have more sales appeal than hard quantitative evidence. (The professional fault must be shared by the customers for not demanding more substance.) But ultimate responsibility rests with users, commensurate with the risks and consequences of their application.

For high-consequence applications (e.g. the nuclear power industry) the code users cannot shirk their responsibility to verify the codes they are using (only for the options and parameter ranges of interest) if the code vendors have not done a convincing job (i.e. both thorough and thoroughly documented).

It is often argued that conscientious Code Verification is not required if the code's use is only for initial screening of candidate designs, when only trends with parametric variations are of interest. This is an unfounded and dangerous argument. The standards of accuracy for preliminary design decisions may well be less than for a final design code, but V&V is still required. To state what should be obvious: An erroneous code can predict erroneous trends.

3.17 § CODE VERIFICATIONS AT COMPONENT AND SYSTEM LEVELS

During code development, each component or subprogram will naturally be tested independently. This testing may well constitute what we call Code Verification for each component, e.g. Verification of a real-gas equation of state. It is also advisable to verify the entire package, i.e. at the system level, since this also verifies the coupling of components, and accomplishes more in one Verification operation. This is important for "regression testing", a term applied to semi-automated re-verification of codes operating in an unstable environment, e.g. changing computer hardware or compilers in a distributed processing system. (One laboratory routinely runs code re-verification regression testing for a cutting-edge distributed system on a nightly basis.)

Any distinction of components vs systems is of course somewhat arbitrary. The component "Transport Properties" could involve many components and therefore could itself be named "system level" at some zoom level, whereas an entire flow solver "system" (e.g. NASA-Langley's FUN3D) could be merely a component of a design code or a fluid-structure interaction code. The point is, that for final Verification it is advantageous to aggregate as many features as possible into a single test, since this reduces the number of runs required for independent replication (i.e. re-verification or Confirmation). But it is more difficult to come up with an MMS solution that exercises all the components, and in some cases, this is impossible in a single case because some of the code components are mutually exclusive, i.e. they refer to different options. Nor is it necessary.

Kleb and Wood (2004, 2006) insisted (correctly, I believe) on requiring that component level verification tests be selected and performed by the developers[24] (presumably of public domain codes) and published (whatever that may mean today). They endorse the MMS method. They tie this in with a fundamental philosophical criterion for science itself, that of independently-verifiable[25] experiments. They asserted that current (2006) practice often violates the scientific method, supported with their survey that showed only 22% of new models published were accompanied by tests suitable for independently verifying the new *code*.[26] They noted the need for institutionalizing component-level testing (since we are beyond the old "cottage industry" level of code development and application) which will require advances in electronic documentation methods such as the Amrita system (Quirk, 2007).

3.18 § FURTHER APPLICATIONS OF MMS

3.18.1 § Further Applications of MMS outside of Code Verification

Besides its original use in code verification, MMS has been used to evaluate methods for Solution Verification (Chapter 5). In this application, MMS is used to generate *realistic* exact solutions for RANS turbulent flows to assess calculation verification methods like the GCI and Least Squares GCI, for estimation of iteration errors, and for estimation of errors due to outflow boundary conditions; see V&V1, Roache (2002), Pelletier and Roache (2006), Eça and Hoekstra (2006a, 2007, 2009b). [See also Section 5.10.10.3) and Eça et al (2005, 2007).] Methods for detection of singularities in Computational Solid Mechanics have also been evaluated with this approach, termed "Tuned Test Problems" or TTP by Sinclair et al (1997, 2006); see Section 5.10.4. They also used realistic manufactured solutions to determine acceptable grid resolution for the actual problems, another application of MMS/TTP outside of Code Verification. The MMS may also be used in code development to assure that the solver is working correctly on any solution grid (Ghia, 2008).

3.18.2 § Further Applications of MMS in Code Verification

Since the publication of V&V1, there have been many successful applications of MMS to a wide range of difficult problems. The tutorial paper (Roache, 2002) is reproduced in Appendix C and contains references to MMS problems including compressible flows, radiation problems including eigenvalues, and others. The book by Knupp and Salari (2003) is recommended, notably for a blind study of code error detection by MMS for a compressible flow code. Other expositions are given in Pelletier and Roache (2006) and Wang and Jia (2009). Roy et al (2004) successfully applied MMS to convincing verifications of Euler and compressible Navier-Stokes codes. Blackwell et al (2009) applied MMS to enclosure radiation, verifying their non-rigorous theoretical analysis that indicated $p = 2$ in contrast to another

[24] Their claim in the first paper that the V&V community (including myself) did not endorse component verification was retracted in the second paper.

[25] However, in the older terminology of the philosophy of science, e.g. Popper (1980), "verify" applies to what we now term "validate" whereas our "verify" would correspond to what was classically just a matter of checking one's hand and slide-rule calculations.

[26] Actually they used the phrase "verify the new *model*" which does not fit our terminology; codes are Verified, models are Validated. They also use *verify* in the sense of demonstration of algorithm accuracy on simple problems.

analysis that indicated $p = 3$. Appendix A of V&V20 contains an MMS heat conduction problem with discontinuous step change in conductivity and contact resistance.

Bond et al (2007) presented an exemplary study applying MMS to CFD code verification with several insightful observations. The FEM code being verified solved Euler, Navier-Stokes and RANS equations on skewed, non-uniform, unstructured 3-D meshes. Particular emphasis was placed on verification of numerical boundary conditions: slip, no-slip (adiabatic and isothermal), and outflow (subsonic, supersonic, and mixed), and on code segments that calculate solution gradients, a non-trivial issue in hexahedral grids with high aspect ratios near boundaries. The more demanding L_∞ norm was used and recommended, as well as the usual L_1 and L_2 norms. Among many interesting results, one provided a particular caution regarding precision issues. The symbolic manipulation code used to generate source functions wrote source code in double precision but with only single precision constants, which later corrupted the initial Verification exercise. The authors recommended an additional criterion for claiming Verification of double-precision accuracy; the relative errors should be smaller than the single precision limit. Another caution involves orientation of the outflow boundary in supersonic flow along a constant pressure surface, which might permit certain coding errors to go undetected. (The difficulty arose due to an ambitious approach of building boundary condition values into the MS, rather than treating them crudely with the source term.) Especially noteworthy was the success of MMS is disclosing a weakness of the solution algorithm in regard to partitioning of multiprocessors. The paper is also valuable for presenting anecdotal debugging history, rather than a simple "pass" evaluation.

An illustration of MMS applied to unsteady flows is given by Eça and Hoekstra (2007b). For the 2-D laminar flows, a general formulation was developed that allowed any analyst to specify an arbitrary continuous function that is incorporated into an analytical form for velocities which satisfy the incompressible continuity constraint exactly. Likewise, non-slip and impermeability conditions are met exactly by the MS. Two time-dependencies were considered: and exponentially decaying solution and a periodic solution. The exercise Verified the code, and additionally shed light iteration error.

It is clear that MMS is now undeniably mature and widely accepted as the gold standard for Code Verification.

CHAPTER 4

ERROR ESTIMATION FOR QUANTIFICATION OF UNCERTAINTY: VERIFICATION OF CALCULATIONS

There are no whole truths; all truths are half-truths.
It is trying to treat them as whole truths that plays to the devil.

Alfred North Whitehead

4.1 INTRODUCTION[27]

As noted in the previous chapters, Code Verification, even if performed in some idealized sense (i.e. a verification that a code is completely free of coding errors and algorithm errors), does not remove the requirement for estimating the discretization error for a particular calculation or solution, referred to as Calculation Verification (or Solution Verification, as in V&V20). Also, the task of Code Verification as presented does not, strictly speaking, involve error *estimation* but simple error evaluation, because the Verification is done with respect to a known analytical solution. The solution may be unrealistic, e.g. a physically meaningless "manufactured" solution, but is nonetheless useful for Code Verification. Verification of an individual Calculation, on the other hand, strictly involves error *estimation*. Yet the latter term needs some distinction because of its use in different contexts.

[27] This chapter is taken primarily from Roache (1997), "Quantification of Uncertainty in CFD."

4.2 ERROR ESTIMATION FOR GRID ADAPTATION VS. QUANTIFICATION OF UNCERTAINTY

The focus of this book is the "quantification of uncertainty," the estimation or "banding" of the numerical error of a "final" calculation in computational PDEs. By "final calculation" I mean one which is considered to be used "as is." This provides a different motivation than that of an error estimate to be used for the process of solution adaptive grid generation. Although the present methods may be applicable to grid adaptation, and the developments of some of the methods described herein were motivated by that problem, the grid adaptation problem has vastly different (though usually unacknowledged) requirements.

The key word is "quantification" of uncertainty, as opposed to vague and all too common qualitative assessments. Quantification of Uncertainty may also involve more than just obtaining a good error estimate; in fact, the more conservative procedure based on the Grid Convergence Index (described in the following Chapter 5) reports an error band equal to three times an error estimate. Further, I consider herein only *a posteriori* error estimation, being of the opinion that useful *a priori* estimation is not possible for non-trivial problems.

Local error estimators are used for solution adaptive grid generation algorithms, and they are usually successful for this purpose (e.g., see Oden et al, 1993). However, almost anything intuitive is successful for adaptation purposes, e.g. minimizing solution curvature, or adapting to solution gradients, even though solution gradients *per se* cause no error in most discretization schemes. In one-dimensional problems (e.g., Salari and Steinberg, 1994) or quasi-one-dimensional problems (Dwyer et al, 1980), the gains in computational efficiency from solution adaptivity of the *r*-type (redistribution) in structured grids are very impressive. In strongly multidimensional problems, gains are usually modest (Roache et al, 1984; Hall and Zingg, 1995) but are more significant for unstructured grid adaptation (e.g., see Morgan et al, 1991; Lohner, 1989; Hetu and Pelletier, 1992; Pelletier and Ignat, 1995; Coorevits et al, 1995; Ilinca et al, 1997).

Unfortunately, this task of grid adaptation as typically practiced has little connection to the quantification of uncertainty for a final calculation with any useful error measure. Thus, it is important to recognize that the successes of these local error estimators in guiding grid adaptation must not be taken as demonstrations of their efficacy for the quantification of uncertainty. Even for grid adaptation, the success of local error estimates is only a partial success. Hagen et al (1997) have given examples of strong non-localness in tide-driven simulations of ocean model problems. Increased element resolution in the deep ocean, where local error estimators would not suggest increased resolution, in fact improves accuracy far away from the deep ocean, on the continental slope and shelf. Such behavior may be expected for any wave-dominated problem exhibiting amphidromes, "hanging" shocks, or other local features of strong solution structure with no obvious geometrical cause in the immediate neighborhood.

Lee and Yeh (1994a,b) have shown good correlations between local (normalized) weight functions based on solution-gradient estimators and (normalized) true solution errors as obtained with multiple grid solutions and Richardson Extrapolation, for both laminar and turbulent steady flows (although somewhat less for swirling flows). Their hybrid solution adaptive grid generation (combining a global, *r*-type adaptation followed by a local *h*-type adaptation) obtained significant efficiencies. Nevertheless, Lee and Yeh (1994b) correctly noted the distinction between an error "indicator" and an error "estimator," and noted that "the weight function is employed as an error 'indicator' rather than an 'estimator', in that the accuracy of the predicted values is not a primary concern." Clearly, *normalization* is a key concept. The distribution of errors, provided by the correlation between normalized values, is definitely of interest but is not sufficient for the task of the quantification of uncertainty. Rather, absolute values are required, and correlation of these with local weight functions (and their use as surrogate estimators for measures of interest) must be established anew for each new problem and each error metric of interest.

We need *global* error estimates for Verification of Calculations. By "global" we do not here mean just a global summing up of local values (as sometimes used in the literature of Finite Element Methods) but an evaluation which includes non-local effects, i.e., which takes account of the fact that errors are advected, diffused, "beat" in wave resonance problems, etc.

4.3 TAXONOMY FOR ADDITIONAL INFORMATION FOR ERROR ESTIMATES

Once we have produced a discrete solution of the governing partial differential equations, it is clear that we require some *additional* information in order to quantitatively estimate the uncertainty or numerical accuracy. The following taxonomy of sources in Table 4.3.1 of this additional information will provide a framework for the discussion. By the word "grid," we refer to any measure of discretization, i.e., Cartesian grid, non-orthogonal grid, number of Fourier modes in a spectral solution, number of discrete vortices, etc.

A. Additional Solution(s) of the Governing Equations on Other Grids
   - A.1 Grid Refinement
   - A.2 Grid Coarsening
   - A.3 Other Unrelated Grid(s)

B. Additional Solution(s) of the Governing Equations on the Same Grid
   - B.1 Higher Order Accuracy Solution(s)
   - B.2 Lower Order Accuracy Solution(s)

C. Auxiliary PDE Solutions on the Same Grid

D. Auxiliary Algebraic Evaluations (AAE) on the Same Grid; Surrogate Estimators
   - D.1 Non-Conservation of Conservation Variables
   - D.2 Non-Conservation of Higher Moments
   - D.3 Zhu-Zienkiewicz and Wiberg Type Estimators
   - D.4 Convergence of Higher Order Quadratures

TABLE 4.3.1. Sources of Additional Information for Error Estimation, Given a Discrete Solution of the Governing Partial Differential Equations on a Grid.
(From Roache, 1997.)

The following are brief remarks on this taxonomy, some of which will be justified and amplified in later discussion.

Categories A and B, Additional Solution(s) of the Governing Equations (on Other Grids or the Same Grid) involve the direct, unambiguous evaluation of any error measure of engineering or scientific interest. Note that *only* these procedures can be used, as in the previous chapter, for Verification of Codes, and as a practical matter, only A.1, Grid Refinement, provides for rigorous Verification, because only it can be continued to any desired level of accuracy (like the classical ε-δ limit proofs of calculus).

For Category A, Additional Solution(s) of the Governing Equations on Other Grids, no additional code development or modifications are required.

For Category B, Additional Solution(s) of the Governing Equations on the Same Grid, no additional grid generation is required.

Category C, Auxiliary PDE Solutions on the Same Grid (e.g., Van Straalen et al, 1995) does not simply involve a local evaluation of something. The key aspect here is that errors are transported, advected, diffused, etc. (However, it is also true that a simple local evaluation of something, without advection, is just

what one needs to guide solution adaptation; hence the different needs of error estimation for solution adaptation vs. quantification of uncertainty for a final calculation, as noted above.)

Category D methods, Auxiliary Algebraic Evaluations on the Same Grid, are relatively cheap, need no additional grid generation, and (sometimes) use no significant dynamic memory. However, the "error" evaluated usually has no direct relation to any error measure of engineering or scientific interest; hence my use of the term "surrogate estimators."

Category B, C and D methods, none of which require multiple grids, will be discussed in Chapter 7.

4.4 GRID REFINING AND COARSENING

Systematic grid convergence studies are the most common and straightforward methods, and arguably constitute the most reliable technique for the quantification of numerical uncertainty. Unlike the other methods available, this approach can be used to dependably consider the convergence of any quantity of interest, as well as the usual L_2 and L_∞ norms.

By "grid convergence studies" people usually mean Category A.1, Grid Refinement, but Grid Coarsening, Category A.2, usually would make more sense. If completely solved solutions are obtained on two grids, presumably the finer grid solution would be used, so the coarse grid solution could be used to estimate the error of the fine grid solution. Whether one refines or coarsens just depends on which grid was calculated first! So for completely solved solutions, grid refining and coarsening are identical.

A disadvantage of Category A methods is that multiple grid generations are required. Cartesian grids obviously pose no problem. For boundary-fitted structured grids, the simplest method for grid doubling (halving) is to generate the finest grid first, using whatever method is preferred (e.g., see Thompson et al, 1985 or Knupp and Steinberg, 1993) and then obtain the coarser grids by removing every other point (e.g., see Zingg, 1991,1992). For non-integer grid refinement (coarsening), the same generating equations and parameters should be used. (See Chapters 5 and 6 for further discussion including non-structured grid refinement and structured refinement of non-structured grids.)

Error estimation using other unrelated grids, Category A.3, pose an interesting challenge. By "unrelated grids" we mean two or more grids (usually unstructured) that are overlapping but not simply obtained one from the other; say grid A is finer than grid B in some regions but coarser in others, such as might be obtained in two steps of an *r*-type (redistribution) solution-adaptive grid. It would seem that the two solutions on unrelated grids would provide the additional information necessary to estimate the uncertainty in either, but to our knowledge a method for doing so has not been invented.

In order to quantify the uncertainty with systematically refined (coarsened) grids, we need the convergence rate p to estimate the error. Initially, we will assume that p is known, i.e., that we are using a rigorously Verified code on a well behaved problem, and are now concerned with quantification of the uncertainty of a particular calculation using two grid solutions. The same methods can be converted to Verify a code, i.e., to Verify (or determine) p.

It will be convenient to present the techniques of using systematic grid convergence studies (i.e., grid refinement and/or coarsening) to estimate errors in the next chapter on the Grid Convergence Index, or GCI. The GCI includes not only the error estimation *per se* but also a recommended level of conservatism, i.e. a "factor of safety" or "error banding," which provides some uniformity in the reporting of the Quantification of Uncertainty. However, the techniques for systematic grid convergence studies covered in the next Chapter do not depend on the use of the GCI and can be used independently.

4.5 LEVELS OF SIMULATION USE

Simulations can be used at different levels, with corresponding appropriate error estimates. Benek et al (1996) noted three stages of the evolution of aerodynamic (CFD) applications, with increasing accuracy requirements:

1. provide diagnostic information,
2. supply incremental data,
3. generate baseline data for the performance model data base.

At Level 1, Benek et al noted that "CFD is an ideal diagnostic tool" because of its high information content and variety of data manipulation and presentation, useful for determining the origin of disturbances and flow instabilities, and location and strength of flow structures (e.g., vortices). They noted that the absolute accuracy requirement is not high for diagnostics, because the results will be confirmed by additional analyses and/or experimental simulation. The basic requirement is that the predominant physics be reasonably represented; "Validation concerns are minimal." [The accuracy requirement at Level 1 is basically qualitative.]

At Level 2, quantitative accuracy is required, but only for increments. For example, transonic flow about a wing both in a wind tunnel and in a free stream might be simulated with the expectation not of predicting lift and drag, but only of predicting the correction to experimental data from a wind tunnel to free stream conditions. Often [but not always], "highly accurate increments may be obtained from simulations that are less accurate than those required for base line data." Benek et al cited examples of "excellent agreement between measured and computed corrections" for transonic wall interference, model sting (support) interference, and mismatch between flight and wind tunnel Reynolds numbers. They noted the potential for "evaluation of increments that cannot be readily obtained experimentally," such as geometric compromises between wind tunnel models and flight models, and Reynolds number corrections for inlet swirl. Also, CFD results applied incrementally can be used [essentially as an elaborate interpolating function] for providing higher resolution to low spatial resolution experimental data, e.g. engine inlet pressure data. (They also noted that, in the world of aerodynamics, "the major stumbling block to Validation of wind tunnel correction methodology is that the majority of data is either classified or proprietary, and therefore, not generally available.")

At Level 3, quantitative accuracy is required for the absolute quantities, which is generally most demanding. However, the methodology for error estimation, the sources of additional information (above), etc., are the same, whether the focus is qualitative data, incremental data, or absolute base-line data. Even for absolute data, different requirements will occur, as various quantities will have various accuracy requirements and various sensitivities to grid resolution, etc. Thus the methods described herein are applicable to all three levels.

4.6 VERIFICATION OF COMPUTER ROUND-OFF ERRORS

Round-off error is sometimes confused with discretization error, and to emphasize the distinction people often speak of "computer round-off error" or "machine error" or "floating-point error." Round-off errors arise from the fact that computers do not work in the real number system, but only with a finite word-length subset of the real number system. This finite subset of floating-point numbers becomes an increasingly sparse (and therefore poorer) approximation to the complete real number line as the size of numbers increases. Different computers can have different representations of numbers, although this situation has vastly improved with the widespread adoption of IEEE standards. Even with identical

floating-point representations, different computers can have different hardware implementations of arithmetic operations; again, this situation has improved with the adoption of IEEE standards. Although arguably distinct from round-off errors, for practical user purposes we can also include in this category errors in computer system evaluation of primitive functions such as trigonometric functions, exponentials and logs, as well as higher level functions which are not the domain of the computer manufacturer but of local system libraries, such as Bessel functions, error functions, Gaussian distribution functions, etc.

The basic concept can be demonstrated by the following example. For real numbers, the result of A = 3 × (1/3) gives A = 1, but we cannot *generally* trust the computer implementation of these two operations (first, dividing 1 by 3, then multiplying the result by 3) to give A = 1; it may give something like A = 0.9999997. It may work for an isolated calculation for modern computers, but we cannot depend on it. Some compilers even give a warning if the code uses an IF test for equality of floating-point numbers, since these may not be meaningful; in the above example, the test "IF (A .EQ. 1.0) ..." fails. Likewise, especially when the operations are widely separated with intermediate calculations and storage (actualization of intermediate results) in various subprogram modules, we cannot depend on obtaining the same answer from different computers, nor even on the same computer with different languages, nor even on the same computer and same language but different compilers or simply different options on the same compiler. What we *can* insist is that the various representations be good approximations of the correct, real-number answer and therefore close to one another, "close" being a function of the computer word-lengths. However, elaborate sequences of calculations can produce significant deviations; it is not unusual for calculations performed in double precision to lose enough to floating-point errors to be reduced to single-precision accuracy in the end.

It is amazing that this simple, virtually kindergarten fact of computer life is not recognized by some managers and regulators, who want all calculations to be identical across computer systems. Other sources of discrepancy include variations in user-adjustable numerical parameters such as iteration relaxation factors, iteration convergence criteria, etc.

On the other hand, it is perhaps surprising that round-off errors *most* often are either not a problem at all, or are easily controlled to a acceptable level, in physical simulations of computational PDEs. The reason is the historical fact that computer hardware and system software have not evolved in isolation, but have co-evolved with simulation methods and applications.

Round-off errors in discretization *per se* (as opposed to calculations such as an equation of state evaluation that does not involve discretization) can be identified by grid convergence studies or other, often *ad hoc* approaches, but often they are simply demonstrated to not be significant. When they are significant, they manifest themselves as repeatable but *erratic* behavior as the grid is refined. (The behavior is not *chaotic* in the technical sense of *chaos*, *fractals*, etc., nor *random* in the technical sense of probability distributions, but just *erratic*.) It is not unusual to achieve iteration convergence to "machine zero," e.g. an identically steady state discrete solution, and to demonstrate grid convergence (of discretization error) to 5 or 6 figures, but to then encounter divergence of results as the grid is further refined due to round-off error. This is easy to demonstrate in 1-D, wherein extreme grid refinement is inexpensive. See Table 4.6.1.

Round-off error will become evident at more practical grid sizes when highly stretched grids are used, e.g. in turbulent boundary layer grids. Some algorithms are notoriously sensitive to round-off error, e.g. elliptic marching methods (Roache, 1995). Some physical problems are sensitive to and limited by round-off error, e.g. boundary layer transition calculations beginning from small initial perturbations cannot be carried through to full transition (Haynes et al, 1996). Often round-off errors can be isolated, or demonstrated to be not a problem, by changing word-length. Most modern engineering and scientific codes are designed to be compiled and run in double precision. In fact, most workstations actually run slightly *faster* in double precision than in single precision, being designed for the intended normal case of double precision calculations. This change from double to single precision, or vice versa, can sometimes be

N	Maximum Error	E_4
4	0.00022824765	0.058431
8	0.00001781814	0.072983
16	0.00000125631	0.082333
32	0.00000008359	0.087655
64	0.00000000539	0.090497
128	0.00000000034	0.091967
256	0.00000000002	0.092717
512	0.00000000000	0.093193
1024	0.00000000000	0.537369

Table 4.6.1. Demonstration of Round-Off Error in a Grid Convergence Study for Completed Richardson Extrapolation. Linear advection-diffusion equation with $Re = 1$. Grid spacing $\Delta = 1 / N$ and E_4 = max error / Δ^4. The approximate constancy (convergence) of E_4 up to $N = 512$ indicates the method is 4th-order accurate; the erratic deviation at $N = 1024$ is symptomatic of round-off error accumulation. (From Table IVa, Roache and Knupp, 1993)

accomplished by a simple compiler option on modern computers - otherwise, it can be a surprisingly difficult and clumsy test to implement. Another approach is to introduce *imitation* round-off error by additive or multiplicative pseudo-random number noise generation on selected quantities.

Round-off errors can in principle manifest themselves during Verification of a Code, but are more likely to rear their heads in application calculations because of (typically) more extreme parameter variations and solution scales. Therefore, Verification for round-off error effects is best considered as part of Verification of a Calculation (here in Chapters 4 and 5) rather than Verification of a Code (Chapter 2).

A impressive test of an iterative solver is to demonstrate that it can achieve iteration convergence to machine zero, i.e. to reach a point on the iterative path at which there are identically zero changes in all variables. Also, it can be a valuable debugging exercise on a coarse grid, and Oberkampf et al (1995) stated that "it is highly recommended that this iteration convergence [to machine zero] be demonstrated on coarse grid solutions." However, in my opinion, it is not a necessary requirement for a code or an algorithm, as some authors suggest. Reaching machine zero in an iteration is *itself* an artifact of round-off error; a real number implementation would approach real zero only asymptotically. The fact that a particular code/algorithm actually reaches machine zero is accidental, since the noise of finite word-length arithmetic means that iterative changes near the limit of precision do not follow the real number arithmetic rules on which the iteration algorithm is based. Experience suggests that the simplest iteration algorithms (e.g., direct substitution iteration) are more likely to converge to machine zero, while more complicated algorithms (with more non-real-number arithmetic operations and intermediate storage) will often produce persistent, erratic fluctuations in the last one or two decimal significant figures of the variables. This should not disqualify these algorithms and codes, especially in view of the fact that the more complicated algorithms are often much more efficient than the simple algorithms.

4.7 Δ EFFECT OF DIFFERING FORMULATIONS

One sometimes reads in the literature and in advertisements for commercial codes that different simulation results are to be expected from different formulations, e.g. different discretization methods (e.g., FEM vs. FDM vs. FVM), or different but mathematically equivalent forms of the continuum equations

(e.g., incompressible flow formulated in terms of primitive velocity-pressure variables, vs. vorticity-stream function variables, vs. vorticity-velocity variables), or different equation solvers (direct Gaussian elimination vs. SOR vs. PCG vs. multigrid). Clearly, every different formulation will affect numerical accuracy, because of differences in discretization error, incomplete iteration error, and round-off error, so the answers will not be exactly identical. However, once numerical accuracy is quantified, the error bands from different simulations *must* overlap. If they do not, then a counter-example to the Verification of Calculations has been produced. *Either* Calculation A and its uncertainty estimate (say, a second-order FEM simulation using velocity - pressure variables with 100 × 100 quadrilateral elements and a direct solver), *or* Calculation B (say, a 4th-order vorticity-velocity simulation using 200 × 200 cells and a multigrid solver), *or* the comparison exercise itself is wrong. It should not be necessary to state this, but it apparently is.

Numerical methods are not part of the continuum parameters!

Similarly, the particular grid or mesh resolution and type should be irrelevant, within the uncertainty estimates. While most practitioners recognize that mesh density should not affect the answer within the uncertainty estimates (as shown by the common use of the term slightly abusive term "grid independence"), confusion often occurs in regard to grid types. For example, an airfoil computation may be approached with a variety of grid configurations: C-type grids, H-type grids, unstructured meshes with a variety of geometric elements (quadrilaterals, triangles) or even mesh-less methods. The choice will affect how difficult it is to achieve accuracy, especially around critical areas like the airfoil trailing edge. One grid type may be clearly superior to another for a particular problem and a particular area of concern. But *all* legitimate grid types must be capable of producing the correct solution given sufficient computer resources. If a poor O-grid is much worse for an airfoil than a good C-grid, then the O-grid will require many more cells to achieve an accurate answer. But if the uncertainty estimate of the O-grid solution is comparable to that of the C-grid, it should give comparable answers. That is, if a grid type is not capable of reaching acceptable accuracy with available computer resources, this should be reflected in the error/uncertainty estimates obtained. The computational solution on *any* legitimate grid should not lie.

To expand the previous emphatic statement,

Neither numerical methods (FEM, FDM, FVM, spectral, ...)
nor continuum formulations (primitives, vorticity, compressible flow equations, ...)
nor grid types (structured, unstructured, O-grids, C-grids, block structured, grid-free, ...)
nor grid resolutions (100 × 100 × 100, 1000 × 1000 × 1000, ...)
are part of the continuum parameters!

CHAPTER 5

SYSTEMATIC GRID CONVERGENCE STUDIES AND THE GRID CONVERGENCE INDEX (GCI)

"If you want a new idea, read an old book."

Anonymous

5.1 Δ INTRODUCTION

This chapter is taken primarily from Roache (1994), " Perspective: A Method For Uniform Reporting Of Grid Refinement Studies." Additional material is added from Roache (1997), "Quantification of Uncertainty in CFD," Roache (1995a), "Verification of Codes and Calculations," Westerink and Roache (1995), "Issues in Convergence Studies in Geophysical Flow Computations," and Roache (2003a), "Error Bars for CFD." The chapter organization repeats some of the material in previous chapters but is used here to make the chapter more nearly self contained, and because the material deserves repetition, especially since it is applicable to commercial codes.

The only type of uncertainty considered in this chapter is the commonly used $U_{95\%}$, i.e. an "expanded" uncertainty (V&V20) targeted to bracket 95% of the data from the parent population. Consideration of standard uncertainty u and other levels of expanded uncertainty is deferred until Chapter 11.

In (Roache, 1994) I proposed the use of a Grid Convergence Index (GCI) for the uniform reporting of grid convergence studies in Computational Fluid Dynamics and related disciplines. Since then, it has been confirmed in many hundreds of applications. The (slightly modified) method has been approved for all papers submitted to the *ASME Journal of Fluids Engineering* (Celik et al, 2008). The method provides an objective asymptotic approach to quantification of uncertainty of grid convergence. The basic idea is to approximately relate the results from any grid convergence test to the expected results from a grid doubling

using a 2nd-order method. The GCI is based upon a grid convergence error estimator derived from the theory of generalized Richardson Extrapolation. It is recommended for use whether or not Richardson Extrapolation is actually used to improve the accuracy. A different form of the GCI applies to reporting coarse grid solutions when the GCI is evaluated from a "nearby" problem. The simple formulas may be applied *a posteriori* by editors and reviewers, even if authors are reluctant to do so.

The generalized Richardson Extrapolation described can be used independently of the GCI, for either Verification of Calculations or Verification of Codes. Likewise, as will become clear, the GCI is recommended for use in uniform reporting of grid convergence tests even if the conditions for the theory do not hold strictly.

5.2 BACKGROUND ON GRID CONVERGENCE REPORTING

There are other possible techniques for the quantification of numerical uncertainty, but systematic grid convergence studies are the most common, most straightforward and arguably the most reliable. The motivation for development of the uniform Grid Convergence Index was the inconsistent and confusing reporting of grid convergence studies in the engineering and scientific literature. The following hypothetical examples will suffice to illustrate the confusing reporting.

One paper states that the grid density was increased by 50%, resulting in a difference in some solution norm of 4% (of the fine grid solution) using a 1st-order accurate method. In another paper, grid density was doubled, resulting in a difference of 6%, using a 2nd-order method.

Which fine grid solution is more reliable?, i.e. better converged? More importantly, can the reader have any reasonable expectation that these numbers represent a "% accuracy error band," i.e. that the fine-grid calculations are probably accurate to within 4% or 6% of the true solution of the continuum equations?

Note that we are concerned herein with Verification of a particular Calculation, i.e. estimating and banding the grid convergence accuracy of a particular discretized solution. We assume that the code itself has already been Verified for the same class of problems, so that coding errors are not an issue and, if done properly, the order of accuracy has been Verified for well behaved problems (see Chapter 3). Nor are we concerned with code Validation, e.g. that a turbulence model is adequate. (See Chapter 2 for terminology.) Also, it is worth repeating that this approach (and similar grid convergence studies) address only "ordered" discretization errors, which by definition vanish as grid spacing Δ or $h \to 0$. Specifically, the errors introduced by the use of far-field computational boundaries must be assessed separately (see Section 6.10; also Chapters 2 and 6 of Roache, 1998b). Further, I consider herein only *a posteriori* error estimation, being of the opinion that useful *a priori* estimation is not possible for non-trivial problems.

5.3 RICHARDSON EXTRAPOLATION

Richardson Extrapolation, also known as "h^2 extrapolation" and "the deferred approach to the limit" and "iterated extrapolation," was first used by Richardson in 1910 (in a structures problem of determining the stresses in a masonry dam) and later embellished in 1927. The discrete solutions f are assumed to have a series representation in the grid spacing h (or Δ) of

$$f = f_{exact} + g_1 h + g_2 h^2 + g_3 h^3 + \cdots \tag{5.3.1}$$

The functions g_1 , g_2 , etc. are defined in the continuum and do not depend on any discretization. For infinitely differentiable solutions, they are related to all orders to the solution derivatives through the elementary Taylor series expansions, but this is not a necessary assumption for Richardson Extrapolation,

nor is the infinite series indicated in Eq. (5.3.1). It is only necessary that Eq. (5.3.1) be a valid definition for the order of the discretization. Thus, the extrapolation may be valid for finite volume solutions, finite element solutions, etc.

For a 2nd-order method, $g_1 = 0$. Then the idea is to combine two separate discrete solutions f_1 and f_2, on two different grids with (uniform) discrete spacings of h_1 fine grid) and h_2 (coarse grid), so as to eliminate the leading order error terms in the assumed error expansion, i.e. to solve for g_2 at the grid points in Eq. (5.3.1), substitute this into Eq. (5.3.1) and obtain a more accurate estimate of f_{exact}. The result is the original statement (Richardson, 1927) for h^2 extrapolation.

$$f_{exact} = \frac{h_2^2 f_1 - h_1^2 f_2}{h_2^2 - h_1^2} + \text{H.O.T.} \tag{5.3.2}$$

where H.O.T. are higher order terms. Using the grid refinement ratio $r = h_2 / h_1$ (defined to be > 1), this can be conveniently expressed in terms of a correction to the fine grid solution f_1, dropping H.O.T.

$$f_{exact} \cong f_1 + \frac{f_1 - f_2}{r^2 - 1} \tag{5.3.3}$$

The most common use of this method is with a grid doubling, or halving. (These are *identical*. Both use two grids, one twice as fine as the other, i.e. we have a coarse grid and a fine grid. Whether we "doubled" or "halved" just depends on which calculation came first.) With
$r = 2$, Eq. (5.3.3) becomes

$$f_{exact} \cong 4/3\, f_1 - 1/3\, f_2 \tag{5.3.4}$$

It is often stated that Eq. (5.3.4) is 4th-order accurate if f_1 and f_2 are 2nd-order accurate. Actually, as known by Richardson, this is true only if odd powers are absent in the expansion (5.3.1), which he achieved by assuming the exclusive use of 2nd-order centered differences. If uncentered differences are used, e.g. upstream weighting of advection terms, even if these are 2nd-order accurate (3-point upstream), the h^2 extrapolation is 3rd-order accurate, not 4th. As a practical limitation, even extrapolations based on centered differences do not display the anticipated 4th-order accuracy until the cell Reynolds number *Rc* is reduced; for the one-dimensional advection-diffusion equation with Dirichlet boundary conditions, $Rc < 3$ is required (Roache and Knupp, 1993; see also Chapter 3).

Although Richardson Extrapolation is most commonly applied to grid doubling, and is often stated to be *only* applicable to integer grid refinement (e.g., Conte and DeBoor, 1965), grid doubling is not required. In order to use Eq. (5.3.3) it is necessary to have values of f_1 and f_2 at the same points, which would seem to require commonality of the discrete solutions, and therefore integer grid refinement ratios r (grid doubling/halving, tripling, etc.). However, even in his 1910 paper, Richardson looked forward to defining a continuum f_2 by higher order interpolation, and in the 1927 paper he had a specific approach worked out. Ferziger (1993) alluded to this approach with less detail but more generality. Similarly, Richardson Extrapolation is commonly applied only to obtaining a higher-order estimate on the coarse grid with $h_2 = 2h_1$, but Roache and Knupp (1993) showed how to obtain 4th-order accuracy on *all* fine-grid points by simple 2nd-order interpolation, not of the solution values f_2, but of the extrapolated *correction* from Eq. (5.3.4), i.e. by 2nd-order interpolation of 1/3 ($f_1 - f_2$). The use of simple 2nd-order interpolation avoids complexities with non-uniform grids and near-boundary points. S. A. Richards (1997) extended the method

for "Completed Richardson Extrapolation" to space and time, including possible mixed space-time order of accuracy and more generality than grid doubling.

Richardson (1910,1927) also considered 6th-order extrapolation (using 3 grid solutions to eliminate g_2 and g_4), parabolic and elliptic equations, staggered grids (then called "interpenetrating lattices"), rapid oscillations and the $2h$ wavelength limit (the "cell Reynolds number" limit for "wiggles" in CFD; see Roache, 1998b), *a priori* error estimates, singularities, integral equations, statistical problems, Fourier coefficients, and other non-calculus problems. For example, Richardson (1927) showed the power of the method in an elegant example of extrapolating two very crude approximations to a circle, namely an inscribed square and an inscribed hexagon, to get an estimate of π with 3-figure accuracy, without using any trigonometry. (This calculation will be given later in Section 5.5.)

The usual assumptions of smoothness apply, as well as the assumption (often verified) that the local error order is indicative of the global error order. The extrapolation must be used with considerable caution, since it involves the additional assumption of monotone truncation error convergence in the mesh spacing h. This assumption may not be valid for coarse grids, or possibly other conditions. The discretization method must be 2nd order everywhere, including boundaries; for example, Gourlay and Morris (1968) showed how to successfully apply Richardson extrapolation in nonlinear hyperbolic systems with over-determined boundary information, and gave references to earlier work on Richardson Extrapolation involving starting values in multi-step ODE solvers. Also, the extrapolation magnifies machine round-off errors and incomplete iteration errors (Roache, 1998b). In spite of these caveats, the method is extremely convenient to use compared to forming and solving direct 4th-order discretizations, which involve more complicated stencils, wider bandwidth matrices, special considerations for near-boundary points and non-Dirichlet boundary conditions, additional stability analyses, etc., especially in non-orthogonal coordinates which generate cross-derivative terms and generally complicated equations. Such an application was given by the present author in Roache (1982); see also Chapter 6.

The method is in fact oblivious to the equations being discretized and to the dimensionality of the problem, and can easily be applied as a *postprocessor* (Roache, 1982) to solutions on two grids with no reference to the codes, algorithms or governing equations which produced the solutions, as long as the original solutions are indeed "2nd-order accurate" or more generally, "p-th order accurate." We use the common but somewhat abusive terminology of "2nd-order accurate solution" to mean a solution obtained by a Verified 2nd-order accurate method applied in the asymptotic range of grid spacing. Even the expression "2nd-order accurate method" itself can be misleading by unrealistically raising expectations that the method will be 2nd-order accurate applied to any problem. A clumsy but more precise description is "a method whose leading order term in the Taylor series or other convergence analysis is 2nd order."

The difference between the 2nd-order solution and the extrapolated 4th-order solution is itself a useful diagnostic tool, obviously being an error estimator (although it does not provide a true *bound* on the error). It was used very carefully, with an experimental determination rather than an assumption of the *local* order of convergence, by de Vahl Davis (1983) in his classic Benchmark study of a model free convection problem. See Nguyen and Maclaine-Cross (1988) for application to heat exchanger pressure drop coefficients. Zingg (1991,1992) applied the Richardson error estimator to airfoil lift and drag calculations in body-fitted grids. Also, his data indicate that Richardson Extrapolation can be applied to the estimation of far-field boundary errors, with the error being 1st-order in the inverse of distance to the boundary. (See Chapter 6, Section 6.10.) Blottner (1990) used Richardson Extrapolation to estimate effects of artificial dissipation terms in hypersonic flow calculations. (See Chapter 6, Section 6.11.) See other early applications by Anderson and Batina (1988), Caruso et al (1985), Kessler et al (1988), Williams (1989).

An important aspect of Richardson Extrapolation is that it applies not only to point-by-point solution values, but also to solution functionals, e.g., lift coefficient C_L for an aerodynamics problem or integrated discharge for a groundwater flow problem, provided that consistent or higher-order methods are used in the

evaluations (e.g., 2nd- or higher order quadratures for lift) as well as the basic assumption that the order of the method applies globally as well as locally. If Richardson Extrapolation is applied to produce (say) 4th-order accurate grid values, one could in principle calculate a 4th-order accurate functional like C_L from the grid values, but it would require careful implementation of 4th-order accurate quadratures. It is much simpler to apply the extrapolation directly to the C_L's obtained in each grid, requiring only 2nd-order quadratures. Indeed, this is a major attraction of Richardson Extrapolation compared to using 4th-order accurate stencils solved either directly or by deferred corrections. (Note, however, that the two approaches yield different answers, although both are 4th-order accurate if done properly.)

A very significant yet often overlooked *disadvantage* of Richardson Extrapolation is that the extrapolated solution generally is not "conservative" in the sense of maintaining conservation properties (e.g., Roache, 1998b). This could well dictate that Richardson Extrapolation not be used to obtain an extrapolated or "corrected" solution. For example, if it were used on the ground-water flow simulations for the Waste Isolation Pilot Plant (WIPP PA Dept., 1992; Helton et al, 1995,1996), it would be "more accurate" in some metric, but would introduce additional non-conservative (i.e., lack of conservation property) source terms into the radionuclide transport equation. It is also noteworthy that Richardson (1927) pointed out that the accuracy of the extrapolation does not apply to arbitrarily high derivatives of the solution. The extrapolation can introduce noise to the solution which, although low level, may decrease the accuracy of the solution higher derivatives.

Thus, it is not advocated here that Richardson Extrapolation necessarily be used to improve the reported solution, i.e. to produce an extrapolated or corrected solution, since that decision involves these considerations and possibly others. What is advocated is that, regardless of whether Richardson Extrapolation is used to improve the solution, it can be used to estimate the discretization error, and the Grid Convergence Index (defined herein and based on the generalized theory of Richardson Extrapolation) can be used to uniformly *report* grid convergence studies.

5.4 Δ GENERALIZATION OF RICHARDSON EXTRAPOLATION

Without assuming the absence of odd powers in the expansion of Eq. (5.3.1), we can generalize the usual Richardson Extrapolation (5.3.4) to p-th order methods and r-value of grid ratio, again eliminating the leading term in the error expansion, as follows.

$$f_{exact} \cong f_1 + \frac{f_1 - f_2}{r^p - 1} \tag{5.4.1}$$

If the next term in the series of Eq. (5.3.1) is zero, e.g. if centered differences were used, then the extrapolation is $(p + 2)$ order accurate. But generally, and notably if upstream-weighted methods for advection have been used, the extrapolation is $(p + 1)$-order accurate.

It may easily be verified that Eq. (5.4.1) is valid for multidimensions in any coordinates, including space and time, provided that the same grid refinement ratio r is applied, and the order p is uniform, in all space and time directions.

In Eq. (5.4.1), the correction to the fine grid solution f_1 obviously provides an error estimator of the fine grid solution. (The error of any estimate is defined as the estimated value - true value, so the error estimate for f_1 is the negative of the correction term.) Expressing this as an **E**stimated fractional error E_1 for the fine grid solution f_1, we have

$$E_1[\text{fine grid}] = \frac{\varepsilon}{r^p - 1} \tag{5.4.2}$$

$$\varepsilon = \frac{f_2 - f_1}{f_1} \tag{5.4.3a}$$

Alternately, we could express this as an **E**stimated *dimensional* error[28] using the dimensional form

$$\varepsilon = f_2 - f_1 \tag{5.4.3b}$$

Defining the **A**ctual fractional error A_1 of the fine-grid solution as usual,

$$A_1[\text{fine grid}] = \frac{f_1 - f_{exact}}{f_{exact}} \tag{5.4.4}$$

and using Eqs. (5.4.1 - 4) and the binomial expansion gives

$$A_1[\,fine\ grid\,] = E_1 + O(h^{p+m}, E_1^2) \tag{5.4.5}$$

where m = 1 generally or m = 2 if centered differences have been used. Thus, E_1 is an *ordered error estimator*, i.e. an ordered approximation to the actual fractional error of the fine grid solution. E_1 is a good approximation when the solution is of reasonable accuracy, i.e. when $|E_1| << 1$.

This is generally *not* true of ε in Eq. (5.4.3), which is the quantity commonly reported in past grid convergence studies. That is, ε is not always an error estimator since it does not take into account r or p. For $r < 2$ and $p = 1$, ε alone is optimistic, under-estimating the grid convergence error compared to E_1 (by a factor of 2 for r = 1.5). For $r = 2$ and $p = 2$, ε alone is conservative, over-estimating the grid convergence error compared to E_1 (by a factor of 3). Note that |ε| can be made (almost) arbitrarily small, just by choosing $r \cong 1$. [The only restriction is that r is limited by the integer character of the number of grid points, so the smallest possible $r = N/(N-1)$ where N is the number of grid points in each direction of the fine grid.] This is analogous to the situation wherein an arbitrarily small tolerance on *iteration* convergence can always be met by using an arbitrarily small relaxation factor, belying the adequacy of such an iteration convergence criterion (Roache, 1998b; Ferziger, 1993, Ferziger and Peric, 1996).

E_1 may of course be expressed as a relative error, either a fraction or a % of some normalizing value. If the normalizing value is taken as the local discrete value f_1 itself or (if known) the local (or global) exact value f_{exact}, then, like any *relative* error indicator, it will become meaningless when f_1 or f_{exact} is zero or small relative to $(f_2 - f_1)$. In such cases, the denominator of Eq. (5.4.3a) should be replaced with some suitable normalizing value for the problem at hand, as would the usual definition of actual relative error A_1 in Eq. (5.4.4).

As described earlier, one may chose to not use Richardson Extrapolation to produce an extrapolated or corrected solutions for good reasons, e.g., due to concern over the actual order of the method, or accumulation of round-off error, or incomplete iteration convergence error, or uncertainty that the

[28] Non-dimensionalizing can cause confusion, but is convenient to use because people like to think intuitively in terms of % errors. Obviously, these become misleading or meaningless if the divisor => 0. Also, it is not recommended to non-dimensionalize by the extrapolated solution since this will often introduce more noise in the calculation.

asymptotic range has been reached, or lack of the conservation property in the extrapolated results, etc. But whether or not one chooses to use or report the extrapolated solution, one can still use the theoretical basis to *consistently* report the results of the grid convergence study.

5.5 RICHARDSON'S EXTRAPOLATION FOR π

Richardson (1927) showed the power of his extrapolation method in an entertaining example of extrapolating two very crude approximations to a circle, namely an inscribed square and an inscribed hexagon, to get an estimate of π with 3-figure accuracy, without using any trigonometry. This exemplary calculation, repeated here, shows the generality of the concept of "grid spacing" and "discretization measure" h or Δ.

We use the fundamental definition of π, the ratio of the circumference of the circle to its diameter, and consider the circle of unit radius to be the limit as $N \to \infty$ of the same ratio defined for a regular inscribed N-sided polygon. Call that ratio Π_I where I indicates "inscribed" polygon. For $N = 4$, $\Pi_{I,4} = 2\sqrt{2}$. For $N = 6$, each of the 6 triangles is equilateral, and (without using trigonometry) we get $\Pi_{I,6} = 3$ exactly. At this point, lacking trigonometry, we have to just assume that convergence is quadratic in the discretization measure $\Delta = 1 / N$, i.e. $p = 2$. [Trigonometry would show that this is indeed true; the exact formula for $\Pi_{I,N} = N \sin \pi / N$ has the series expansion $= \pi - 2/3\ \pi^3 \Delta^2 + O(\Delta^4)$. See also Ferziger, 1981.] In terms of the grid discretization ratio r, for Eq. (5.4.1), with coarse "grid" $N = 4$ and fine "grid" $N = 6$, we have $r = (1/4)/(1/6) = 3/2$, so $r^2 = 9/4$, and $1/(r^2 - 1) = 4/5$. Applying this to Eq. (5.4.1), we obtain the extrapolated estimate $\Pi_I = 3 + (4/5)\ (3 - 2\sqrt{2}) = 3.1372583$, which is only 0.138% low, or correct to better than 3 figures significant figures.

Richardson noted that this error is only 1/33 of the error of the $N = 6$ estimate. To obtain as good a result without extrapolation would require $N = 35$, and you would have to know some trigonometry and do *much* more calculating.

[The result for the *circumscribed* polygons is $\Pi_C = 2\sqrt{3} + (4/5)\ (2\sqrt{3} - 4) = 3.0353829$, which is 3.381% low, or correct to only 1 significant figure. The problem is that the leading error in the series expansion for the *tan* required for circumscribed polygons is twice as large as that for *sin* used for the inscribed polygons.]

5.6 GRID CONVERGENCE INDEX FOR THE FINE GRID SOLUTION

Although the error estimator E_1 of Eq. (5.4.2) is based on a rational and consistent theory, it is certainly not a *bound* on the error. Nor is a reliable and practically tight bound on solution error for nontrivial and nonlinear problems likely to be forthcoming, in our opinion. What is generally sought in engineering calculations is not a true "error bound" but just an "error band," i.e. a tolerance on the accuracy of the solution which may in fact be exceeded, but in which the reader/user can have some practical level of confidence. The error estimator E_1 itself does not provide a very good confidence interval. One might expect that it is equally probable that E_1 be optimistic as conservative, i.e. it is just as likely that the actual error A_1 be greater than E_1 as less than E_1. This would correspond roughly to a 50% confidence band[29]. A well-founded probability statement on the error estimate, such as a statistician would prefer (e.g., a 2σ limit) is not likely forthcoming for practical PDE problems. However, based on cumulative experience

[29] For further discussion see Section 5.14.2

in the CFD community, at least a practical confidence level exists for the ε of Eq. (5.4.3) obtained using a grid doubling and a verified 2nd-order accuracy code.

That is, for a grid doubling with a 2nd-order method, and some indication that the calculations are within the asymptotic range of convergence, most practitioners would accept the ε of Eq. (5.4.3) as a reasonable error band, in the flavor of a statistician's 2σ range or an experimentalist's 20:1 odds (Kline and McClinstock, 1953). An ε of (say) 6% would be taken to indicate (not absolutely, but with reasonable confidence) that the fine grid solution was within 6% of the asymptotic answer. This confidence is well justified by the theory of Richardson Extrapolation, which shows, from Eqs. (5.4.2 - 3) with $r = 2$ and $p = 2$, that the error estimate E_1 is only 1/3 of this error band, or 2%.

The idea behind the Grid Convergence Index is to approximately relate the ε of Eq. (5.4.3) obtained by whatever grid convergence study is performed (whatever p and r) to the ε that would be expected from a grid convergence study of the same problem with the same fine grid using $p = 2$ and $r = 2$, i.e. a grid doubling with a 2nd-order method. The relation is based on equality of the error estimates. Given an ε from an actual grid convergence test, the GCI is derived by calculating the error estimate E_1 from Eqs. (5.4.2 - 3), then calculating an equivalent ε that would produce approximately the same E_1 with $p = 2$ and $r = 2$. The absolute value of that equivalent ε is the (relative error) Grid Convergence Index for the fine grid solution, which is conveniently expressed as

$$\mathrm{GCI}\left[\text{fine grid}\right] = F_s \frac{|\varepsilon|}{r^p - 1} \quad , \quad F_s = 3 \tag{5.6.1}$$

where ε is defined in Eq. (5.4.3-a or -b).

Obviously, if the denominator of Eq. (5.4.3a) is small, ε should be normalized by some other characteristic quantity for the calculation (as discussed earlier) or alternately it can be evaluated as an absolute (rather than relative) quantity (i.e., without the division by f_1 or any normalizing value) as in Eq. (5.4.3b), in which case Eq. (5.6.1) produces an "absolute error" GCI.[30]

We note immediately that for a grid doubling ($r = 2$) with a 2nd-order method ($p = 2$), the denominator = 3, and we obtain GCI = |ε|, as intended. For any r and p, if the coefficient F_s is chosen as $F_s = 1$, then the GCI = E_1. Thus F_s may be interpreted[31] as a "factor of safety" over the Richardson Error Estimator E_1.

The purpose of the GCI is not to preclude more convincing grid convergence tests, such as using Richardson Extrapolation over several grid refinements. The modest purpose herein is just to get minimal two-calculation grid convergence exercises onto a uniform reporting basis.

The GCI, like the theory of Richardson Extrapolation on which it is based, is equally applicable not only to grid values, but also to solution functionals (e.g., C_L) and to plotted curves, wherein ε may be read visually or calculated from interpolated tabular values. Thus it may be used to produce plots of the estimated error band about a fine grid solution by post-processing the results of any two grid solutions. Non-physical oscillations in the solutions ("wiggles," e.g. see Roache, 1998b) are of course a cue that the solutions are not in the asymptotic range, Richardson Extrapolation is not accurate, E_1 of Eq. (5.4.2) is not a valid error estimator, and confidence in the GCI as an error band is not justifiable.

Applying Eq. (5.6.1) to the hypothetical cases in the second paragraph of Section 5.2, we see that a 4% difference from a grid refined by 50% using a 1st-order method gives a fine-grid GCI = 24%, whereas a 6% difference from a doubled grid using a 2nd-order method gives a fine-grid GCI = 6%. Even though the

[30] Several papers have confused discussion of relative merits of extrapolation methods by focusing on differences in the normalizing term for ε, which is clearly a non-essential aspect.

[31] As suggested to me by Prof. J. Westerink.

first paper's reported raw deviation ε from coarse to fine grid calculations might appear at first glance to be better than that reported in the second paper (4% compared to 6%), it is in fact not nearly as well converged (24% compared to 6%), as indicated by the Grid Convergence Index. (The ratio does not depend on the value of the "factor of safety" F_s used in Eq. (5.6.1), but would also apply if the Richardson Error Estimator E_1 were used, i.e. with $F_s = 1$.)

For a less hypothetical example, consider the grid convergence results reported by the present author (Roache, 1982; Chapter 16 of Roache, 1998b) for Benchmark calculations of weakly separated flows obtained using Richardson Extrapolation applied with grid doubling. The reported quantification of convergence was the maximum fractional deviation ε_4 between the fine-grid 2nd-order and the extrapolated 4th-order solution f_4 from Eq. (5.3.4),

$$\varepsilon_4 = \frac{f_1 - f_4}{f_4} \tag{5.6.2}$$

The values reported were $\varepsilon_4 = 0.17\%$ for wall vorticity and 0.13% for a velocity profile at a longitudinal station traversing the separation bubble. This ε_4 is easily related to the ε of Eq. (5.4.3); combining Eqs. (5.3.4), (5.4.3) and (5.6.2) shows (for $r = 2$, $p = 2$, $F_s = 3$)

$$\varepsilon = 3\varepsilon_4 + O(\varepsilon_4^2) \tag{5.6.3a}$$

$$GCI[\,fine\ grid\,] = 3\,|\,\varepsilon_4\,| \tag{5.6.3b}$$

The reported grid convergence criteria (Roache, 1982) of 0.17% for wall vorticity and 0.13% for velocity *for the fine grid 2nd-order solution* would now be replaced by the much more conservative GCI [fine grid] = 0.51% and 0.39%.

On the other extreme, it is recognized that ostensibly 2nd-order algorithms may fail to attain 2nd-order performance in a particular calculation, due to coding quirks or errors, subtleties in nonlinear problems, overly strong grid stretching, failure to attain the asymptotic range, etc.[32] Unless the analyst has convincingly Verified that the code actually attains the theoretical order, at least on a nearby problem, the more conservative value of $F_s = 3$ should be used in reporting the GCI in Eq. (5.6.1).

Two calculations of the same problem with the same value for GCI, say a 1st-order calculation on a finer grid and a 2nd-order calculation on a coarser grid, are not quite indifferent as to the uncertainty of the calculations. The GCI of the 1st-order calculations is based on an only 2nd-order accurate error estimator, whereas the GCI of the 2nd-order calculations is based on a 3rd- or 4th-order accurate error estimator. Thus, even with the same GCI, the 2nd-order calculations are more reliable, i.e. have less uncertainty (in their uncertainty estimates) than the 1st-order calculations. Generally, it is more difficult to judge grid convergence for 1st-order methods; e.g., see Leonard and Drummond (1995).

5.6.1 Δ Grid Convergence Index for the Extrapolated Solution

In cases wherein Richardson Extrapolation is actually used to produce a higher order accurate solution (the extrapolated or corrected solution) rather than just to estimate the error of the 2nd-order fine grid solution, the GCI of Eq. (5.6.1) (or 5.6.3b) appears to be unfairly conservative. The solution used is the (say) 4th-order accurate solution, but the reported GCI would be the same even if only the 2nd-order

[32] See, e.g., de Vahl Davis, 1983; Steinberg and Roache, 1985; Shirazi and Truman, 1989; Roache et al, 1990; Westerink and Roache, 1995. See also Chapter 6.

accurate fine grid solution were used. That is, E_1 and GCI are respectively the Richardson Error Estimator and Grid Convergence Index for the fine grid 2nd-order solution, not for the 4th-order solution. Although we expect the extrapolated solution to be more accurate than the 2nd-order fine grid solution, we would need additional information (a solution on a third grid) to rationally estimate the error of the extrapolated solution itself. Such a third grid solution could be used in principle (possibly not in practice, for difficult nonlinear problems) to extrapolate a 6th-order accurate solution. The error estimate (and therefore the GCI) will always lag the best solution estimate. This is quite conservative when the conditions for validity of Richardson Extrapolation have been convincingly demonstrated by numerical experiments[33]. A heuristic extension for such situations (Roache, 1994) is to report the GCI for the extrapolated solution based on Eq. (5.6.1) with ε replaced by ε_4 from Eq. (5.6.2), giving (for $r = 2$, $p = 2$, $F_s = 3$)

$$GCI[\textit{extrapolated solution}] \cong | E_1[\textit{fine grid}] | = | \varepsilon | / 3 \qquad (5.6.1.1)$$

This agrees with the reported grid convergence criteria (Roache, 1982) of 0.17% for wall vorticity and 0.13% for velocity.

An alternative heuristic uncertainty estimate for the extrapolated solution is given in the ITTC Manual (ITTC, 2002), Eq. (22), transcribed here as

$$GCI[\textit{extrapolated solution}] \cong (F_s - 1) | E_1[\textit{fine grid}] | = | (F_s - 1) | \varepsilon | \qquad (5.6.1.2)$$

with no limitation stated for F_s but only $F_s = 1.25$ seems appropriate and approximately consistent with Eq. (5.6.1.1), producing $|\varepsilon|/4$ compared to $|\varepsilon|/3$ for Eq. (5.6.1.1). As noted, these heuristic uncertainty estimates for the extrapolated solution cannot be as well founded as those for the fine grid solution. Eça et al (2004) confirmed, using a non-trivial exact analytical benchmark solution, that the uncertainty estimation for the extrapolated solution is less reliable than that for the fine grid solution.

5.7 GRID CONVERGENCE INDEX FOR THE COARSE GRID SOLUTION

Seemingly, if we have a fine grid and a coarse grid solution, we would be expected to use the fine grid solution, so reporting of the above fine-grid GCI of Eq. 5.6.1 would apply. However, a practical scenario occurs for which the contrary situation applies, i.e. we use the coarse grid solution.

Consider a parametric study in which hundreds or thousands of variations are to be run. For example, consider a three-dimensional time-dependent study of dynamic stall, with perhaps 3 Mach numbers, 6 Reynolds numbers, 6 airfoil thickness ratios, 3 rotor tip designs, and 2 turbulence models: a total of 648 combinations. For another, consider the Monte Carlo type study of groundwater transport with thousands of runs (WIPP PA Dept., 1992; Helton et al, 1995,1996). A scrupulous approach would require a grid convergence study for each case, but most engineers would be satisfied with one or a few good grid convergence tests, expecting, e.g., that a grid adequate for a NACA 0012 airfoil could be assumed to be adequate for a NACA 0015 airfoil. (In fact, this is often not justified by experience, e.g. stall characteristics can be quite sensitive to thickness ratio.) So for the bulk of the stack of calculations, we would be using the coarse grid solution, and we want a Grid Convergence Index for it. That is, we derive the (relative error) GCI from Eq. (5.4.1), not as the correction to the fine grid solution f_1, but as the correction to the coarse grid solution f_2. In this case, the error estimate changes and *must be less optimistic*.

[33] E.g. Roache, 1982; Shirazi and Truman, 1989; Blottner, 1990; Roache and Knupp, 1993.

	Fine Grid GCI				Coarse Grid GCI		
p	*r* = 2	1.5	1.1	*p*	*r* = 2	1.5	1.1
1	3.00%	6.00%	30.00%	1	6.00%	9.00%	33.00%
2	1.00%	2.40%	14.29%	2	4.00%	5.40%	17.29%
3	0.43%	1.26%	9.06%	3	3.43%	4.26%	12.06%
4	0.20%	0.74%	6.46%	4	3.20%	3.74%	9.46%

Table 5.7.1. Grid Convergence Index (GCI) with $F_s = 3$ calculated from Equations 5.6.1 and 5.7.2 for common values of grid ratios (*r*) and orders of the basic numerical method (*p*), for both coarse grid solutions and fine grid solutions, normalized to ε = 1%.

$$f_{exact} \cong f_2 + (f_1 - f_2)\, r^p / (r^p - 1) \tag{5.7.1}$$

The coarse-grid GCI is then

$$GCI\,[coarse\ grid\,] = F_s\,|\varepsilon|\, r^p / (r^p - 1),\ \ F_s = 3 \tag{5.7.2}$$

$$GCI[coarse\ grid] = r^p GCI[\,fine\ grid\,] \tag{5.7.3a}$$

$$GCI[coarse\ grid] = GCI[\,fine\ grid\,] + F_s\,|\,\varepsilon\,| \tag{5.7.3b}$$

The last of these is easy to interpret; the error estimate for the coarse grid solution is just the error estimate for the fine grid solution, plus the difference between the solutions, which is ε. The GCI difference is then just F_s ε.

Applying this equation to the hypothetical cases in the second paragraph of Section 5.2, we see that a 4% difference from a grid refined by 50% using a 1st-order method gives a coarse-grid GCI = 36%, whereas a 6% difference from a doubled grid using a 2nd-order method gives a coarse-grid GCI = 24%. Note that the higher-order method only appears to be working against us here, because we are coarsening, rather than refining, the grid. In actuality, the ε for the higher order method will be smaller for the same grid refinement close to convergence.

GCI values for some common combinations of *r* and *p*, normalized to ε = 1%, are given in Table 5.7.1.

5.8 EXAMPLE GCI CALCULATION

A simple example of the calculation of a Grid Convergence Index follows. We choose the easily reproduced case of a steady-state Burgers equation

$$-V V_x + V_{xx} / \mathrm{Re} = 0, \quad V(0) = 1, \quad V(1) = 0 \tag{5.8.1}$$

for *Re* = 1000 solved with 2nd-order centered differences on a uniform grid, and evaluate the one-dimensional "shear" $f = dU/dx$ at $x = 1$. Using a fine grid calculation with 2000 cells, we obtain $f_1 = -529.41$. Then we coarsen the grid to 1600 cells ($r = 1.25$) and obtain $f_2 = -544.48$. The quantity typically reported from Eq. (5.4.3) would be $|\varepsilon| = 100\% \times (f_2 - f_1) / f_1 = 2.85\%$. The factor $(r^p - 1)$ is $(1.25^2 - 1) = 0.5625$. The magnitude of the Richardson Extrapolation error estimator for the fine grid solution from Eq.

(5.4.2) is $|E_1| = |\varepsilon| / (r^p - 1) = 2.85\% / 0.5625 = 5.07\%$. The fine grid value of the Grid Convergence Index from Eq. (5.6.1) with $F_s = 3$ is GCI [fine grid] = $3\ |\varepsilon|/(r^p - 1) = 3 \times 2.85\% / 0.5625 = 15.20\%$. Comparison with the exact solution $f_{exact} = -500.00$ indicates that the exact magnitude of the fine-grid error A_1 is $100\% \times |(f_2 - f_{exact}) / f_{exact}| = 100\% \times |-529.41 + 500.00| / 500.00 = 5.88\%$. As is typical, the Richardson Extrapolation error estimator E_1 is not conservative (5.07% < 5.88%), whereas the GCI is conservative and quite so with $F_s = 3$ (15.20% > 5.88%), in the spirit of a 2σ error band. With $F_s = 1.25$ (see the following Section 5.9), the GCI = 6.34% > 5.88%, still conservative but more palatable.

If the coarse grid solution (or the coarse grid solution to a "nearby problem") were to be used, the Richardson Extrapolation error estimator would be increased to $|E_1| + |\varepsilon| = 5.07\% + 2.85\% = 7.92\%$. The GCI with $F_s = 3$ would be increased by $3|\varepsilon|$ as in Eq. (5.7.3b) to GCI [coarse grid] = $15.20\% + 3 \times 2.85\% = 23.75\%$. The actual magnitude of the coarse grid error is $100\% \times |(f_2 - f_{exact}) / f_{exact}| = 100\% \times |-544.48 + 500.00| / 500.00 = 8.90\%$. Again, the Richardson Extrapolation error estimator E_1 is not conservative for the coarse grid (7.92% < 8.90%), whereas the GCI is conservative and quite so with $F_s = 3$ (23.75% > 8.90%). With $F_s = 1.25$, the GCI = 9.90% > 8.90%, still adequately conservative.

5.9 SHOULD THE COEFFICIENT BE "1" OR "3" OR "1.25"?

5.9.1 Determining the Factor of Safety

The functional form of the definition of the GCI as in Eqs. (5.6.1), (5.7.2), (5.7.3) is rational and objective, but the coefficient $F_s = 3$ is a judgment call. It could arguably be "1," or conceivably "1.5" or "2" or something else between 1 and 3.

F_s is essentially a "factor of safety," and the value $F_s = 3$ is possibly too conservative. As the quality and rigor of the grid convergence study increases, so does the conservatism of using the coefficient $F_s = 3$ in the definition of the GCI. However, consider the increased uncertainty associated with practical complications such as the following:

- difficulty of attaining the asymptotic range in 3-D,
- rapidly varying coefficients from turbulent eddy viscosities and/or
- strong grid stretching,
- nonlinear systems,
- non-uniform behavior of various error metrics,
- experimental determination of spatially varying p,
- non-monotonic convergence,
- vagaries associated with defining a roughly equivalent r when
    - when subgrids are not strictly geometrically similar,
    - when power-law grid stretching is used,
    - when regions are partitioned geometrically giving non-smooth spatially varying r, (as in Domain Decomposition or multi-block grid generation methods),
    - and the surprisingly common and strongly not recommended practice of using $r_x \neq r_y$.

Such complications, while not necessarily contradicting the ultimate applicability of Richardson Extrapolation (i.e., in the asymptotic range), do increase the uncertainty associated with the error estimate for many practical engineering calculations. Likewise, if the grid convergence exercise is only performed for a representative nearby problem, uncertainty is increased. (See Cosner, 1995 for experienced testimony

that "nearby" problems adequate to predict grid resolution requirements can be difficult to find.) Also, see discussion in Westerink and Roache (1995) and in Chapter 6 on the distinction between formal, actual asymptotic, and observed convergence rates. These considerations provide additional rationale for retaining "3" as the coefficient, in the sense of a "factor of safety," for reporting minimal two-grid convergence results.

As noted, using $F_s = 3$ makes grid doubling with 2nd-order methods the standard of comparison. This is not intended to make 2nd-order methods the goal, only the standard. (Like an IQ of 100, it is not meant to discourage genius.) It just means that for $p = 2$ and $r = 2$, we obtain GCI [fine grid] = ε. That is, it does not change what authors who use grid doubling with a 2nd-order method already have been reporting, namely ε.

Using the value $F_s = 1$ would make the GCI equal $|E_1|$, the error estimator obtained from Richardson Extrapolation. As noted earlier, since this is the best estimate we can make given only the information from calculations on two grids, we can only expect equal probability that the true answer is inside or outside of this band. Also, simple tests on the steady-state Burgers equation will quickly demonstrate that $F_s = 1$ is *not* usually conservative; see Table 5.9.1 below.

So the direct use of the Richardson Extrapolation Error Estimator E_1, i.e. GCI with $F_s = 1$, shows less than 50% probability of being conservative on this simple, model, 1-D problem. Is <50% probability acceptable for an error band? We think not. Also, use of $F_s = 1$ would make grid doubling with 1st-order methods into the standard of comparison, i.e. for $p = 1$ and $r = 2$, the GCI [fine grid] calculated using $F_s = 1$ would give GCI = ε. Clearly, we do not want 1st-order methods to be the standard of comparison!

A 50% "factor of safety" over the Richardson Error Estimator would be achieved with the value $F_s = 1.5$, or the naive value of $F_s = 2$ might prove to be a neat and reasonable compromise. But much experimentation would be required over an ensemble of problems to determine a near-optimum value and to establish the correspondence with statistical measures such as the 2σ band. Note that a true optimum would likely depend upon the family of numerical methods (e.g., medium-order FVM, high-order FDM or FEM, etc.) and upon the family of problems (e.g., turbulence, transonic, free surface, etc.).

All things considered, and after discussions solicited from many (on the order of 200) CFD practitioners, I had recommended in (Roache, 1994) the use of the value $F_s = 3$ in the definition of the GCI, even though I recognized and stated that it would be too conservative for high quality grid convergence studies. Of course, there is nothing to preclude an author from reporting *both* the GCI and the Richardson Error estimator E_1. As noted, the motivation for using $F_s > 1$ is that $F_s = 1$ corresponds to a 50% error band, which is not adequate. For many reasons (see discussion above, and in Chapter 8) this is not unduly conservative when only two grids are used in the study. However, it is became clear that $F_s = 3$ is overly conservative for carefully performed grid convergence studies using three or more grid solutions to experimentally determine (or Verify) the observed order of convergence p_{obs} (e.g., see the papers in Johnson and Hughes, 1995). For such high quality studies, a modest and more palatable value of $F_s = 1.25$ appears to be adequately conservative. However, for the more common two-grid study (often performed reluctantly, at the insistence of journal editors) we still recommend the value $F_s = 3$ for the sake of uniform reporting and adequate conservatism.

The 30 cases for Burgers equation in the above Table 5.9.1 were repeated for $F_s = 1.25$. These tests involved only 2 grids. There is no ambiguity about the order of convergence of the methods, with well-defined $p = 1$ or 2, but not all of the cases are clearly in the asymptotic regime, and some of the error estimates were fairly large, e.g. 20 % in Case 5. The question addressed was the following. Is the recommended $F_s = 1.25$ still overly conservative? The answer is, clearly not. Of the 30 cases, 18 were "Normal" (as defined in the Table legend) or more conservative, i.e. either the coarse or fine grid reliably conservative.

Case	p	Re	r	(mc, mf)	Behavior
1	2	10	2	20,40	Normal
2	2	10	1.1111	36,40	Normal
3	1	10	1.1111	36,40	Normal
4	2	50	1.5	100,150	Normal
5	2	1000	1.25	800,1000	Normal
6	1	1000	1.25	800,1000	Normal
7	2	5	2	500,1000	Normal
8	1	1000	1.1111	900,1000	Normal
9	2	30	3	100,300	Normal
10	2	50	2	50,100	Normal
11	2	50	1.1111	90,100	Normal
12	2	50	1.0101	99,100	Normal
13	1	50	2	50,100	Normal
14	1	50	1.1111	90,100	Normal
15	1	50	1.0101	99,100	Normal
16	1	0.1	2	20,40	all are conservative
17	1	0.5	2	20,40	coarse grid E_1 is also conservative
18	1	0.5	2	500,1000	Normal
19	2	–1	2	20,40	Normal
20	2	–10	2	20,40	Normal
21	1	–10	2	20,40	all are conservative
22	2	–100	1.1111	90,100	Normal
23	1	–100	1.1111	90,100	all are conservative
24	2	1000	2	1000,2000	Normal
25	1	1000	2	1000,2000	Normal
26	2	10	4	25,100	Normal
27	2	10	3.0303	33,100	Normal
28	2	10	2	50,100	Normal
29	2	10	1.1111	90,100	Normal
30	2	10	1.0101	99,100	Normal

TABLE 5.9.1. STEADY BURGERS EQUATION SOLUTIONS DEMONSTRATING THAT F_s = 1 IS NOT USUALLY CONSERVATIVE.

The Richardson Error Estimator (E_1) and the Grid Convergence Index (GCI) are applied to the Steady-State Burgers Equation $-V V_x + V_{xx}/Re = 0$. p = order of method. $Re > 0$ indicates boundary conditions of $V(0) = 1$, $V(1) = 0$, suggestive of stagnation flow. $Re < 0$ indicates boundary conditions of $V(0) = 0$, $V(1) = 1$. r is the grid refinement ratio = mf / mc, where mf = number of fine grid cells, mc = number of coarse grid cells. Under the heading "Behavior," the entry "Normal" indicates the following pattern: compared to the exact solution, (a) the E_1 is not conservative, i.e. $|E_1| >$ |true error|, for both the coarse and fine grid estimators; and (b) the GCI is conservative, i.e. GCI < |true error| for both the coarse and fine grids.

This is, of course, a small sample, but it does not indicate anything like the desired 2σ band (~1 in 20) which would only allow for 1 or 2 non-conservative cases out of the 30. The low order methods faired worse, as expected. Considering only the 25 stagnation-like flows ($Re > 0$), 12 of 15 cases (more like a 1σ band) were conservative for $p = 2$ (the 2nd-order method), whereas only 3 of 10 cases were conservative for $p = 1$. Furthermore, these 3 were the very low *Re* Cases (16,17,18).

All 30 cases were conservative for $F_s = 1.85$, and 26 of 30 were conservative for $F_s = 1.35$. The implication is that the 2σ band for this small sample would be achieved for roughly $F_s = 1.5$. However, I again emphasize that this is a toy problem: one-dimensional, steady state, unambiguous theoretical order of convergence, thorough iteration convergence, simplest possible boundary conditions, etc. It would seem that $F_s = 1.5$ is the minimum one could use reliably for 2 grid solutions. If one performs thorough grid convergence tests for a suite of representative problems of interest and establishes a convincing correlation, one could possibly convince the audience that $F_s = 1.5$ is akin to a 2σ band. But in the absence of such a thorough and specific study, I still recommend $F_s = 3$ for 2-grid studies, and reserve $F_s = 1.25$ only for carefully performed grid convergence studies using three or more grid solutions to experimentally determine the *observed* order of convergence *p*. (See further discussion in Chapter 8.)

5.9.2 Δ Summary Recommendations for the Factor of Safety

The summary recommendations on the Factor of Safety for the GCI are thus as follows.

(a) Use $F_s = 1.25$ for convergence studies with a **minimum** of three grids to experimentally confirm that the observed order of convergence p_{obs} for the actual problem is reasonable, and
(b) use $F_s = 3$ for two-grid convergence studies (since a p_{obs} cannot be calculated and therefore there is no way to demonstrate that the grids are in or at least near the asymptotic regime).

$F_s = 1.25$ should not be used if the results from the minimum three grids produce a suspicious observed p_{obs}. (See Section 5.10.6.) It is imprudent to use p_{obs} > theoretical *p* in the GCI formula (unless there is some rare good reason to expect superconvergence). It is also recommended to *look* at the grid convergence results; no matter what the calculated numbers are for p_{obs}, if the results look suspicious, then at least use $F_s = 3$, and possibly continue the grid convergence study. An example of suspicious results would be (from coarse to fine grid solutions) $f = 0.12, 0.14, 0.15, 0.13$, which might suggest that the coarsest two grids are outside the asymptotic range. (E.g. see Eça et al, 2009, Figure 2a.) Also, there is a real possibility that unlucky sampling of oscillatory convergence has given a false indication of p_{obs} close to theoretical. Unless there is some other indication that the convergence is monotone (e.g., more complete results for a nearby problem) then more than three grids may be appropriate, depending on the consequences. (It is easier to justify a single grid triplet calculation for a scientific paper than for a nuclear reactor safety analysis.)

If agreement of p_{obs} with expected *p* is good, e.g. $p_{obs} = 1.97$ or 2.05 for an expected 2nd-order method, one can proceed with some confidence, especially if this level of agreement applies to point values and more than one grid triplet has been used to calculate p_{obs}. If p_{obs} is very bad, e.g. $p_{obs} = 0.6$ for one grid triplet and $p_{obs} = 3.7$ for another, clearly you have a problem, and more grid studies are required to get into the asymptotic regime. As always, the gray areas are difficult to characterize with any generality. Is $p_{obs} = 1.7$ acceptable? Perhaps, and more likely if it holds (roughly) for more than one grid triplet and for point values as well as solution functionals. But the decision will depend on the consequences. The general recommendation is to look at all the results you have, weigh the consequences of your decisions, and consider doing more grid studies, applying upward pressure to management if necessary. Note that the farther away you are from apparent asymptotic convergence, the more likely it is that convergence is actually oscillatory and you might be getting misleading samples in the grids used.

5.9.3 § Error Estimate vs Uncertainty Estimate or Error Bar

There exists considerable confusion over the distinction between an error estimate, e.g. that supplied by Richardson Extrapolation, and an uncertainty estimate, e.g. that supplied by the GCI. The source of the confusion is the time-honored term "error bar" which is not, as one might reasonably think at first glance, an error estimate at all, but rather an uncertainty estimate or confidence interval, i.e. a range of expected possible errors (to some "probability" or "confidence[34] level" or "coverage"). The following Table 5.9.3.1 (adapted from Roache, 2003a) lists the distinctions.

An Error Bar (or Error Band) ...	**An Error Estimator...**
...is a $U_{95\%}$.	...is a (signed) $U_{50\%}$.
...uses $\|E_1\| > 0$	...uses signed $E_1 > 0$ or < 0
...is not an ordered approximation but an empirical correlation based on computational experiments.	...is an ordered approximation, based only on asymptotic theory, not based on computational experiments.
...may be accurate (statistically) even outside the asymptotic range.	...depends for accuracy upon the grid sequence being in the asymptotic range.
...could be determined from data for the problem ensemble without an error estimator.	...is not determined from data for the problem ensemble.
...is what is needed for Calculation Verification prior to Validation.	...is what is needed for an RE corrected solution.

Table 5.9.3.1 Summary of Distinctions between Error Bars and Error Estimators

Also contributing to the confusion is the phrase "conservative error estimate" which also is better considered in terms of uncertainty rather than error. People argue that, since the RE result $E1$ is an error estimate, then $F_s \times E1$ is just a more conservative error estimate, hence it should be called an "error estimate." This may seem plausible at first, but a little thought will lead one to recognize that there is no

[34] "Confidence" is another loaded technical term. As a statistical term, it is dependent on assumptions and techniques of statistical analysis, rather than just a straightforward "coverage" or counting of cases for which U_{num} was conservative or not, compared to the actual error. A claim of "confidence interval" would be based on a conceptual model in which the examined data set of an individual study is taken to represent the entire population, which is inaccessible. If this data set is small, "small sample correction" techniques (like Students *t*-test) are applicable. The simpler "coverage" approach just claims (say) "89.2% coverage" for an individual study (perhaps small) with the contextual idea that the results of these small studies will eventually be aggregated (probably informally). If small sample corrections for "confidence interval" are made to each study, later aggregation is confused (because the small sample corrections are nonlinear and non-distributive). See also Section 5.14.

limit to conservatism, so F_s could be any number > 1. Then, there can also be such a thing as a non-conservative error estimate, so F_s could be $0 < F_s < 1$. This leads to the logical conclusion that any number with the same sign as $E1$ is an error estimate, either conservative or non-conservative. But half the real number line hardly deserves the dignity of the designation "error estimate."

Oberkampf and Trucano (2007, 2008) make the following distinctions about uncertainty in measured quantities:

(a) an interval uncertainty, i.e. there exists[35] a single true value, and that true value is believed to lie in the stated interval, but no other information is available concerning the true value;

(b) an imprecise probability distribution, i.e. the true quantity is a random variable [rather than a single value as in (a)] characterized by a known family of probability distributions whose parameters are only stated as intervals;

(c) a precise probability distribution, i.e. the true quantity is a random variable characterized by a known family of probability distributions whose parameters are accurately known.

Although presented for experimental measurements, the distinctions are relevant to numerical uncertainty. Clearly, numerical uncertainty in general, and the GCI in particular, is in category (a), an interval uncertainty. But while recognition of these distinctions is worthwhile, we also note that there is no distinction made[36] in the combinations of different types of experimental uncertainties so the ultimate combination of numerical and experimental uncertainties will not be affected; see Chapter 11.

5.9.4 § Mixed Order Methods

The term "mixed order methods' could indicate different orders in different coordinate directions (e.g. first order in t, second order in x and y) as above. However, it also could indicate the common situation in which some terms (e.g. advection) are discretized to first order and others (e.g. diffusion) are second order. Usually, as above, the mixed order situations are just treated as a first-order method and the error contribution from that lowest order term is estimated. That is, using the single-term lowest order expansion as in Eq. (5.3.1) repeated here,

$$f = f_{exact} + g_1 h + g_2 h^2 + g_3 h^3 + \cdots \tag{5.9.4.1}$$

we use two grid solutions for f_1 and f_2 to solve for the g_1 term (and f_{exact}) and relegates all the other terms to $O(h^2)$. This simple approach wastes some information, and can be improved upon as in Roy (2003). Instead of limiting the analysis to terms of $O(h)$ and evaluating g_1 from two grid solutions, Roy includes another term and evaluates both g_1 and g_2 from three grid solutions (and higher). Besides providing for more accurate error estimates (for fairly well-behaved problems[37]) the analysis illuminates common difficulties in grid convergence behavior. Depending on sign changes in g_1 and g_2, non-monotonic convergence can be produced by mixed order methods (Roy, 2003) without the complications of shocks, switches in turbulence models, other nonlinearities, etc.

---

[35] Paraphrased slightly for emphasis.

[36] See ASME PTC 19.1 (ASME, 1986, 2006), NIST TN 1297 (Taylor and Kuyatt, 1994), ISO (1995), V&V20 (2009), Coleman and Steele (1995, 2009).

[37] The ITTC Manual (ITTC, 2002), page 5, noted that these multi-term power expansions are more restricted in use; two term expansions require five computational solutions, all of which must be within or close to the asymptotic regime. In general, an n-term expansion would require $2n+1$ solutions.

5.10. ADDITIONAL FEATURES OF GRID CONVERGENCE STUDIES FOR VERIFICATION OF CODES AND CALCULATIONS

The following sub-sections are applicable to the use of the GCI for Quantification of Uncertainty of actual calculations, and were originally presented in that context in Roache (1994). However, the discussion is equally applicable to *any* systematic grid convergence study, whether or not the GCI is used (and, if GCI is used, to any value for F_s), and equally to Verification of Codes as well as Verification of Calculations.

5.10.1 Non-Integer Grid Refinement

Although it is generally assumed that grid doubling is preferable, it is argued here that, especially for the computer limitations frequently encountered in practice for multidimensional problems, it may be better to use a smaller change in grid resolution, perhaps as low as 10%. Consider a base grid, and refine or coarsen. If engineering intuition or studies on related (nearby) problems have led to a good (economical, yet adequately accurate) grid selection for the base grid, then we are likely in the asymptotic range, but perhaps just barely. If we can afford to double the grid, we will certainly get more accurate answers, but the cost can be large. With an optimal numerical solution method, e.g. a good multigrid method, in which the computing cost is merely proportional to the number of unknowns, doubling a grid in three space dimensions and time will increase the cost over the base grid calculation by a factor of 16; if sub-optimal methods are used (which are much more common), the penalty is worse. If we coarsen the grid instead, the economics work for us, i.e. the coarse grid solution is only 1/16 as expensive as the base grid. However, the coarse grid solution may be out of the asymptotic range. This situation is especially evident in turbulent boundary layer calculations, wherein we need sublayer resolution ($y^+ < 1$) for the first grid point off the wall. (For example, see Shirazi and Truman, 1989 or Wilcox, 1993.) This applies only when the turbulence equations are integrated to the wall; different requirements apply if wall functions are used. (For example, see Celik and Zhang, 1993,1995 or Wilcox, 1993.)

Since the theory of generalized Richardson Extrapolation is valid for non-integer r, it is easier to use a small value (unless the analyst is confident that the coarse grid with $r = 2$ will still be in the asymptotic range). However, there are practical limits to small r. For example, increasing the number of grid points by 1 in a base grid calculation of a 100×100 grid gives $r = 1.01$, and the theory is still valid. (Indeed, it would still be better than no grid convergence study at all.) But the results will now be obscured by other error sources, e.g. the "noise" of incomplete iteration convergence and machine round-off error, especially for accurate solutions. That is, as we reduce the change in the discretization error by using $r \rightarrow 1$, the leading discretization error term may be swamped by noise. As an intuitive engineering guess, a minimum 10% change ($r = 1.1$) is recommended. Of course, provided that the coarse grid is within the asymptotic range, it is intuitively obvious that the error estimates are more reliable for larger r, *for grid refinement*.

It is perhaps less obvious that the opposite is true for grid coarsening, i.e. when we keep the answers of the finer grid. In that case, there is *more* additional information, and therefore sharper error estimates, available for $r \sim 1$ than for $r >> 1$ (limited only by noise pollution from round-off and incomplete iteration errors). For example, consider a fine grid using 100 nodes. A coarse grid calculation using 90 nodes ($r = 1.1111...$) contains more information (and is more expensive) than a coarse grid calculation using 50 nodes ($r = 2$). (This argument fails for a "coarse grid" = 100 node, in which case there is no new information.)

A reviewer of Roache (1994) expressed skepticism that one really could learn anything about grid convergence by changing the resolution by only 10%. A set of easily reproduced calculations was performed on the steady-state Burgers equation $-V V_x + V_{xx} / Re = 0$, with $V(0) = 1$ and $V(1) = 0$ (suggestive of stagnation flow), $Re = 10$, and $p = 2$, with a fine grid of 100 interior nodes and coarse grids

from 25 to 99 nodes. The results are shown in Table 5.10.1.1. As seen, the error estimator for the one-dimensional "shear" $f = dV / dx$ at $x = 1$ obtained using the coarse grid of 90 nodes ($r = 1.1111$, or a factor of 0.9 coarsening) is 3.2 times *more* accurate than the error estimator obtained using the coarse grid of 50 nodes ($r = 2$, or a factor of 0.5 coarsening). There are practical limits on $r \rightarrow 1$ from all sources of noise, notably incomplete iteration convergence and switching in RANS turbulence models. Thus the *ASME Journal of Fluids Engineering* policy statement (Celik et al, 2008) recommends the GCI with a minimum $r = 1.3$, but my experience has shown this is unnecessarily conservative for many (non-RANS) problems. Part of the difficulty experienced by RANS modelers may be the common use (default in some commercial CFD codes) of an inadequate criterion for iteration convergence, requiring only 10^{-3} reduction in initial residuals. (See Section 5.10.10.1.)

5.10.2 Δ Independent Coordinate Refinement and Mixed Order Methods

The simplest and most reliable way to apply the Grid Convergence Index is to use a single parameter r to refine/coarsen the grid in all coordinates, space and time. However, there are sometimes good reasons for not doing this. In calculations of boundary layer flows (whether using boundary layer equations, full Navier-Stokes equations, or something intermediate) it is often the case that grid convergence is easy to establish in the longitudinal direction (being essentially dictated by the free-stream flow, which is not sensitive to Reynolds number) but is more problematical in the transverse direction, being sensitive to *Re*. Also, computer resource restrictions may prohibit grid doubling in each direction simultaneously, but allow it one coordinate at a time; e.g., see the convincing studies in Zha and Knight (1996). Especially, in time-dependent problems it is much easier to develop a code that is solution-adaptive in the time-step than in the spatial grid, so that time discretization errors might be independently controlled (e.g., see Roache, 1991, 1992a, 1993) and the systematic grid convergence test would be restricted to the spatial grid.

In such cases, the dominant directional component of GCI can be obtained orthogonally, by independent coordinate refinement.[38] For example, in a 2-D time-dependent boundary layer calculation, if essential grid independence has been achieved in t and x, one may make an error estimate in y only using the 1-D GCI, and add (heuristically) the contributions from t and x.

$$GCI \cong GCI_t + GCI_x + GCI_y \quad for \quad GCI_t + GCI_x << GCI_y \tag{5.10.2.1}$$

It is important to note that the procedure must be performed globally, i.e. with complete global solutions obtained for each refinement in independent coordinate directions; attempts to apply the extrapolation procedure by lines do not produce ordered or usable error estimates.

Consistent Richardson Extrapolation error estimators cannot be obtained from just two calculations (a coarse and a fine grid calculation) when different r are used in different coordinate directions, because there is no basis for separating out the directional contributions. In this case, a conservative GCI could be used, based on the smallest directional r. For example, in a two-dimensional steady flow calculated in a fine grid of 100 × 100 cells and a coarse grid of 50 × 75 cells, unless other theoretical considerations apply, we would have to conservatively attribute the change in solution to the more modest grid refinement, and use $r = 4/3$ to calculate the fine-grid GCI from Eq. (5.6.1). Alternately and preferably, three grid solutions can be

[38] V&V1 incorrectly stated that the technique was more general. Thanks to Dr. F. Blottner for pointing out the error (Blottner and Lopez, 1998).

used in a 2-D problem, as shown by Roy (2004); see below. However, such directional splitting with three grids does *not* work for extracting observed *p*; see Section 5.10.6.1.

A similar situation occurs with mixed-order methods, e.g. the not uncommon situation of a method with 1st-order time accuracy and 2nd-order space accuracy, or first-order in the dominant flow direction (say x) and second-order in y. A conservative approach would be to use $p = 1$ in Eq. (5.6.1), but a better estimate (and more optimistic GCI) would be obtained using separate grid convergence studies in space and time, using $p = 1$ for the time contribution and $p = 2$ for the space contribution from Eq. (5.6.1), and simply adding the results as in Eq. (5.10.2.1) provided that $GCI_t << GCI_{x,y}$. Alternately, one may use different r in different coordinates to compensate for different p. For time-dependent problems, Richards (1997) verified the theoretical order of convergence of an $O(\Delta t, \Delta x^2)$ method by a grid refinement study in which Δx was reduced by a factor of 2 and Δt by a factor of 4, giving a theoretical ratio of the errors on successive space-time grids of 4; his representative experimental values (in his Table I, i) were 3.76, 3.95, 3.97. Similarly convincing verification was obtained for his extension of the "Completed Richardson Extrapolation" (Roache and Knupp, 1993) to space and time. For example, the completed Richardson extrapolation applied to the $O(\Delta t, \Delta x^2)$ method should be $O(\Delta t^2, \Delta x^4)$. Reducing the Δx successively by factors of 3, and Δt by factors of 9, gives a theoretical error reduction by a factor of 81 for each successive grid refinement. Richards' representative experimental values (in his Table I, iv, a) were 78.8 and 80.1, a remarkable Verification.

Kamm et al (2003) have shown how to extract observed convergence orders in space and time, including space-time cross-derivatives. It is surprisingly difficult, even with the assumption of same p in all spatial directions. They write the discretization error equation as

$$f_k^{\ell} = f_{exact} + g_x(\Delta x_k)^p + g_t(\Delta t_{\ell})^q + g_{xt}(\Delta x_k)^r(\Delta t_{\ell})^s + H.O.T. \qquad (5.10.2.1)$$

where subscript k refers to the spatial grid index and superscript ℓ refers to the time discretization index. Note the generality of the four convergence orders and three error coefficients. In their problem in Code Verification for which f_{exact} is known, these seven unknowns require seven equations generated by seven different space-time grid solutions; the coupled equations are solved by Newton's iteration. Roy (2004) stated that "their approach can be easily extended for error estimation by simply adding another mesh level and solving for f_{exact}" but it is doubtful that a method requiring eight discretized solutions, even when limited to the assumption of equal order of convergence in all space coordinates, will see much use beyond Code Verification, even if it is not corrupted by noise in RANS or other difficult calculations.

The following approach due to Roy (2004) is presented for spatial differencing only and is based on known (theoretical) orders of convergence, but does allow for different convergence orders and refinement factors in different space coordinates in the absence of spatial cross derivative terms. The expansion is

$$f_k = f_{exact} + g_x(\Delta x_k)^p + g_y(\Delta y_k)^q + H.O.T. \qquad (5.10.2.2)$$

where subscript k refers to the spatial grid index. The base grid $k = a$ produces the solution f_a. Consider first refinement [or coarsening] in only the x coordinate by r_x, giving the grid $k = b$ and producing the solution f_b. Recall that the coefficients g_x and g_y depend on the continuum solution, so the last term of the equation is constant for both grids. Unlike the situation with $p = q$, we cannot solve for f_{exact} from the two equations but we can solve for an aggregated term f_x defined as

$$f_x = f_{exact} + g_y(\Delta y_k)^q \qquad (5.10.2.3)$$

The two solutions on grids $k = a$ and $k = b$ give

$$f_a = f_x + g_x(\Delta x_k)^p \tag{5.10.2.4}$$

$$f_b = f_x + g_y(r_x \Delta x_k)^p \tag{5.10.2.5}$$

These two can be solved for f_x as

$$f_x = f_a + \frac{f_a - f_b}{r_x^p - 1} \tag{5.10.2.6}$$

and the leading x-coordinate error term,

$$g_x(\Delta x)^p = \frac{f_a - f_b}{r_x^p - 1} \tag{5.10.2.7}$$

Similarly for refinement (or coarsening) in only the y coordinate by r_y, giving the grid $k = c$, producing the solution f_c and giving

$$g_y(\Delta y)^q = \frac{f_a - f_c}{r_y^q - 1} \tag{5.10.2.8}$$

A powerful aspect of this directional refinement process using three grids [even if $p = q$] is that comparison of Eqs. (5.10.2.7) and (-8) can guide the analyst in making further grid refinements; e.g. if $g_x(\Delta x)^p << g_y(\Delta y)^q$ then the total error will be economically reduced by further refinement in y only. This information is not obtainable from a uniform refinement in both directions and two grids.

For hybrid methods that shift locally to two-point upstream differencing for large cell *Re*, the conservative $p = 1$ should be used. However, hybrid methods are often inaccurate, and often produce non-monotonic convergence even for gross properties like reattachment length (Barton, 1995) making it difficult to have confidence in the extrapolation process. Hybrid methods are often not recommended (Barton, 1995; Leonard and Drummond, 1995) but the GCI has produced good uncertainty estimates even with these (see Section 6.23.2).

For methods that use higher order stencils for advection than for diffusion, the error will be dominated asymptotically by the lower order term. For example, for solutions calculated by Leonard's ULTIMATE method (Leonard, 1991) which is 3rd-order for advection, the GCI should be reported conservatively using $p = 2$ in Eq. (5.6.1) or (5.7.2). Note that the accuracy advantage of mixed order methods is not lost in this process. The 3rd-order accuracy for the (dominant) advection term will produce smaller errors than a 2nd-order method, so the *level* or *size* of error will be lower. However, the *slope* of the convergence asymptotically will be 2nd-order, so the extrapolation and error banding should be based upon $p = 2$. (See also Section 5.9.4.) All this applies to calculating a GCI when p's are known or assumed. When the more scrupulous approach of calculating observed p's is used, the techniques are more difficult. See Sections 5.10.6 and 5.11.

5.10.3 Δ Non-Cartesian Grids, Boundary Fitted Grids, Adaptive Grids, Unstructured Grids

5.10.3.1 Non-Cartesian Grids and Boundary Fitted Grids

The procedures for calculating GCI definitely apply to non-Cartesian grids, including non-orthogonal boundary fitted grids, with some special considerations and caveats.

The Taylor series basis of Richardson Extrapolation applies to stretched orthogonal and non-orthogonal grids as long as the stretching is analytical. It is cleanest to apply in the transformed plane (ξ, η, ζ) where r is defined as above. The order of the extrapolation accuracy will now be affected by the order and iteration convergence of the grid generation equations. Shirazi and Truman (1989) found a surprising

sensitivity of the error estimates to discretization of metrics and Jacobians, and to incomplete iteration convergence. For another example, if strong exponential source terms are used (e.g. near trailing edges of airfoils) which depend strongly on *h* (e.g., see Thompson et al, 1985) then a refined-grid generation will pollute the Richardson Extrapolation.

However, even if the grid generation equations are not converged, making the actual Richardson Extrapolation less dependable, it is still recommended that the uniform GCI be reported rather than the simple raw data of ε.

For the simple power-law stretching in one coordinate, commonly used in boundary layer codes, the ratio of successive grid increments *h* is constant, i.e.

$$h^{i+1} = sh^i \tag{5.10.3.1}$$

where *s* is the constant stretching factor. Nodal comparisons can be made during grid convergence without interpolation if the coarse and fine grids have nodes in common. This is accomplished during a grid doubling/halving by requiring (Ferziger and Peric, 1996) that the fine-grid stretching factor s_f be related to the coarse grid s_c by

$$s_f = \sqrt{s_c} \tag{5.10.3.2}$$

and by requiring the initial fine-grid spacing h_{f1} be related to the coarse grid h_{c1} by

$$h_{f1} = \frac{h_{c1}}{1+s_f} \tag{5.10.3.3}$$

The simpler strategy of inserting fine-grid points half way between the coarse-grid points introduces an additional truncation error term to differences centered (in logical space), but the additional error term is itself ordered. For a relatively coarse grid, the error is larger, but asymptotically the convergence rate is still 2nd-order (Ferziger and Peric, 1996).

5.10.3.2 Δ Adaptive Grids

Solution-adaptive grid generation codes may have their own internal *local* error estimators. Most often, solution adaptive grid generation of either the redistribution type (Thompson, et al, 1985; Knupp and Steinberg, 1993) or enrichment type is not based on any true error estimator but on solution behavior (gradient, curvature, or simply resolution requirement) which is only loosely related to *local* error (which in turn is loosely related to the *global* error of interest herein). In such a case, the GCI reporting procedure recommended herein can be applicable if the solution-adaptive procedure is used only to obtain the base grid solution. This grid can then be changed non-adaptively, perhaps refined by a higher order interpolation or coarsened by simply removing every other point (as in Zingg, 1991,1992). Then the GCI of Eqs. (5.6.1) or (5.7.2) can be applied to this new grid. However, for *time-dependent* solution-adaptive grids, practical coding difficulties exist and it is not clear how to perform meaningful *global* error estimation nor uniform grid convergence reporting in this important situation.

5.10.3.3 Δ Unstructured Grids

Another difficulty occurs for unstructured grids. If the base grid is unstructured, the GCI procedure would still apply if one used a systematic method of grid refinement, that is, structured global (vs. local)

refinement of an unstructured grid.[39] For example, refining each base grid triangle into four new triangles (by connecting the mid-points of the sides) gives $r = 2$ for use in Eq. (5.6.1). However, if the refinement is also unstructured, as occurs in some algorithms and in user-interactive grid generation codes, there is no systematic and quantifiable grid refinement index like r to use in Eq. (5.6.1). Such grid refinement FEM studies are often reported simply in terms of the total number of elements used in the coarse (N_2) and fine (N_1) grids[40]. Use in Eq. (5.6.1) of an

$$effective\ \ r = \left(\frac{N_1}{N_2} \right)^{1/D} \tag{5.10.3.3.1}$$

where D is the dimensionality of the problem, and reporting this GCI is clearly preferable to simply reporting ε. But it does not have the firm basis of a structured grid refinement, and may significantly underestimate or overestimate the accuracy, depending on whether the grid refinement algorithm (or the intuition of the interactive user) refined the grid in the critical areas or not. For example, well-placed refinement in a relatively small critical area combined with coarsening in a larger but unimportant area could lead to improvement in global accuracy with a *decrease* in the total number of cells. (For example, see Pelletier and Ignat, 1995.) This would render the effective grid refinement ratio r, and the Richardson Extrapolation procedure, meaningless.

For unstructured grid refinement and structured or unstructured grid adaptation, I suppose it should be the burden of the algorithm developer to convince the reader/user that the local grid adaptivity process (sometimes based on a *local* error estimator or error indicator) can be usefully correlated with a meaningful engineering *global* error estimate, which is the real interest.[41]

For an early application of the GCI to unstructured grids with Finite Element Methods, using unstructured refinement, see Pelletier and Ignat (1995). They also show the correlation between the GCI results based on generalized Richardson Extrapolation with the theoretically distinct single-grid error estimators of the Zhu-Zienkiewicz (1990) class. (See also Chapter 6, Sections 6.7 and 6.8.) Pelletier and Ignat convincingly demonstrated the efficacy of their solution adaptive mesh generation algorithm. With a target increase of a factor of 3 in accuracy with the new mesh generation, they confirmed values *a posteriori* greater than 2. Hagen (1997) used the GCI theory with structured refinement of unstructured triangular grids in ocean calculations.

The systematic refinement of a non-structured triangular mesh described above, producing a well defined $r = 2$, does not work in unstructured 3-D meshes (e.g. see Hay and Pelletier, 2008). Local *h*-refinement of FEM may improve accuracy but produces non-smooth mesh variation. An alternative approach is to use unstructured grid generation algorithms (like the advancing front technique) that distribute mesh elements according to a specified mesh density function, which can be taken as approximately uniform (restricted by boundary geometry), or arbitrarily user-specified, or, in its most

[39] As Roy (2004) observed, grid refinement is usually easier than grid coarsening for unstructured grids, and the opposite for structured grids.

[40] Another common method of reporting unstructured grid refinement is to pick some representative element size, e.g. average √(element area) in 2-D, to calculate effective r. This is recommended in the *ASME JFE* policy statement (Celik et al, 2008) and used in V&V20. It has some appeal of specificity, but of course the dimensionalization does not affect the dimensionless r.

[41] In fluid dynamics, this is very difficult to accomplish convincingly for any problem which a fluid dynamicist would consider non-trivial (mainly because local truncation errors are advected downstream) but it can be done; see Schonauer et al (1981).

powerful application, within a solution adaptive remeshing procedure. If the mesh density distribution function of one mesh is uniformly reduced everywhere by the same factor, that factor will be an approximate or effective *r*. The new mesh is not geometrically similar to the original (and in fact the mesh topology will often change) yet in a rough sense the refinement or coarsening is uniform. Observed *p* have been calculated with good results using effective *r* in unstructured grid refinement in two 2-D studies: heat conduction (see Section 6.4.2 for details) and turbulent separated flow RANS calculations using solution adaptive remeshing (Hay and Pelletier, 2008).

In spite of these difficulties, unstructured grids will continue to be more dominant in computational PDEs. As noted in the review by Carey (2006), structured grids are "geometry challenged" [especially in 3-D]. Ultimately, practice may go to Meshless Methods, as predicted in the review by Pepper (2006), but these are in a relatively early stage of development and no systematic and reliable methods for Calculation Verification are available.

5.10.4 Δ Shocks, Discontinuities, Singularities

In both his 1910 and 1927 papers, Richardson already considered the effect of singularities on the extrapolation procedure. These cases must be considered individually. If the form of the singularity is known *a priori*, it may be removed analytically. If unknown, its presence may be detected by checking to see if the asymptotic range has been reached (see below). Shocks and other discontinuities (e.g., contact surfaces) invalidate the Taylor series basis of Richardson Extrapolation, but unless the flow contains large numbers of complex shock patterns, the GCI procedure herein would still seem to have validity and be recommended, certainly an improvement over simply reporting the raw data for ε. As pointed out by Ferziger (1993), a more appropriate error measure here might be the shock position. Further experience with complex shocked flows is needed.

Blottner (1990) has shown how the concept of Richardson Extrapolation can be applied to systematically estimating the error due to artificial dissipation terms used in hypersonic shock calculations. The contribution of these terms to the GCI must also be calculated orthogonally to the other terms. If these terms are not estimated separately, the grid convergence tests will be polluted, since the (nonlinear) shock dissipation terms depend on *h*, and therefore the continuum problem being approximated changes from grid to grid. (This is the same difficulty that can appear with grid generation equations, noted above.). Blottner's results will be presented in Chapter 6, Section 6.11. (See also Kuruvila and Anderson, 1985.)

Also, the theory of Richardson Extrapolation is not applicable to nonlinear flux limiters, but again we expect these to be local applications, and still recommend the reporting of GCI over simply reporting the raw data for ε. But we would also recommend more detailed investigation, e.g. perhaps 3 grids (see below). The point is that the presence of shocks, other discontinuities or singularities can complicate grid convergence studies whether or not the GCI is used for reporting the results, so these complications do not constitute a criticism of the GCI.

5.10.4.1 § Detection and Treatment of Singularities

In a paper on Finite Element Analysis (FEA) of Computational Solid Mechanics (CSM) stress problems, Sinclair et al. (2006) presented a thorough work on the detection and treatment of singularities during grid convergence studies. The work should be equally applicable to other areas of computational PDEs although it remains to be tested. The methods can automatically detect and distinguish between cases of power singularities, logarithmic singularities, or simply grids not yet in the asymptotic range. Of course, there is a gray area and account is taken of ambiguous results; the most difficult cases to distinguish are weak singularities (e.g. a terribly difficult behavior of Hertzian contact stress converging at $p \sim 0.1$.)

The authors used the approach of evaluating the performance of the singularity treatments by using realistic exact solutions with prescribed singular behavior produced by the Method of Manufactured Solutions. Their development of MMS was independent, and they referred to it as Tuned Test Problems (TTP) in a paper with both CSM and CFD examples (Sinclair et al, 1997). The evaluation exercises were thorough, with 21 numerical experiments on 14 trial problems with power singularities, 21 experiments on 5 problems with log singularities, and 103 experiments on 18 problems with nonsingular stresses. The authors examined convergence for a stress of interest, denoted here[42] by *f*. All the tests used a grid refinement factor r ≈ 2, and the empirical terms given below will depend on this. They considered several levels of convergence checks for non-singular problems that are also worthwhile. Alternative methods of convergence checking from the previous literature were found to be lacking: the two-mesh check (i.e. no evaluation of observed *p*) and the seriously misleading "linearly-increasing mesh sequence" (see discussion in Section 5.10.6.3). Rather than targeting 95% certainty as in the GCI approach, Sinclair et al. adopted a practical alternative approach to calculation uncertainty, to be described in Section 5.12.

The singularity detection methods of Sinclair et al (2006) are as follows. We relate here only their initial provisional convergence which is required in the singularity detections. With f_1 = solution on the finest grid of a grid triplet, f_2 = on the medium grid, and f_3 = on the coarse grid, the computational solutions are judged to be converging if

$$| f_2 - f_3 | > | f_1 - f_2 | \qquad (5.10.4.1.1)$$

Of course, if all three values are so close that these differences are in the noise level, one would simply say that the solution *has already* converged.

Power singularities

With *R* being the dimensionless radial distance from the singular point, the local stress *f* for power singularities by definition behaves as

$$f = O(f_0 R^{-\gamma}) \quad as \quad R \to 0 \qquad (5.10.4.1.2)$$

where f_0 is an applied stress and γ is the singularity exponent. Local computed values of stress are typically extrapolated from nearby points in the elements adjoining the singular point. The signature behavior of the power singularity during mesh refinement ($\Delta \to 0$) is then as follows.

$$f_2 / f_3 \sim f_1 / f_2 \sim r^{\gamma} \quad as \quad \Delta \to 0 \qquad (5.10.4.1.3)$$

Appropriate limitations on the convergence behavior (i.e. on f_1, f_2, f_3) are required to avoid indeterminacies or worse, e.g. divide by zeros. To implement this asymptotic result, one obtains successive estimates of the singularity exponent γ as follows.

$$\gamma_{23} = [\ln(f_2 / f_3)] / \ln r \qquad (5.10.4.1.4a)$$

$$\gamma_{12} = [\ln(f_1 / f_2)] / \ln r \qquad (5.10.4.1.4b)$$

[42] The notations are changed from that of Sinclair et al (2006) to be consistent with the rest of this Chapter. The usual notation for stress is σ.

again with appropriate limitations on f_1, f_2, f_3. Then, one judges a power singularity to be present if these two successive estimates are approximately constant. The specific criterion tested by Sinclair et al (2006) is that the change in the successive values is less than 10% of the average value.

$$\frac{2\,|\,\gamma_{23}-\gamma_{12}\,|}{\gamma_{23}+\gamma_{12}}<0.1 \tag{5.10.4.1.5}$$

If this relation does not hold, then one judges there to be no power singularity. The authors were clear that this criterion is provisional, and depends on the grid refinement ratio $r \sim 2$, but they did test it on many cases, as already noted.

Logarithmic singularities

The local stress f for power singularities by definition behaves as

$$f = O(f_0 \ln R) \quad as \quad R \to 0 \tag{5.10.4.1.6}$$

The signature behavior of the power singularity during mesh refinement ($\Delta \to 0$) is then

$$f_3 - f_2 \sim f_2 - f_1 \sim f_0 \ln r \quad as \quad \Delta \to 0 \tag{5.10.4.1.7}$$

Successive estimates of the increment in stress δf are obtained as

$$\delta f_{23} = f_2 - f_3 \tag{5.10.4.1.8a}$$

$$\delta f_{12} = f_1 - f_2 \tag{5.10.4.1.8b}$$

which are useful only if they do not change sign during over the grid triplet, tested as

$$\delta f_{23}\,\delta f_2 > 0 \tag{5.10.4.1.9}$$

Then, one judges a log singularity to be present if these two successive estimates are approximately constant; again, the specific criterion tested is the change in the successive values less than 10% of the average value.

$$\frac{2\,|\,\delta f_{23}-\delta f_{12}\,|}{|\,\delta f_{23}+\delta f_{12}\,|}<0.1 \tag{5.10.4.1.10}$$

Implementation and Interpretation

Eqs. (5.10.4.1.1, -5 and -10) are all evaluated for the grid triplet. As stated, if Eq. (-1) and Eq. (-5) hold, then one judges a power singularity to be present. If Eq. (-1) and Eq. (-10) hold, then one judges a log singularity to be present. If none of them hold, then one terms the solutions non-convergent, meaning not converging over this grid triplet (or not *yet* converging).

Possible Masking of Singular Behavior

Typically with CSM stress singularities, other (non-singular, or regular) contributions to stress are negligible near the singular point, i.e. the non-singular stresses are $O(1)$ as $R \rightarrow 0$. [Likewise for dependent variables in singular problems of fluid dynamics.] However, Sinclair et al (2006) noted that occasionally CSM stress singularities "occur in concert with a hydrostatic pressure that can mask their presence." They stated that Eq. (-8) and Eq. (-10) can still be used to detect log singularities, but Eq. (-4) and Eq. (-5) require adaptation for power singularities. If the magnitude of hydrostatic pressure is known, it can be subtracted out and Eq. (-5) used. If not, one needs to run another computation on a fourth, finer grid to obtain f_{11} and to evaluate the following equations (again with appropriate limitations to avoid indeterminacies).

$$\gamma_{23} = \ln A / \ln r, \quad A = \frac{f_1 - f_2}{f_2 - f_3} \tag{5.10.4.1.11a}$$

$$\gamma_{23} = \ln B / \ln r, \quad B = \frac{f_{11} - f_1}{f_1 - f_2} \tag{5.10.4.1.11b}$$

Then Eq. (5.10.4.1.11) replaces Eq. (-4) in the evaluation of Eq. (-5), with the same interpretations. Even with this adjustment, Sinclair et al (2006) stated that it is possible for there to be masking of singular stresses to some extent.

Future Extensions

This approach has already been convincingly evaluated for CSM 2-D stress problems, and could be used as a template for new studies of singularity detection, the details of which would likely vary for different problems. An important generalization to be developed and tested would be grid refinement factors $1 < r < 2$.

5.10.5 Achieving the Asymptotic Range

The theory of Richardson Extrapolation, and therefore of the Grid Convergence Index, depends on the assumption that the Taylor series expansion (or at least, the definition of the order of the discretization implied by Eq. 5.3.4) is valid asymptotically, and that the two grids are within (or close to) the asymptotic range. For smooth elliptic problems, this is easy to achieve. (A 2nd-order accurate discretization of a Laplace equation with smooth boundary values is well behaved over virtually all discretizations.) For small parameter problems (e.g. Reynolds numbers >> 1) it is more problematic, and more than two grid solutions are required. The methodology presented herein does allow for detecting this situation in a straightforward manner, provided that the order of the method, p, is uniform.

If an exact solution is known to a model problem, we can monitor

$$E_b = \frac{\text{error}}{h^p} \tag{5.10.5.1}$$

as h is refined. Then the (approximate) constancy of E_p is a generally faithful verification of the order p and

an indication that the asymptotic range is achieved.[43] In the practical case wherein the exact solution is not known, we perform at least three grid solutions and calculate two GCI, from fine grid to intermediate grid (GCI_{12}) and from intermediate grid to coarse grid (GCI_{23}). Then the (approximate) constancy of $E_p \approx$ GCI $/(F_s\, h^p)$, or

$$GCI_{23} = r^p GCI_{12} \tag{5.10.5.2}$$

indicates that the asymptotic range has been achieved.

This indication that the asymptotic range has been achieved is usually faithful, in the author's experience (Roache, 1982; Roache et al, 1990; Roache and Knupp, 1993). But an exception (and unfortunately, an important one) occurs in problems with multiple scales of solution variation wherein a finer scale of the problem variation has been completely missed in the grid refinements. For example, in dual-continuum models of transport in porous media [44] the time scale for diffusion and storage in the material matrix blocks may be orders of magnitude less than the time scale for essentially advective transport in the fracture system. Time-step refinement may indicate no substantial change in the results (i.e., a false indication of convergence) if the time step is of the order of the advective time scale. Similar situations occur in turbulent boundary layer studies where some minimal viscous sublayer resolution is required (see Shirazi and Truman, 1989; Wilcox, 1993) and in chemically reacting flows which can have more time scales than species. Adaptive ODE solvers are good at detecting multiple time scales, but in multidimensional flows, at present there seems to be no substitute for an independent estimate (from theory or experiment) of the physical scales of interest.

Since so many journal authors are reluctant to perform even the most minimal grid convergence tests with two grids (Roache et al, 1986; Roache, 1990), it may seem unrealistic to recommend *three* grids as a matter of course. In fact, it is required to have an indication that the calculations are in the asymptotic range, if this is not already inferred from experience with a nearby calculation as in, e.g., Nguyen and Maclaine-Cross (1988) or Blottner (1990). In any case, the GCI provides an easily achieved improvement over the simplistic reporting of raw data on ε.

G. de Vahl Davis (1983), in his classic benchmark calculations of a buoyancy-driven cavity, indicated local convergence rates of *less* than 1st-order for the relatively coarse grids used, even though the method was asymptotically 2nd-order accurate. In the absence of such meticulous work as that of de Vahl Davis, the reporting of a GCI based on the assumed $p = 2$ would be preferable to simplistic reporting the raw data of ε, but if there is any indication of less-than-theoretical convergence rates, the more conservative estimate with GCI evaluated from Eq. (5.6.1) or (5.7.2) using $p = 1$ should be reported.

5.10.6 Δ Extraction of the Observed Order of Convergence From Grid Convergence Tests

If an exact solution is known or manufactured (see Chapter 3), it is straight-forward to extract the order of convergence (corresponding to p in Eq. 5.4.1) from results of a systematic grid convergence test using a minimum of two grid solutions. This serves to Verify a Code. However, it is also desirable to Verify the observed order for an actual problem, since the observed order of convergence depends on achieving the asymptotic range, which is problem dependent, and since the observed order may differ from the theoretical order, or from the order verified for a test case, for a variety of reasons. (See discussion in Westerink and Roache, 1995 and in Chapter 6.) The following general method was presented in Roache

[43] For examples, see Chapter 3, Section 3.5, or Richardson, 1927; Steinberg and Roache, 1985; Roache et al, 1990; Blottner, 1990; Roache and Knupp, 1993.

[44] WIPP PA Dept. (1992), Helton et al (1995, 1996)

(1995a), "Verification of Codes and Calculations." For an equivalent method with different notation (notably on the definition of the grid refinement factor) see Celik and Zhang (1995).

Blottner (1990) and others used graphical means, plotting the error on log paper and extracting the order from the slope. This procedure requires evaluation of the error itself, which is generally not known. If the finest grid solution is taken to be the reference value (unfortunately, often called the "exact" value, which it obviously is not) then the observed order will be accurate only for those grids far from the finest, and the calculated order approaching the finest grid will be indeterminate. Blottner (1990) improved on this by estimating the "exact" value by Richardson Extrapolation (see also Shirazi and Truman, 1989), but this procedure is somewhat ambiguous since the order is needed in order to perform the Richardson Extrapolation.

If the grid refinement is performed with constant r (not necessarily $r = 2$), the order can be extracted directly from three grid solutions, without a need for estimating the exact solution, following G. de Vahl Davis (1983, p. 254). With "1" being the solution on the finest grid in the present notation,

$$p = \ln\left(\frac{f_3 - f_2}{f_2 - f_3}\right) / \ln(r) \tag{5.10.6.1}$$

A generalization of this procedure, not restricted to constant r, is possible using the generalized theory of Richardson extrapolation. Eq. (5.10.5.2) may be used to Verify an assumed order p. (It is not necessary to use the GCI itself.) One calculates

$$\alpha = \frac{\mathrm{GCI}_{23}^{fine}}{\mathrm{GCI}_{12}^{fine}} \tag{5.10.6.2}$$

If $\alpha \approx r^p$, then p is the observed order. However, Eq. (5.10.6.2) requires r to be constant over the three grid set, and it cannot be used to calculate p directly since p is implicitly present in the GCIs. The more general procedure is to solve the equation

$$\frac{\varepsilon_{23}}{r_{23}^p - 1} = r_{12}^p \left(\frac{\varepsilon_{12}}{r_{12}^p - 1}\right) \tag{5.10.6.3}$$

for p. This is simple for r constant (not necessarily 2 or integer), giving

$$p = \ln\left(\frac{\varepsilon_{23}}{\varepsilon_{12}}\right) / \ln(r) \tag{5.10.6.4}$$

(Note that Eq. (5.10.6.4) differs from Eq. (5.10.6.1) only by higher order terms that depend on the non-dimensionalization of ε's.) But if r is not constant during the grid refinement, Eq. (5.10.6.3) is nonlinear in p. Usual solution techniques can be applied, e.g., direct substitution iteration, Newton-Raphson, etc. (even graphical). When considering the stability of the iteration, one should allow for observed $p < 1$. This can happen even for simple problems at least locally (de Vahl Davis, 1983). Unfortunately, behavior far away from asymptotic convergence can be non-monotone. Also, $r \sim 2$ will be easier to solve than $r \sim 1$, and $r >> 2$ is probably not of much interest. For well behaved synthetic cases which have been tested, direct substitution iteration with a relaxation factor $\omega \sim 0.5$ works well. With ρ = previous iterate for p, the iteration equations are

$$p = \omega\rho + (1-\omega)\frac{\ln(\beta)}{\ln(r_{12})} \tag{5.10.6.5a}$$

$$\beta = \frac{(r_{12}^{\rho} - 1)}{(r_{23}^{\rho} - 1)}\frac{\varepsilon_{23}}{\varepsilon_{12}} \tag{5.10.6.5b}$$

Note this form of the iteration gives the exact answer in one step for the case of r = constant and $\omega = 0$.

The slightly modified GCI of Celik et al (2008) eliminates indeterminacies caused by opposite signs of ε_{12} and ε_{23} but this can lead to a misplaced confidence when grid convergence is noisy.

Once p is known with some confidence, one may predict the next level of grid refinement r^* necessary to achieve a target accuracy, expressed as a target Error Estimate E_1 or GCI_1, call it GCI^*. With GCI_{23} being the value from Eq. (5.6.1) for the previous two grids,

$$1/r^* = \sqrt[p]{\frac{GCI^*}{GCI_{23}}} \tag{5.10.6.6}$$

This result, of course, depends only on the assumed definition of order of the discretization error, i.e. only on E_b = error / h^p, and not on the GCI theory itself. (If $F_s = 1$ is used in Eq. (5.6.1), then the GCI = Error Estimate E_1.)

For those special cases in which an exact solution is known (e.g. in Code Verification, or for special metrics like dilatation = 0 for incompressible flow) the observed p may be extracted from only two grid solutions by solving Eq. (5.4.1) for p.

$$\begin{aligned} d &= (f_1 - f_2)/(f_{exact} - f_1) \\ p &= \ln(1+d)/\ln(r) \end{aligned} \tag{5.10.6.7}$$

5.10.6.1 § Asymmetrical Grid Refinement[45]

As noted in Section 5.10.2, the extraction of p from grid refinement results does not work unless the refinement factor r is constant in all coordinates.

Eça and Hoekstra (2002b) demonstrated thoroughly that lack of strict geometric similarity in the grid sequence is a major contributor to noisy values of observed rate of convergence p. It is obvious that geometric similarity requires the same grid refinement factor in each coordinate. Salas (2006) investigated this effect in a systematic way, and disclosed a widespread mistake associated with it. By citing error estimation results from Workshops, he pointed out the prevalent practice of using the power series form for a 1-D problem (from Eq. 5.3.1),

$$f_e \sim f_c + c\,h^p \tag{5.10.6.1.1}$$

and using it in multidimensional problems in an erroneous way. Users calculate a grid refinement factor r as the ratio of representative grid spacings defined as $h = (h_x\,h_y)^{1/2}$ or possibly others, such as the diagonal $h = (h_x^2 + h_y^2)^{1/2}$. The form does not really matter, since the grid refinement ratio r scales out. However, that practice only makes sense if the same r applies in each coordinate. Otherwise, another coefficient is introduced for each coordinate, and it would require four grid solutions (rather than three) to determine observed p using the correct 2-D form (Eq. 1.4 of Salas (2006)).

[45] This Section is taken from Roache (2006).

$$f_e \sim f_c + a\, h_x^p + b\, h_y^p \qquad (5.10.6.1.2)$$

Salas claimed there is widespread misunderstanding of this effect, and that current practices are flawed, particularly in external aerodynamics.

To avoid confusion, let us refer to the observed p calculated using the 1-D form applied to a grid sequence with $r_x \neq r_y$ as observed pseudo-1-D p, denoted p_{1s}.

Using an exact solution for transonic flow (the Ringleb solution) Salas demonstrated that his theoretically second-order code exhibited observed $p = 2.2$ when $r_x = r_y$, but exhibited $p_{1s} = 9.94$ for a particular case with $r_x \neq r_y$. (See Eça and Hoekstra (2002b) for sometimes comparable anomalous results.)

Salas included a simple synthetic problem to illustrate his point. Upon examination, this problem raises another point that could be further investigated. It bears on the question of how one might use p_{1s} > theoretical p in error estimations.

For p to apply in a multidimensional problem, r must be the same in all directions, unless one solves for the coordinate coefficients separately. It is also true that *convergence* itself does not depend on this condition. Consider the computational solutions on all possible 2-D (I×J) grids, starting with a 2×2 cell grid, as follows.

2,2	2,3	2,4	2,5	2,6	2,7	2,8	2,9
3,2	*3,3*	**3,4**	3,5	3,6	3,7	3,8	3,9
4,2	4,3	*4,4*	4,5	**4,6**	4,7	4,8	4,9
5,2	5,3	5,4	*5,5*	5,6	5,7	**5,8**	5,9
6,2	6,3	6,4	6,5	*6,6*	6,7	6,8	6,9
7,2	7,3	7,4	7,5	7,6	*7,7*	7,8	7,9
8,2	8,3	8,4	8,5	8,6	8,7	*8,8*	8,9
9,2	9,3	9,4	9,5	9,6	9,7	9,8	*9,9* ...

Figure 5.10.6.1.1. All possible 2-D (I×J) grids. The grid doubling sequence is underlined. Symmetrical refinement with I = J proceeds along the *diagonal*. Any sequence that proceeds both down and to the right, such as the **bold font** path, also converges to the exact continuum solution.

The computational solutions f_g (I, J) would form a (discrete) single-valued solution surface above this discrete 2-D domain of definition. The exact (continuum) solution f_e is approached down and to the right (but not just down, and not just to the right, which are only one-coordinate refinements). The grid doubling sequence is underlined; it is not necessary to follow this path. The *preferred* paths are anything along the diagonal (*italics*), for which r is not necessarily constant in the sequence (along the diagonal), but it is the same in each direction. But if we took another path down and to the right, e.g. the **bold font** path, we would still be heading towards the exact solution f_e as (I, J) $\rightarrow (\infty, \infty)$.

The evaluation of an observed p enables us to extrapolate along the path, analogous to a directional derivative. (I suppose the analogy could be made precise by somehow generalizing the discrete surface to a continuum surface.) The extrapolated value can be used as a better estimate of the exact (converged) value f_e and so gives an error estimate. Any path along the diagonal corresponds to the same r in each direction. This diagonal direction does not uniquely determine the observed value of p because different values of r

give somewhat different p (except asymptotically). The grid doubling sequence gives a somewhat different value of the observed p than the unit sequence (2,2), (3,3), (4,4), (5,5), etc.[46]

The extrapolation slope is clearly path dependent, since the solution surface is not flat. Although any path along the diagonal corresponds to the same r in each direction, others are possible. If the observed p or the p_{1s} is not real, presumably it cannot be used, and we are far outside the asymptotic region. But if it is real, can it be used? If we are following the bold font path, we clearly do *not* want to use theoretical p, even if it is correct for the diagonal path. We want to use the slope appropriate for our path.

The question is: Will this extrapolation be accurate? Might it be that the somewhat arbitrary definition of h and therefore r, and the evaluation of a path-dependent p_{1s} , produces an accurate extrapolation when used with a consistent evaluation of the coefficient c of Eq. (5.10.6.1.1)?

It would be of interest to try it for the exact (Ringleb) solution of Salas (2006) but solution values were not presented (only L_1 errors in velocity). But when applied to the synthetic second-order accurate problem results given in Tables 1 and 2 of Salas (2006) it works very well indeed. The synthetic problem is devised with $f_e = 1$, $p = 2$, $a = 1$, $b = 5$ in Eq. (5.10.6.1.2). Using the same r in both directions ($r_x = r_y = 2.0$ for the first refinement, = 1.5 for the second) reproduces the "true" (diagonal path) value of $p = 2$, along with $c = 12.9$ and the exact solution value $f_e = 1$, as expected. If the same r is not used in each direction ($r_x = 1.6$ and $r_y = 2$ for both refinements), the three grid solutions produce $p_{1s} = 2.36$ and $c = 34.75$, very different from the diagonal path, as is to be expected; however, used consistently with Eq. (5.10.6.1.1) they produce $f_e = 0.999 \sim 1$. That is, extrapolation with the path-dependent observed p along that path is correct for the synthetic problem.

The conditions for use of p_{1s} to estimate error remain to be determined. In a real problem with other contributors to noise, we would still be reluctant to base an uncertainty estimate on some observed $p = 9.94$, and the consensus at the Lisbon Workshops (Eça et al, 2005, 2007a, 2009) was to enforce an upper limit on p to avoid too-optimistic uncertainty calculations. The suggestion to limit $p \sim$ theoretical may be unnecessarily conservative if correct extrapolation along the convergence path is performed.

5.10.6.2 § Consistent Quadrature

In another fundamental paper, Salas (2008) demonstrated a source of noise in calculation of observed order of convergence p. Calculation of functionals of solutions (e.g. aerodynamics drag) require quadratures. As noted in Section 5.3, Richardson Extrapolation requires that the order of the quadrature must be consistent with the order of the computational PDE method, i.e. at least 2nd order quadrature if the computational PDE method is 2nd order. But more subtly, Salas has shown that quadratures that are algebraically consistent with the discretization produce observed order in much more faithful agreement with theoretical order. His results were limited to constant grid spacing. For more general situations, it is recognized that higher order quadratures are more dependable than those just matching the discretization order.

On the other hand, when quadrature accuracy is adequate, it is common experience that solution functionals often (not always) converge more smoothly than point values. See, e.g., Salari and Roache (1990), Hay and Pelletier (2008) and Eça and Hoekstra (2008b,c).

[46] The unit change sequence (2,2), (3,3), (4,4), (5,5), etc. would produce a better approximation of a secant evaluation of a directional derivative, if we were envisioning a continuum surface of solutions. But in actual calculations, it would contribute to noise in p because of incomplete iteration error and round-off error, especially as $(I,J) \rightarrow (\infty,\infty)$.

5.10.6.3 § Misleading Convergence Rate

In Section 5.10.4.1, we cited the paper by Sinclair et al. (2006) on Computational Solid Mechanics (CSM) stress problems which, among other contributions, showed that the technique of judging convergence by observing the "linearly-increasing mesh sequence" is seriously misleading.

This linearly increasing mesh sequence uses a mesh sequence such as 100, 110, 120 total elements *irrespective of the dimensionality*, and convergence is judged (effectively extrapolating) by a log-log plot. However common it may be in FEA CSM, it has no basis in analysis. Nevertheless, it occasionally has been used in CFD as well. Its inadequacy is easily demonstrated by a synthetic problem that converges at first order in 1-D, with the exact discretization error E_{exact} on the *k*-th grid given by

$$E_{\text{exact}} = (f_k - f_{\text{exact}}) \approx g_1 \Delta x = g_1/\text{N} = g_1/(\text{I-1}). \qquad (5.10.6.3.1)$$

A plot of E_{exact} vs N will look like plot of a scaled function ζ *(*N*)* = 1/N, characteristic of a 1st order method. To disguise this 1st-order convergence, extend the dimensionality to 2-D with no variation in *y*, and plot E_{exact} vs N = (I-1)(J-1), noting that E_{exact} is still given by Eq. (5.10.6.3.1). The result will look like a plot of ζ *(*N*)* = $1/\text{N}^2$, characteristic of a 2nd order method, converging much faster. Go to 3-D for 3-rd order appearance.

5.10.7 Method of Characteristics and Spectral Methods

It is not clear how or if the GCI would be applicable to calculations obtained by the classic method of characteristics, as used in gas dynamics, because of the possibly discontinuous solutions and the irregular gridding. The various Modified Method of Characteristics (e.g., see references in Roache, 1992b or Chapter 13 of Roache, 1998b) will produce more systematic grid refinement, but the concept of "order" is more tenuous for the Flux-Based MMOC (as evidenced by the fact that the accuracy *improves* as the Courant number increases above 1), and the GCI may not be applicable. Similarly, for spectral and pseudo-spectral methods, and certainly for spectral elements, different extrapolation procedures would be required. It is not known at this time how well the GCI would apply or could be extended.

5.10.8 Non-Smooth Property Variation and the GCI

In aerodynamics problems, one typically deals with smooth property variations over modest ranges, and often with constant property problems. In groundwater flow and transport calculations, uncertainty and range are much greater, and sensitivity studies are often performed with Monte Carlo techniques used to generate property variations of orders of magnitude, even from one grid block (finite volume) to the next (e.g., see WIPP PA Dept., 1992; Helton et al, 1995,1996). Geologic layering produces discontinuous variations in properties of several orders of magnitude.

In these situations, it is not advisable to use non-integer grid refinement parameters *r*, because additional errors would be introduced by interpolation of properties. This confusion would be aggravated by the common use of harmonic averaging for properties (e.g., Roache, 1991, 1992a, 1993). Likewise, grid coarsening is not advisable if a coarsened grid would not resolve the scale of the property variations (often the case for expensive two-phase flow calculations, e.g. WIPP PA Dept., 1992; Helton et al, 1995,1996). The only approach applicable is a brute-force grid refinement by a factor of 2, which avoids any necessity for interpolation of properties.

The use of the GCI still has two contributions to such problems: (a) including the effect of the order *p* of the method in Eq. (5.6.1), and (b) economizing a consistent treatment of *further* grid refinement. That is,

a third grid (second refinement) need not involve the expense of another grid doubling (to a quadrupling of the base grid resolution) but can be done on a tripled grid, and reported consistently with the GCI. The difference in computer time between calculating the sequence (base h_0, 1/2 h_0, 1/4 h_0) and the sequence (base h_0, 1/2 h_0, 1/3 h_0) can be significant. Consider an *optimal* method with base-case computer time = T_2 in 2-D and T_3 in 3-D, and time-step resolution increased in proportion to the spatial grid resolution. The quadrupling sequence in 2-D costs $73T_2$ while the tripling sequence costs $36T_2$; in 3-D, the costs are $273T_3$ and $98T_3$, respectively. These savings of a factor of 2 in 2-D and somewhat less than 3 in 3-D will be greatly amplified if *sub-optimal* direct solution methods are used.

5.10.9 Non-Smooth Property Variation and Geostatistical Realizations

A more fundamental question arises when geostatistical methods are used to generate particular realizations of grid-block property variations with specified statistical parameters. Only the statistical results are of interest, not the solutions of the individual realizations. The question is then, should the grid refinement studies be performed separately from the geostatistical realizations? That is, should the solution of the partial differential equations be converged on finer grids with the assumed continuum property variation fixed at a geostatistically generated coarse-grid distribution, or should the geostatistical generation also change as the grid is refined?

This is not an easy question. Although definition of a fixed continuum problem for the grid refinement studies is conceptually easier, it is clear that substantial computer savings could accrue to the combined convergence approach. The results from both approaches should be statistically consistent, though not identical. In either approach, the grid increments should be less than (be able to partially resolve) the correlation length of the property variation.

5.10.10 Δ Iteration convergence

5.10.10.1 Δ Stopping Criteria for Iteration Convergence

Iterative methods are always required for nonlinear problems solved by implicit formulations and often are used as part of an explicit formulation as well. Before any discretization error estimation is calculated, it must be ensured that iteration convergence is achieved. Otherwise, the incomplete iteration error will pollute the error estimation and uncertainty estimation. The iteration error is more critical for the error estimation than for the solution itself, and Richardson Extrapolation amplifies incomplete iteration errors (Section 5.3).

The stopping criteria for iteration convergence is similar to grid convergence testing in one important aspect: both must be properly normalized. For an iterative solution procedure in which the new iterate value f^{k+1} is under-relaxed by a factor ω, that ω must appear in the denominator of the stopping criteria.[47] That is, many codes test something like

$$L_\infty = \max_{i,j} \left| \frac{f_{ij}^{n+1} - f_{ij}^{n}}{f_{norm}} \right| \le tol \qquad (5.10.10.1)$$

[47] For example, see Roache (1972b, p. 174 ff; 1975, 1995b, 1998b), Roache and Ellis (1975).

where n is the iteration level, f_{norm} is some suitable normalizing value (or just non-dimensionalizing value) and *tol* is some convergence criterion like 10^{-4} or 10^{-6}. This test can be met meaninglessly, regardless of the actual convergence, in a relaxation procedure such as

$$f^{n+1} = \omega f^{new} + (1-\omega) f^{old} \tag{5.10.10.2}$$

by choosing ω small enough. A simple modification of the convergence test is the form

$$\frac{L_\infty}{\omega} \le tol \tag{5.10.10.3}$$

For simple iteration schemes (such as SOR without immediate updating) it may be shown that this is algebraically equivalent to a criterion based on directly evaluating the residual.

Such iterative stopping criteria are still somewhat heuristic. Results by Eça and Hoekstra (2006a, 2007, 2009b) indicated that iteration error estimators based upon the last performed iteration (like Eq. 5.10.10.1) systematically underestimate the iteration error, a serious conclusion. They developed a method for estimation of iteration error based on extrapolating by geometric progressions; see Section 5.10.10.3. Another conclusion that contradicts common practice is that convergence of the L_2 norm is not a reliable indicator for iteration convergence [48], whereas the more demanding L_∞ norm is reliable.

What is intended is an estimate of the incomplete iteration error, defined as

$$\varepsilon_i^n = f^n - f^\infty \tag{5.10.10.4}$$

where the notation f^∞ suggests the limit of iterations. In the absence of accumulated round-off error, f^∞ would be the exact solution of the discretized equations (rather than the continuum equations). Roughly,

$$f^\infty \cong \lim_{n\to\infty} f^n \tag{5.10.10.5}$$

for convergent iterations, the approximation being due only to effects of machine round-off error. An estimate of incomplete iteration error in terms of the iteration change was given by Ferziger (1988, 1993) as

$$\varepsilon_i^n \cong \frac{f^{n+1} - f^n}{\lambda_1 - 1} \tag{5.10.10.6}$$

where λ_1 is the (assumed) *real* principal eigenvalue of the iteration matrix (defined by the iteration scheme), which can be estimated from the history of the iteration.

For a simple iterative solver like SOR applied to a discretized Poisson equation, the use of a relaxation factor ω less than the optimum value produces a real principal eigenvalue of the iteration matrix, making the normalizing of the stopping criteria easier to determine from the history of the iteration solution path. For SOR relaxation with larger ω values, and for more intricate iteration schemes (generally the more efficient ones), the principal eigenvalue may be complex. Ferziger and Peric (1996) presented a method for normalizing the stopping criteria in this case, based on some assumptions and approximations that were

[48] However, see Roy and Blottner (2003,2004) cited below.

justified by their test cases. These numerical experiments indicated that data from as many as 50 iteration steps must be used to produce a reliable normalization for the complex eigenvalue situation, but the amount of coding and the computational time penalty are insignificant compared to solving the discretized equations. This method is recommended for consideration, and in fact was a starting point for later methods to be described in Sections 5.10.10.3-4. However, Ferziger and Peric (1996) noted that simpler methods based on monitoring either the reduction of the residual or the difference between successive iterates ε (as above in Eq. 5.10.10.3) can also be used if normalized properly. (See Section 5.10.10.3 for a more recent method.)

A feasible approach for more efficient methods with complex eigenvalues would be to simply pause the iteration with the efficient method (e.g. GMRES, or SOR with optimum ω) and replace it for some number of iterations with another less efficient method with real eigenvalues of the iteration matrix (e.g. SOR with smaller ω) to estimate the iteration error. Since so many solutions published currently involve 10,000 - 100,000 iterations, the penalty for estimating iteration error periodically over (say) 100 relatively inefficient iterations would be small. The assumption here is that the incomplete iteration error at step n is roughly a "state variable" for any step in the solution procedure, i.e. it is not strongly path-dependent but just depends on the values of the (provisional) solution at n. If the iteration error estimation method is valid, it should provide a useful estimate for any late stage in the process. Thus, the methods to be described in Sections 5.10.10.3-4, which depend on real eigenvalues, could be used. This process would also serve to selectively reduce high-frequency components of discretization error (e.g. Roache, 1998b), improving RE estimates.

For a stopping criterion, a commonly used but unjustifiable rule of thumb is to require at least three orders of magnitude decrease in properly normalized residuals for each equation solved over the entire computational domain. Two studies by Roy et al (2003, 2004), using MMS to produce a benchmark exact solution, indeed show that L_2 norm "residual reduction tracks extremely well with global iterative error for a wide range of nonlinear flow problems" (cited in Roy, 2004). However, the scaling between residuals and iteration error is problem dependent and the prescribed three orders of magnitude reduction in residuals from their initial values based on poorly defined initial conditions is arbitrary. This criterion is used as a default in some commercial codes, but is demonstrably inadequate for many problems even for basic accuracy, without considering the added requirements of uncertainty estimation. For time-dependent simulations in which time accuracy is required (as opposed to just using time-dependent calculations to solve the steady-state equations), intra-time step iteration convergence at every time step should be checked, and example convergence trends should be documented for selected, critically important variables. The preferred approach is to reduce the iterative error to a level negligible compared to the discretization error. This does not necessarily require iteration to (nearly) machine zero.

5.10.10.2 § Interaction of Iteration Convergence with Discretization Error

Iteration error and its interaction with discretization error has been thoroughly studied by Eça and Hoekstra (2006a, 2009b) for one class of problems with a RANS turbulence model; there is no reason to assume that other problems are more benign. A method for estimation of iteration error based on extrapolation by geometric progressions was developed and justified, and applied to realistic turbulent flows. These results show that the iteration error needs to be 2-3 orders of magnitude smaller than the discretization error to guarantee a negligible influence. This is often assumed, though seldom demonstrated convincingly. If the uncertainty U_i contributed by the (estimated) iteration error is much less than U_d contributed by the (ordered) discretization error, then obviously we can take the numerical uncertainty U_{num} to be

$$U_{num} = U_d \tag{5.10.10.2.1}$$

where U_d = GCI or another numerical uncertainty estimate. If more care is taken and U_i is to be added, it is *not* adequate (conservative) to use RMS (Root-Mean-Square) addition, because the iteration error affects the results for discretization error, i.e. U_i and U_d are not uncorrelated, violating the underlying assumption of RMS addition. Rather, the two must be combined by less optimistic simple addition.

$$U_{num} = U_d + U_i \tag{5.10.10.2.2}$$

The same caveats that apply to discretization error convergence testing also apply to iteration convergence testing. Different variables, or the same variable evaluated at different locations, can iteratively converge at significantly different rates.

5.10.10.3 § Estimation of Incomplete Iteration Error and Uncertainty by Least-Squares

Here we present details of the method for estimation of iteration error and uncertainty developed by Eça and Hoekstra (2006a, 2007, 2009b) and further details on their general results already described in Sections 5.10.10.1-2. The estimation of numerical uncertainty was obtained using the Least Squares GCI (to be described in Section 5.11) but their results on iteration error are independent of this and are applicable to standard GCI or other uncertainty estimators.

The methodology employed in this study involves an excellent application of the Method of Manufactured Solutions (Chapter 3) outside of Code Verification. The earlier study on a real problem reached some of the same conclusions (e.g. that iteration error is usually underestimated) but the lack of an exact solution in the earlier study left "some room for dispute." Using MMS to provide a realistic exact solution mimicking turbulent boundary layer flow, and a numerical solution converged to machine accuracy (~ 15 decimal figures), reliable evaluations[49] of the iteration and discretization errors were made for intermediate numerical solutions at stages of the iteration convergence and/or grid convergence processes, as follows.

• The difference between an intermediate numerical solution on any grid and the exact solution gives the total numerical error e_{tm} (true[50], not estimated) on that grid.
• The difference between an intermediate numerical solution on any grid and the solution converged to machine accuracy on the same grid gives the iteration error e_{im} (true, not estimated) on that grid.
• The difference between the solution converged to machine accuracy on any grid and the exact solution gives the discretization error e_{de} (true, not estimated) on that grid.

With the true errors for the study problem, the estimators for iteration error and discretization error can be evaluated, as well as the method of combining them to obtain as estimate of total numerical error.

The Estimation Methods

The iteration error estimation is based on a geometric-progression extrapolation of the difference between consecutive iterations, using the iteration counter n as the independent variable. Many codes

[49] Assuming that ~ 15 decimal figure precision is sufficient to make round-off errors negligible.
[50] The use of the notation e_{tm} here departs from e used by Eça and Hoekstra (2009b).

(certainly multiphysics codes) involve several levels of iterative procedures; the iteration counter n refers to the outermost (or overall) iteration process.

The iteration error was estimated using norms of the change in the solution over the iterations. Two norms were considered. From previous experience (Eça and Hoekstra, 2006a) the principle focus was on the L_∞ norm of the variable change between consecutive iterations.

$$L_\infty(\Delta f) = MAX(|\, f^n - f^{n-1}\,|) \qquad 1 \le i \le N_P \tag{5.10.10.3.1}$$

where N_P is the number of grid nodes (or cells) and Δf is the iterative change in f which refers to any observed variable[51]. Besides this L_∞ norm, the authors also tested the L_{RMS} because it is often used as a convergence criterion.

$$L_{RMS}(\Delta f) = \sqrt{\sum_{i=1}^{N_P} (f^n - f^{n-1})^2 / N_P} \tag{5.10.10.3.2}$$

Values of $L = L_\infty$ or L_{RMS} at any iteration could be tried as iteration error estimator, but experience showed that none were reliable, especially when convergence was slow, so the authors adopted representations by a geometric progression

$$L(\Delta f)|_n = L(\Delta f)|_{n_0}\, 10^{-q(n_0 - n)} \tag{5.10.10.3.3}$$

where n_0 is the last iteration performed and q, related to the convergence rate, is determined from the ratio

$$\rho = \frac{L(\Delta f)|_n}{L(\Delta f)|_{n-1}} = 10^q \tag{5.10.10.3.4}$$

For convergence, it is required that $\rho < 1$ or $q < 0$. Eq. (5.10.10.3.3) is equivalent to

$$L(\Delta f)|_n = L(\Delta f)|_{n_0}\, e^{-B(n_0 - n)} \tag{5.10.10.3.5a}$$

$$B = \log_{10} q \tag{5.10.10.3.5b}$$

Taking the logarithm of Eq. (5.10.10.3.3) gives

$$\log[L(\Delta f)|_n] = \log[L(\Delta f)|_{n_0}] - q(n_0 - n) \tag{5.10.10.3.6}$$

Because of noisy convergence, it is not generally safe to solve this equation for q using only two consecutive iterations. The recommended procedure is to solve it in a least squares sense using the data of the last (m+1) iterations, where m remains to be determined. For economy of notation, we denote the logarithm terms by

[51] For reference to the original papers, note that the original notation φ has been changed here to f for consistency with previous Sections.

$$\Psi_n = \log_{10}\left(L(\Delta f)\big|_n\right), \quad \Psi_0 = \log_{10}\left(L(\Delta f)\big|_{n_0}\right) \tag{5.10.10.3.7}$$

For a least squares fit, both q and Ψ_0 are to be determined. We distinguish between q and its least squares fit, denoted by Q, and again for economy of notation denote the least squares fit of the logarithm term Ψ_0 by Ψ (without subscript). The fitting procedure gives[52]

$$Q = \frac{4\sum_{n=n_0-m}^{n_0}(n-n_0)\Psi_n + 2m\sum_{n=n_0-m}^{n_0}\Psi_n}{4\sum_{n=n_0-m}^{n_0}(n-n_0)^2 - m^2(m+1)} \tag{5.10.10.3.8}$$

$$\Psi = \frac{\sum_{n=n_0-m}^{n_0}\Psi_n}{m+1} + \frac{m}{2}Q \tag{5.10.10.3.9}$$

When the convergence is far from smooth, these fits Q and Ψ may be corrected to Q^c and Ψ^c using the standard deviation of the fit D_f as follows.

$$D_f = \sqrt{\frac{\sum_{n=n_0-m}^{n_0}\{\Psi_n - [\Psi + Q(n-n_0)]\}^2}{m}} \tag{5.10.10.3.10}$$

$$\Psi^c = \Psi + D_f \tag{5.10.10.3.11}$$

$$Q^c = Q + 2D_f / m \tag{5.10.10.3.12}$$

These corrections add the standard deviation of the fit to the logarithm of the norm at iteration n_0 and the slope is corrected subtracting and adding D_f at the extremes of the interval used for the fit, $[n_0 - m, n_0]$ (Eça, 2009b). (These corrections were included in the subject studies.) For $\rho < 1$ (convergent), the sum of all terms of the geometric progression Eq. (5.10.10.3.3) with $n \geq n_0$ is then

$$e_i = \frac{10^{\Psi}}{1-10^{Q}} \tag{5.10.10.3.13}$$

which is the desired extrapolated estimate of iteration errors, either $e_{i\infty}$ for $L = L_\infty$ or e_{iRMS} for $L = L_{RMS}$.

Not surprisingly[53], these estimates of the iteration error are not reliably conservative. A candidate for a $U_{95\%}$ uncertainty estimate for iteration error, denoted by U_i, is suggested by the GCI studies for discretization uncertainty, using a Factor of Safety = 1.25.

In summary, true iteration errors e_{im} were compared to the four estimated quantities $e_{i\infty}$, e_{iRMS} , $U_{i\infty}$, and U_{iRMS} , where

[52] These formulas are corrected from Eça and Hoekstra (2009b) by Eça (2009b).
[53] See Section 5.14.2.

$$e_{i\infty} = L_{\infty}(\Delta f) \quad (5.10.10.3.11a)$$

$$e_{iRMS} = L_{RMS}(\Delta f) \quad (5.10.10.3.11b)$$

using Eq. (5.10.10.3.13), and

$$U_{i\infty} = 1.25\, e_{i\infty} \quad (5.10.10.3.11c)$$

$$U_{iRMS} = 1.25\, e_{iRMS} \quad (5.10.10.3.11d)$$

The value of (m+1), the number of iterations over which to perform the least squares fit for the geometric progression, was determined by experimentation in the study. Larger m give better fits, provided that the initial erratic convergence behavior is avoided. The recommendation arrived at by empirical study is to not begin counting the sequence until a partial iteration convergence is reached using $e_t \leq 10^{-2}$ (see Eq. 5.10.10.3.12 below).

The Study

The authors used two measures of quality to evaluate the estimators e_i (referring to either $e_{i\infty}$ or e_{iRMS}) and U_i (referring to either $U_{i\infty}$ or U_{iRMS}).

- The ratios between the estimators e_i or U_i and the true iteration error e_{im}, i.e. e_i / e_{im} and U_i / e_{im}, desired to be > 1.
- The percentage F of grid nodes where the estimators e_i or U_i fail to be conservative, i.e. where $e_i < e_{im}$ or $U_i < e_{im}$, desired to be $F \sim 5\%$.

An optimal error estimator would give $e_i / e_{im} \sim 1$ everywhere. Although $U_i / e_{im} > 1$ is desirable to accomplish conservatism, values >> 1 would suggest excessive conservatism.[54] The value of e_i / e_{im} varies greatly over the solution field for a turbulent boundary layer, so "the maximum value may not be very significant" and the minimum value over the grid was monitored.

The study problem was an exact manufactured solution (Eça et al, 2007a,b) resembling a turbulent boundary layer modeled with the baseline Wilcox k - ω model (Wilcox, 2006). (The general MS developed is capable of accommodating several turbulence models.) The continuity equation is satisfied identically. For isolating the sources of numerical errors, the turbulence quantities can be evaluated from the MS for use in the numerical solution, or more realistically the turbulence quantities can be evaluated from the numerical solution.[55] The code used was PARNASSOS, which uses theoretical 2nd order discretization for all terms except convection terms, which are 3rd order (including the transport equations for turbulence terms). Options for flux limiters were not used. The quantities examined were the cartesian velocity components u_x and u_y and the pressure coefficient C_p. The code and the MS have been applied in various types of grids including non-orthogonal boundary-fitted grids (Eça and Hoekstra, 2004, 2006a, 2008b) but the subject study was based on orthogonal stretched cartesian grids. The stretching parameter in the boundary layer direction was 0.05, that is, the first cell dimension was 0.05 that of an equispaced grid.

The grid set included 18 geometrically similar grids covering an overall grid refinement factor of 6.67. The finest grid was 401×401 and the coarsest was 61x 61, producing 19x19 physical locations that are common to all the grids, avoiding the ambiguities of interpolation.

[54] See Section 5.15.

[55] Only the latter results are given in Eça and Hoekstra (2009b). Both results are given in Eça and Hoekstra (2007).

The exact solution from MMS was not used for an initial estimate, since this would have obscured the behavior of iteration error.

The internal convergence criterion for the (outer, or overall) iteration[56] is specified by the input parameter e_t, stopping when the maximum (over the grid) changes all satisfy the following.

$$(\Delta u_x)_{max} < e_t, \quad (\Delta u_y)_{max} < e_t, \quad (\Delta C_p)_{max} < 0.1\ e_t \qquad (5.10.10.3.12)$$

A key methodology of the study is to vary e_t and store all the flow fields for examination. 13 values of e_t were used, the smallest $e_t = 0.5 \times 10^{-14}$, corresponding to machine accuracy, and the remaining 12 values = 10^{-N} for N = 1, 2, 3, ...12. So the entire flow fields were stored and examined for all 18 grid sets and 13 values of e_t. This is obviously not intended to be a model of industrial studies, but rather a research study for reliable evaluation of methods for estimation of iteration error that can be used in industrial studies without such exhaustive work.

The Study Results

Highlights of the study results have been given in Section 5.10.10.1-2, but are repeated here with additional details. For more complete results in tabular and graphical form and much additional discussion, see Eça and Hoekstra (2007, 2009b).

Not surprisingly, some irregularities were noted for the initial iterations in the finest grids. The results for crude iteration convergence criteria ($e_t = 10^{-1}$ to 10^{-4}) are probably not of much interest. The useful criterion developed is to use turn-off criteria of $e_t \leq 10^{-5}$ and to not begin the least-squares fit of the geometric progression Eqs. (5.10.10.3.7-10), i.e. determining (m+1), until the criterion of Eq. (5.10.10.3.12) is met to a tolerance of at least 10^{-2}.

The L_∞ norm (with the geometric-progression extrapolation) produced the most reliable error estimator. The L_{RMS} norm was not a good error estimator. Both e_{iRMS} and U_{iRMS} fail essentially for Cp.

The percentage of cases where $e_{i\infty}$ failed to be conservative is < 5% "for almost all combinations" of grid spacing and e_t but there were a few exceptions for u_y. For those cases, $U_{i\infty}$ was conservative except for some easily explained cases (unreasonable $e_t = 10^{-1}$, or the coarsest grid with "strange behavior of the convergence rate.") The minimum value of U_i / e_{im} was > 1 but < 10 for most of the cases, "an important result because it shows that U_i does not overestimate the iterative error by orders of magnitude."

Both the L_∞ norm and the L_{RMS} norm failed to be reliable when the geometric-progression extrapolation is replaced with the commonly used change over the last iteration, even with the Factor of Safety. The L_∞ norm was the better choice of the two, but its performance degraded with improving grid refinement. The L_{RMS} norm was "completely inadequate."

The discretization error studies showed that estimated U_d generally increased (roughly linearly) with iteration error, though this was not guaranteed. As expected from diverse experience, e_t had a drastic influence on observed convergence rate p_{obs}. For tight convergence criteria (small e_t) p_{obs} was monotonic for all three flow variables in close to 100% of the locations. For loose convergence criteria p_{obs} became erratic. This confirms again that sufficient iteration convergence is required for Richardson Extrapolation (see Section 5.3). The more significant lesson is that many cases of erratic convergence described in publications are due to sloppy numerical work, specifically, inadequate iteration convergence, rather than deficient discretizations. The papers may be consulted for additional details. Probably the most important conclusion regarding iteration and discretization errors is the rule of thumb that the estimated iteration error

[56] Details on what this entails in the PARNASSOS code are given in Eça and Hoekstra (2007, 2009b).

e_i should be reduced 2 to 3 orders of magnitude below the estimated discretization error e_d in order to have a negligible effect on estimated U_d.

Finally, the study showed that the usual statistical assumption of RMS combination of e_i and e_d is not justifiable. This failed to be conservative in more than 5% of the case, and in most of the cases by more than 50%. Significantly, this failure rate is not appreciably improved by higher grid resolution or tighter iteration convergence. This suggests that the fundamental assumption of independence of e_i and e_d is not correct, regardless of good estimates for the individual components.[57] However, simple addition as in Eq. (5.10.10..2.2) gave zero failures in all combinations tested.

It will be easy to make an excuse to ignore these important results because they are limited to one problem, but it is unlikely that many researchers will be willing to perform such a thorough study. Probably the best to be expected is that conscientious modelers will follow these guidelines and report successes and failures anecdotally.

5.10.10.4 § Alternative Estimation of Incomplete Iteration Error

Formally, the least squares iteration estimation method described above could be extended to f = computed functionals, which would not involve summations over nodes; this extension has not yet been tested but would seem to surely work.

Roy and Blottner (2000, 2001) earlier developed a method for estimation of incomplete iteration error and applied it to solution functionals (e.g., surface heat flux) rather than solution norms. Their method was based on similar assumptions but did not incorporate a least squares fit of the assumed parameters, so it is probably not as robust as the least squares approach of Eça and Hoekstra (2009b) but is simpler to derive and has been applied[58] convincingly to solution functionals in difficult problems (2-D hypersonic flows with 1- and 2-equation turbulence models). The method was derived for time iteration in a time-accurate code but of course is applicable to any roughly time-like iteration with real eigenvalues of the iteration matrix over a (continuum) iteration parameter τ. The iteration error ε^n at quasi-time step n is expressed as

$$f^n = f^\infty + \varepsilon^n \tag{5.10.10.4.1}$$

where the notation f^∞ is used again to suggest the limit of long-time iteration. Neglecting round-off error accumulation, f^∞ would be the exact solution of the discretized equations, not the continuum equations. The observed iteration convergence roughly fits the empirical description of exponential decay in late time τ,

$$\varepsilon^n = \alpha e^{-\beta\tau} \tag{5.10.10.4.2}$$

These two preceding equations are combined to give

$$\beta\tau = \ln\alpha - \ln(f^n - f^\infty) \tag{5.10.10.4.3}$$

This equation is applied to three consecutive time steps. The three equations are used first to eliminate α, and the final solution for f^∞ is facilitated by assuming constant time increments. The result is

[57] The same is true for combinations of outflow boundary error estimates and discretization error estimates; see Section 6.10.2.

[58] They did not address the issues of interaction of iteration error and discretization error.

$$f^{\infty} = \frac{(f^{n} - \Lambda f^{n-1})}{(1-\Lambda)} \tag{5.10.10.4.4}$$

where

$$\Lambda = \frac{(f^{n+1} - f^{n})}{(f^{n} - f^{n-1})} \tag{5.10.10.4.5}$$

The incomplete iteration error estimate is then

$$\varepsilon^{n} = -\frac{(f^{n+1} - f^{n})}{(1-\Lambda)} \tag{5.10.10.4.6}$$

and the % error (relative to the estimated fully iteration converged value) is

$$\%\ error\ f^{n} = -100\frac{(f^{n+1} - f^{n})}{(f^{n} - \Lambda f^{n-1})} \tag{5.10.10.4.7}$$

with the usual warnings about normalization and dangers of dividing by near-zero. The parameter Λ corresponds in the work of Ferziger and Peric (1985) to the magnitude of the largest eigenvalue (spectral radius) of the iteration matrix, limited here to real Λ.

Consideration of Eq. (5.10.10.4.5) reveals that it is delicate, with $\Lambda \rightarrow 0/0$ indeterminacy as iteration convergence is approached. $\Lambda < 1$ is required for convergence; in the continuum quasi-time parameter τ this means $\partial^2 f/\partial\tau^2 < 0$. The indeterminacy problem will be worse for small time steps. This is the same behavior for iteration error estimation as the problem with grid refinement factor $r \rightarrow 1$ in grid convergence studies. Here it may be mitigated by considering "*n*" to refer to an iteration macro-step, e.g. over perhaps 10-100 inner iterations or τ-steps, with the same kinds of trade-offs discussed in Section 5.10.2 for $r \rightarrow 1$. Even with the basic approach of Eq. (5.10.10.4.5-7), the method proved to be useable in difficult problems (Roy and Blottner, 2000, 2001, 2003).

5.10.11 § Discretization error estimation from a grid triplet without explicit evaluation of *p*

The AES method of Elizalde-Blancas et al. (2008) utilizes the idea that the extrapolation to f_{exact} can be accomplished (in some cases) without explicitly evaluating observed *p*. [See also Celik and Zhang(1995) and Celik and Li (2005).] Following their approach, but assuming a well behaved problem with constant *r* and *p*, the procedure for Richardson error estimate E_1 leads to an interesting result (Roache, 2008b). The errors E_1 for the fine grid solution f_1 , and E_2 for the medium grid solution f_2 , are evaluated using the dimensional form of Eqs. (5.4.2 - 3).

$$E_1 \equiv f_1 - f_{exact} = [f_2 - f_1] / [r^p - 1] \tag{5.10.11.1}$$

$$E_2 \equiv f_2 - f_{exact} = [f_3 - f_2] / [r^p - 1] \tag{5.10.11.2}$$

Subtracting the first equation from the second leads to

$$1/[r^p - 1] = [f_2 - f_1] / [f_3 - 2f_2 + f_1] \tag{5.10.11.3}$$

This leads to a formula for the error estimation that does not involve r or p (although this could be misleading, since constancy of r and p are necessary requirements). Using (5.10.11.3) in (5.10.11.1) gives

$$E_1 = [f_2 - f_1]^2 / [f_3 - 2f_2 + f_1] \tag{5.10.11.4}$$

Using the ε notation of Eq. (5.4.3b) (dimensional) with

$$\varepsilon_{21} = f_2 - f_1 \text{ and } \varepsilon_{32} = f_3 - f_2 \tag{5.10.11.5}$$

then (5.10.11.4) becomes

$$E_1 = \varepsilon_{21}^2 / \ [\varepsilon_{32} - \varepsilon_{21}] = \varepsilon_{21} / \ [(\varepsilon_{32} / \varepsilon_{21}) - 1] \tag{5.10.11.6}$$

The latter form is better behaved for round-off error, and allows a confidence check to see if the results are consistent with expected convergence behavior. In the usual method, the observed p is compared to the theoretical p. As stressed by M. Hoekstra (Eça and Hoekstra, 2008), agreement for a single grid triplet is no guarantee of monotone behavior, but it is comforting, and more confidence is built with more grid triplets. In the present approach, we would look for $[(\varepsilon_{32} / \varepsilon_{21}) - 1] \sim [r^p - 1]$ using known r and expected (theoretical) p.

Eq. (5.10.11.6) has been confirmed on synthetic problems, for which it is exact. For the well behaved first problem in Section 6.12, the method gives a factor of 2.956 when the assumed $r = 2$ and $p = 2$ would give a factor of 3. The GCI calculation with $p = 2$ shows that the GCI ratio (for $r = 2$), which should = 4 for these assumptions, is actually 3.95. So both the GCI calculation with explicit evaluation of p, and Eq. (5.10.11.6), are consistent with observed p very slightly less than 2. For the airfoil problems in Section 6.9 (turbulent flow) and Section 6.14 (inviscid), the convergence behavior is noticeably not ideal but fairly well behaved, and the p - free extrapolation works as well as the usual method with explicit evaluation of p.

5.11 § LEAST SQUARES GCI

Even when convergence behavior is far from ideal, the GCI can still provide reliable error bars if $F_s = 3$ is used. However, tighter error bars often can be obtained using $F_s = 1.25$ and using a least-squares approach developed by Eça and Hoekstra[59] to evaluate observed p. In bad cases, both $F_s = 3$ and least squares may be needed, as well as replacement of ε by the data range, and some other refinements, as described below. (Applications will be cited in Chapters 6 and 8.) First, we will consider tests to characterize the grid convergence behavior.

5.11.1 § Characterization of Apparent Grid Convergence Behavior

The grid convergence sequence of solutions is not always monotone. Causes of oscillatory convergence will be discussed further in Section 8.1; these include inadequate coarse grid resolution (being outside the asymptotic range), mixed order discretization, multi-block grid generation, shocks, interface tracking, etc. The following 3-grid test for observed (or apparent) convergence *type* [60] is based on ratios of successive

[59] Eça and Hoekstra (2000a,b, 2002a,b, 2004, 2007, 2008, 2009a, 2009b), Eça et al (2005,2007,2009), Raven et al (2002).

[60] Expanded from Stern et al, 2001; see also Roache, 2003 and Pelletier and Roache, 2006.

differences of solution values. With subscripts 1,2,3 referring to fine, medium and coarse grid solutions, we calculate the discriminating ratio *R* and recognize four *apparent* convergence conditions.

$$R = [\, f_1 - f_2 \,]/[\, f_2 - f_3 \,] \tag{5.11.1.2a}$$

(i)	Monotone convergence for $0 < R < 1$	(5.11.1.2b)
(ii)	Oscillatory convergence for $R < 0$ and $\|R\| < 1$	(5.11.1.2c)
(iii)	Monotone divergence for $R > 1$	(5.11.1.2d)
(iv)	Oscillatory divergence for $R < 0$ and $\|R\| > 1$	(5.11.1.2e)

The issue of possible misinformation has been discussed hypothetically by Coleman et al (2001) and in real calculations by Eça and Hoekstra (previously cited). Depending on the sampling from just a 3-grid sample, an actual oscillatory convergence can possibly appear to be either oscillatory, monotone diverging, or monotone converging. Moreover, an oscillatory *diverging* sequence can possibly appear likewise (Roache, 2003). Actually, the only conclusive 3-grid test result is that demonstrating oscillation, with no indication of it being oscillatory diverging or oscillatory converging. As a practical matter, such behavior is usually detected during exploratory calculations. Hypothetically, with nonlinear chaotic solutions a possibility, any kind of non-regular solution sequence is conceivable. The only way to *rigorously* determine convergence would be to perform a complete grid sequence, e.g. 51×51, 52×52, 53×53, ... 98×98, 99×99,... Not only is this economically infeasible, it would fail because of corruption by round-off error and incomplete iteration error. As a practical matter, as the number of grid triplets increases, if the observed *p*'s stabilize, and especially if these are close to the expected theoretical rate (e.g. $p = 1.96, 2.01, 1.93...$ for a nominal second-order method) confidence increases.

5.11.2 § Noisy and Degraded Convergence Rates

Even if convergence is monotone (i.e. $0 < R < 1$ in Eq. 5.11.11), the observed convergence rate *p* still can be noisy. If the observed *p* is indeed close to the theoretical *p* for the method, e.g. observed $p = 1.97$ for a nominally second-order method, one may proceed with some confidence. However, it must be remembered that a variety of factors can cause noisy *p*, i.e. a different grid triplet can produce a different observed *p*. Furthermore, sampling of the possible grid triplets could produce a misleading observed $p \sim 2$, when in fact more complete calculations would show considerable noise (or even non-convergence). Nevertheless, in the spirit of the targeted 95% certainty, if the observed *p* is close to 2, one may proceed with some confidence. A more scrupulous approach is to verify that the observed *p* is ~ constant by calculating *p* for at least two separate grid triplets. This requires performing a minimum of 4 grid calculations, which would allow as many as 4 grid triplets and observed *p*'s. Note that 4 grids (a,b,c,d) give 4 possible grid triplets of (a,b,c), (a,b,d), (a,c,d), (b,c,d). However, Eça and Hoekstra (2000a, 2002a) limit their grid triplets to $r \le 2$, which may eliminate some triplets from a 4 grid set.

The consequences of noisy *p* should be kept in perspective. Noisy *p* does not necessarily indicate an unstable algorithm or divergence. The methods and even the grid resolution may be adequate for accuracy. Noisy *p* just makes the error estimation and uncertainty estimation somewhat problematical.

Another contributor to noisy observed *p* or simply degraded *p* (e.g. observed $p \sim 1.2$ for a theoretically second order method) is the use of any kind of interface tracking or moving boundaries and/or re-meshing algorithms. These can be a challenge to achieving and convincingly demonstrating second-order convergence rates even in the Code Verification stage. Likewise, the presence of singularities can degrade the observed convergence rate, even though the coding may be demonstrably error-free.

When observed p over 4 over more grids is far from constant, the recommended procedure is to use a least-squares determination of effective p.

5.11.3 § Evaluation of Observed p by Least Squares

The evaluation of observed p by least-squares may seem to be a natural extension for noisy convergence, but few have attempted it. Eça and Hoekstra (previously cited) have developed it fully. It requires much testing to evaluate the procedures and justify the F_s in applications. They took an exhaustive look at grid convergence for several problems from laminar 2-D driven cavity to 3-D turbulent free-surface flows, using as many as 24 grid sets. i.e. grid triplets. (They are serious about quantification of uncertainty.)

They demonstrated that grid convergence can be remarkably consistent with theory for simple problems (the well-behaved Laplace problem, for which virtually any grid is within the asymptotic regime, gives observed p = 2.00) but for realistic CFD problems (3-D RANS solutions for the Wigley Hull and the KVLCC2 tanker) convergence is often not monotone and the observed p often involves significant scatter (noise) and is undependable. Chance grid sets may show observed $p \sim 2$, but other nearby sets fail. This is not unique to their problems or codes, but is (we believe) representative of CFD and possibly computational PDEs in general.

The authors showed that a major contributor to this noise is the difficulty of attaining geometric similarity of the grids with non-integer grid refinement and especially multi-block grid generation. The latter appears to be an unavoidable limitation. Without strict geometric similarity, the grid refinement factor r is not defined strictly. Numerical interpolation and/or quadrature is also a contributor. There are two very positive conclusions. Turbulence modeling is not a contributor, if switching functions are not used. And, the Reynolds Number does not have a significant effect on the intensity of the scatter in observed p.

For such problems with resulting data scatter in observed p, Eça and Hoekstra developed the least-squares procedure, requiring a minimum of 4 grid solutions, for determination of effective convergence rates, which provide improved error and uncertainty estimation for the difficult problems. For realistic problems, more than the minimum 4 grids may be necessary; they obtain (Raven et al, 2002) "fairly stable results using about 6 grids with total [overall] refinement ratio near 2." The later incarnations of the method limit the p used, since it clearly would be imprudent to calculate GCI with observed $p >>$ theoretical p. Although such superconvergence can occur, and would be appropriate to use if one were actually using the extrapolated solution (Section 5.4), it is usually not real but rather an artifact of noisy convergence and sampling. Observed $p >$ theoretical p is generally an unreliable result, and it is recommended for uncertainty calculations that max $p \sim$ theoretical p be used. On the other hand, there seems to be no reason to categorically reject 0 < observed p < 1. If observed p is < 1, it probably means that the coarsest grid is somewhat outside the asymptotic range, and the resulting uncertainty estimate of the GCI may be overly conservative. This is not an impediment to publication or reporting.

Eça and Hoekstra (2002a) applied the least squares approach to several models of convergence including the one-term expansion with unknown order p considered herein. Other possibilities considered were one-, two- or three-term expansions with fixed exponents. (For example, a two term expansion with p's = 1 and 2 could be appropriate for mixed-order discretization arising from first-order advection terms and second-order diffusion terms (Roy, 2003), or perhaps directional bias.) The simplest method works as well, and is recommended, as follows. (The notation is that of Pelletier and Roache, 2006.) The assumed one-term expansion of the discretization error is

$$f_i - f_\infty \cong \alpha \Delta_i^p \tag{5.11.3.1}$$

The least squares approach is based on minimizing the function[61]

$$S(f_\infty, \alpha, p) = \sqrt{\sum_{i=1}^{Ng} \left[f_i - (f_\infty + \alpha \Delta_i^p) \right]^2} \tag{5.11.3.2}$$

where the notation f_∞ suggests the limit of fine resolution (in the absence of round-off and iteration errors). Setting the derivatives of S with respect to f_∞, α, p equal to zero leads to

$$f_\infty = \frac{1}{Ng} \left\{ \sum_{i=1}^{Ng} f_i - \alpha \sum_{i=1}^{Ng} \Delta_i^p \right\} \tag{5.11.3.3a}$$

$$\alpha = \frac{Ng \sum_{i=1}^{Ng} f_i \Delta_i^p - \left(\sum_{i=1}^{Ng} f_i \right) \left(\sum_{i=1}^{Ng} \Delta_i^p \right)}{Ng \sum_{i=1}^{Ng} \Delta_i^{2p} - \left(\sum_{i=1}^{Ng} \Delta_i^p \right) \left(\sum_{i=1}^{Ng} \Delta_i^p \right)} \tag{5.11.3.3b}$$

$$\sum_{i=1}^{Ng} f_i \Delta_i^p \log(\Delta_i) - f_\infty \sum_{i=1}^{Ng} \Delta_i^p \log(\Delta_i) - \alpha \sum_{i=1}^{Ng} \Delta_i^{2p} \log(\Delta_i) = 0 \tag{5.11.3.3c}$$

The last equation is nonlinear and is solved iteratively by a false position method for observed p . The number of grids Ng must be > 3, and Eça and Hoekstra consider only $0 < p < 8$. As noted previously, for use in calculating an uncertainty estimator as in the GCI, they further recommend restricting max p used to theoretical p. Another feature follows.

5.11.4 § Replacement of ε by Data Range

The least-squares version of the GCI (LS-GCI) primarily involves least-squares evaluation of observed p, as described above. A later feature developed by Eça and Hoekstra is the replacement of ε of Eq. (5.4.3), the change in the quantity of interest over the last two grids in the sequence or ε_{12} , by the data range Δ_M over a range of grids. These are somewhat subjectively defined as all the grids calculated (Ng above) or those grids that seem to be in the acceptable range (≤ Ng), or just the last two grids, in which case $\varepsilon = \Delta_M$. This produces a more conservative uncertainty estimate, and is recommended for difficult cases.

However, what is definitely not recommended are *ad hoc* attempts to simplify the task of uncertainty estimation by replacing the GCI with an uncertainty estimator = $F_s \times \Delta_M$ (or simply Δ_M or ε). To do so neglects the important functional dependence of $1/(r^p - 1)$. Neither Δ_M nor ε are ordered error estimators, and ε can be made (almost) arbitrarily small by choosing r close to unity, limited only by $r = 1 + 1/N$. (See Roache, 2008b for further discussion.) To give a specific demonstration of just how serious this mistake could be, consider a well behaved problem with $p = 1$, and a 100^D grid. Calculate the change in the solution ε for a change in the grid to 99^D. (It could be done in principle: see Table 5.10.1.1.) Richardson Extrapolation shows that the ordered error estimate is

[61] Wood and Kleb (2002) have also developed software for least-squares determination of observed p.

$$\varepsilon /[r^p - 1] = \varepsilon / [(100/99) - 1] = 99 \times \varepsilon \qquad (5.11.14.1)$$

So ε alone would be unconservative by a factor of 99. Some estimator !

5.11.5 § Choice of *Fs* for the Least Squares GCI

Even with these later features of the LS-GCI (limitation of p and replacement of ε by Δ_M) poorly behaved problems (notably RANS turbulence models outside the asymptotic range) do not produce reliable uncertainty estimates with F_s = 1.25. (Examples will be cited in Chapters 6 and 8.) Consistent with the recommendations in Section 5.9.2, the simple solution is to use F_s = 3, resulting in a larger and somewhat unpalatable uncertainty estimate, when the observed p is suspicious, e.g. p > 2.2. But there is one more difficulty. Eça and Hoekstra (2008b) noted that this recommendation works well for a single case and a single verification quantity. But for a range of cases, e.g. base pressure variation over Reynolds number, the test may give different results for different ranges. Even for one problem, different verification quantities (e.g. velocity components) can produce slightly different p's. Though not a problem in itself, the difficulty is that an insignificant variation in observed p (over location or parameter variation) can cause a jump in F_s from 1.25 to 3 and an unreasonable jump in uncertainty estimate. Obviously one could blend the change in Fs, but this introduces additional empirical parameters that can only be justified, practically speaking, on small scale *ad hoc* studies. This final aspect remains somewhat unsatisfactory at this time; the analyst can choose between the (perhaps) overly conservative F_s = 3, or tighten the uncertainty estimate but possibly introduce disconcerting jumps as F_s may switch from 1.25 to 3, for these poorly behaved problems. The preferred route is to do the work to achieve the asymptotic range.

5.11.6 § Further Refinements and Summary for the Least Squares GCI

Eça and Hoekstra developed further refinements to the Least Squares GCI. A good summary and evaluation is contained in Eça and Hoekstra (2009b) using MMS to produce a realistic exact solution for turbulent boundary layer flow using a k - ω turbulence model. The paper also carefully examines incomplete iteration error (as described in Section 5.10.10.2) and its interaction with discretization error estimates.

The basic LS-GCI method uses Eqs. (5.11.3) to evaluate observed p by least squares. For a nominally second-order method, if observed convergence is monotonic and this observed p is between (roughly) 1 and 2, the GCI is used with F_s = 1.25, i.e. GCI = 1.25×E where E is the Richardson error estimator. But if observed p < 1 (roughly), E tends to become overly conservative so E is replaced by the minimum of E and the data range Δ_M (Section 5.11.4). Next, the estimated discretization uncertainty U_{num} = GCI can be increased by the (sample) standard deviation of the least-squares fit U_s ,similar to the D_f term of Eq. (5.10.10.3.10). This term is essentially never the most important part of the estimated uncertainty; it could only happen if the scatter were very large[62], indicating something seriously wrong in the computation (Eça,

[62] Logan and Nitta (2006) perceptively noted that the additional uncertainty U_s arises conceptually because the LS-GCI involves a "model of a model," i.e. the least squares fit or *model* (with its uncertainty U_s) of a one-term power series *model* of the convergence. Their small set of problems with "intentional choice of grid studies with oscillations in both exponent p and output quantity f " led to such large U_s that it alone provided conservatism without the need for F_s > 1. This result will be misleading for more reasonable problems, for which U_s is negligible and F_s > 1 is necessary to attain 95% coverage. They also considered the variant of least squares minimization of U_s +|E| rather than E, which result does not behave correctly

(2009). Finally, if observed convergence is not monotonic, ε is replaced by the data range Δ_M and $F_s = 3$ is used. These refinements are summarized as **(A-C)** following (Eça and Hoekstra, 2009b).

(A) The apparent convergence condition is determined by Eqs. (5.11.1). Observed p is evaluated by LSQ fit of Eqs. (5.11.3). A correction (usually small) by (sample) standard deviation of the fit U_s can be evaluated.

(B) For apparent monotonic convergence, the discretization uncertainty U_{num} is estimated as follows, with the GCI evaluated with $F_s = 1.25$.

(B1) For $0.95 \le p < 2.05$, $U_{num} = \text{GCI} + U_s$

(B2) For $0 < p < 0.95$, $U_{num} = \min \{ [\text{GCI} + U_s], [1.25\ \Delta_M] \}$

(B3) For $p \ge 2.05$, $U_{num} = \max \{ [\text{GCI} + U_s], [1.25\ \Delta_M] \}$

(C) If the apparent convergence is not monotonic,

$$U_{num} = 3\ \Delta_M$$

As noted previously (Section 5.11.5) the use of switches means that an insignificant variation in observed p (over location or parameter variation) can cause an unreasonable jump in uncertainty estimate U_{num} for poorly behaved problems. The preferred approach is to do the work to achieve the asymptotic range.

5.12 § PRACTICAL ALTERNATIVE APPROACH TO CALCULATION UNCERTAINTY

Rather than targeting 95% certainty as in the GCI approach, Sinclair et al. (2006) adopted a practical alternative approach to the evaluation of accuracy and prediction for Computational Solid Mechanics problems, including detection and treatment of singularities (see Section 5.10.4.1). They aimed to achieve and predict (in the error estimation) one of the four accuracy levels:

- excellent, or < 1% error in stress prediction
- good, or <5%
- satisfactory, or <10%
- unsatisfactory, or ≥ 10%.

They evaluated the error estimate as an unambiguous success if the correct *level* is predicted, and acceptable if they missed by only one level, without worrying about being inside or outside target error bars, which is the GCI approach. This criteria is rational and seems appropriate, especially for the difficult problem of singularity detection, and probably would be easier to correlate with single grid estimators (Chapter 7). It would seem to be more in the spirit of expert opinion[63]. The GCI uncertainty approach has become well established, but the approach of Sinclair et al is a reasonable alternative and in step with engineering practice "in the trenches."

---

for more benign problems but provides further robustness, and another variant involving a "prediction interval correction extrapolation to $h = 0$." They also use the term "Response Surface Method" for LS-GCI. Although their methods are effective, the approach of Eça and Hoekstra is straightforward and more tested.

[63] As was the original GCI, where the criterion for acceptance was relating the actual ε to an equivalent ε for a second order method with grid doubling, deemed acceptable by expert opinion. See Section 5.6.

5.13 § INCORRECT ALTERNATIVE APPROACH TO UNCERTAINTY AND VALIDATION

In November 2004 a Workshop on V&V was organized by the U.S. National Institute of Standards and Technology (NIST) at their offices. The purpose was (Fong and de Wit, 2006) to "advance the research on a framework of methodologies for the verification and validation (V&V) of computer models of complex engineering systems *with or without experimental data*." (Emphasis added.) Of course, Validation "*with or without experimental data*" is an oxymoron, so credibility was lost in the first sentence. Also requested was input from V&V practitioners (31 of the 50 participants were from outside NIST) on the NIST approach, notably to "assess and improve the resulting metrology-based approach to V&V" referred to as MV&V (Fong et al, 2004). For comments on the Workshop itself, see Roache (2006).

The basic approach, and the basic flaw, of metrology-based V&V is to treat the results of simulations like one treats statistical variations in a production manufacturing run.[64] MV&V would replace the inconvenience of experiments with a kind of democratic approach called a "consensus mean" in which the results of various simulations are weighted and combined with statistics to arrive at the benchmark values. One might hope that at least a 4th order solution on 10^7 cells would be weighted more heavily than a 1st order solution on 10^2 cells, rather than a *pure* democracy of "one code, one vote." But the coarse grid, 1st order solution would be counted as part of the Validation replacement for experimental data. We in the V&V community must be clear and unequivocal:

$$\textit{No Experiments} \Rightarrow \textit{No Validation}$$

When it comes to complex engineering systems, I am not a purist. I would be open to the possibility of considering a claim to Validation and/or Verification of a system based on V&V of components and their interactions. There will be doubts about unanticipated coupling, exceeding parameter ranges, etc. Many have made a good case for the more demanding claim that only full systems experiments should deserve the claim of Validation. Likewise, I can defer to engineering judgment of experts in regard to interpolation and even extrapolation in the input parameter space, and determining the limits of the domain of Validation. And of course the criteria for acceptable level of Validation for applications may necessarily be compromised by one's inability to perform good experiments (e.g. limitations on weapons testing) or the impossibility of controlled experiments (e.g. true astrophysical experiments, climate modeling, etc.). Again, I have no problem with claiming Validation for weak agreements, provided that some quantitative correspondence exists (e.g. parameter trends) and that the coarse level of agreement has some engineering or scientific utility. But the referent must be physical measurements, not simply an ensemble of un-validated computations.

The suggested MV&V approach also covers Calculation Verification, in the same democratic fashion. This approach is not just questionable; I believe that it can demonstrated definitively to be wrong. If MV&V for Calculation Verification works at all, it should work for the best-behaved computational cases:

(a) correct codes (i.e. no coding errors),
(b) simple well-behaved linear problems (no singularities, no advection terms, e.g. a Poisson problem with a smooth source term),
(c) regular mesh generation (e.g. uniform quadrilaterals or triangles),
(d) rigorous iteration convergence criteria or use of direct solvers with insignificant round-off accumulation,
(e) high enough resolution on all meshes used to provide monotonic mesh convergence behavior (very easy to achieve for the example Poisson problem).

[64] This concept is applicable to a statistical analysis of a state of the art; see Section 10.24.

For these best-behaved cases, consider the MV&V approach of processing the results of N1 hypothetical codes on N2 meshes. The hypothetical N1 codes all use the same continuum equations, including boundary conditions, so there exists an unambiguous correct mathematical answer. The hypothetical codes cover a range of orders of formal accuracy; to be specific, we consider orders of convergence (theoretical and observed) p = 2,4,6,8. Consider meshes starting from coarse resolution (say 10 elements in each direction, giving accuracy of perhaps ~20%) but still fine enough (for this simple, well-behaved problem) to be within the asymptotic range so that mesh convergence is monotonic. At the high end, we consider a large number, say 10^4 elements in each direction.

The MV&V method will *always* give a mean solution that is worse than the best solution. (The best solution would be the highest order method on the finest mesh.) This must be true because MV&V weights the best with the worst solutions (lowest order method on the coarsest mesh). Furthermore, the variance of the solutions will always give error bars (i.e. mean $\pm$ δ) that are inside the extremes of the results; yet the true answer is in fact always outside the set of all results. This is because the convergence in these best-behaved cases is monotonic, i.e. the convergence is one-sided, so the correct answer is approached asymptotically, and it will never be obtained by *any* interpolation or averaging of individual computations. In fact, the best estimate is obtained by *extrapolation*, whereas the MV&V approach is always some kind of interpolation.

If we looked only at codes using 2nd order methods, and if the mesh resolution sampling were well covered, the MV&V estimate might loosely be expected to be close to the solution on a 10^3 mesh, whereas the true answer lies beyond that of the 10^4 mesh, and outside the error bars provided by the variance of the results.

If we looked at only one mesh, say the 10^2 mesh, the MV&V estimate might loosely be expected to be close to the solution obtained by the p = 2 or 4 method, whereas the true answer lies beyond the p = 8 solution. However, for this situation the variance might provide a conservative error band.

Not surprisingly, the example MV&V exercise in Fong et al (2004) showed no significant mesh convergence tests (except for case CPS4). The essential point to bear in mind is that all the (correct) codes will give the correct answer asymptotically as the mesh is refined. So why not do it? And why give any weight at all to the CPS4 solutions in 1×2×10 and 1×4×20 meshes, when we have a 1×4×80 solution that must be better than the first two? (The information from the first two solutions can of course be well used to extrapolate to a better estimate and provide error estimates, but this is not considered by the authors.) Also, since all the (correct) codes will give the correct answer asymptotically as the mesh is refined, there is no need to run the case "in as many FEA codes as possible."

In summary, the MV&V has nothing to do with Validation (experimental agreement) and demonstrably cannot correctly do the Verification. The basic choice of a "consensus mean" is incorrect. To continue the analogy used in Fong et al (2004) with experimental results from different laboratories, the MV&V approach would be like statistically analyzing 100 laboratory results for the speed of light, one of which was provided by the Stanford Physics Labs, one by CERN, and the other 98 by high school physics projects.

5.14 § BAYESIAN VS STRICT FREQUENTIST INTERPRETATIONS FOR GCI

The interpretations of the uncertainty given so far for the GCI and similar numerical uncertainty estimators are intuitive and straightforward. As we described in Section 2.3.2.2, the most common type of error bar or "expanded uncertainty" denoted by $U_{95\%}$ indicates that we want to put a $\pm$ error bar (or error band) around our calculated value that we expect to contain the true (mathematical) value in about 95% of cases, or with about ~20:1 odds. However, we noted that there are some underlying issues that are long-standing controversies in the statistical and engineering communities. Intuitively, and in normal

conversation, one could also speak of ~95% level of confidence, or ~95% probability. But these are loaded technical terms to statisticians, and the associated debates strike at the heart of science philosophy. To begin, the word "cases" needs some consideration.

5.14.1 § Representative Populations for Computational Uncertainty

[65]To begin speaking carefully (which can be tiresome) about statistics, one needs to define a population on which statistics are done. As in Coleman (2002), the estimated computational *or* experimental solution value S is related to the true (computational *or* physical) value T by the definition of Uncertainty $U_{95\%}$,

$$| S - T | \leq U_{95\%} \tag{1}$$

in 95% of the cases.[66]

The very concept of error bars or probabilistic uncertainty statements for a computational result is not without a controversial aspect, and therefore must be addressed herein, even at the risk of pedantry. The computation of a single problem is deterministic, not probabilistic; there is a single unambiguous correct answer achievable with adequate grid (space and time) resolution. While an error estimate (such as that obtained with Richardson Extrapolation) is unquestionably appropriate, it has been argued (Oberkampf et al, 2004) that a probabilistic statement comparable to that used for experimental results is not appropriate. "We believe that representing the uncertain estimate of [numerical] error as a probability ... is not defensible, since convergence of numerical errors is more closely analogous to bias errors in experimental measurements." Others (Coleman, 2002; Roache, 2003c) have argued that it is a reasonable approach to use uncertainty for computations. The key concept is that of a gambler's bet, and formulating a consistent problem ensemble (or population, or class, or collective).

We might consider the class of all CFD problems, or more generally all computational PDE problems, or less generally all transonic airfoil problems. The more specific the class, the easier it would be to make precise the error measure for the class and to avoid normalizing ambiguities, i.e. we could look at C_D or C_L for airfoils. But if some fuzziness in the metric is allowed, we could meaningfully speak of a % error for a broad class of problems. For specificity in the discussion, consider the following "Publication Ensemble": the class of all computational PDE problems (and CFD, MHD, laser, plasmadynamics, heat transfer, etc.) submitted to both the *AIAA Journal* and the *ASME Journal of Fluids Engineering* during the decade of the 1990's. This would define a population or collective of some hundreds of *physical* problems, and even more *computational* problems, since each solution presented (e.g. different methods, parameter sets, final grids) constitutes a separate entry ("case") for our statistical ensemble. Whether or not an error estimate has been made for each entry, it is true that each entry has with it an associated exact asymptotic answer.

We are considering only grid convergence, i.e. the exact answer for the continuum differential equations, without considering the modeling accuracy of either (a) of the governing partial differential equations (the Validation issue, e.g. we are not considering the adequacy of turbulence models) or (b) of the far-field boundary conditions. The latter is another Calculation Verification issue, which can be addressed by methods analogous to a grid convergence exercise, e.g. systematically testing the sensitivity of answers to the position of the outflow boundary and the boundary condition imposed there, and thereby estimating

[65] This subsection is taken from Roache (2003a).

[66] Recall that there are two "true" values, the computational true solution (an idealization approached as the limiting solution of the mathematical model) and the physical true solution (also an idealization, approached conceptually as the limit of the perfect experiment). Assessment of the disagreement between these two true values, clouded by the uncertainties of each, is the subject of Validation.

that error. (See Section 6.10.) A complete "numerical error bar" would include these other factors, such as the estimate of uncertainty from finite domain size, as suggested by Karniadakis (1995).

To say that each entry in our ensemble has associated with it a true answer does not of course say that we *know* the true answer, any more than it does in an experiment. In both the computational and experimental situations, the "true answer" is an abstraction that is still useful as the standard. In both cases, it is meaningful to estimate bias errors, but in the experimental case one also can run repeat experiments (with different set-up position errors, different atmospheric conditions, drift in instrumentation errors, etc.) for a single problem and produce estimates of the random error. As the number of experimental replications increases, the mean converges (by *definition* of bias error) towards the correct value (defined in relation to the abstract "true answer") except for bias error. This claim of convergence of the mean to the correct answer *except for bias error* might seem to import great power to repeat solutions, but this is illusory; it occurs simply because the difference between the converged mean and the true answer is *defined* as bias error. The estimation of bias error can be approached by repeated testing in various facilities, but the mean of these results does not approach the true answer, i.e. there almost certainly are experimental errors common to all facilities that are not random about the true answer, e.g. tunnel free-stream turbulence effects.

One could imagine a similar approach in the computational case, with a single physical problem being re-calculated with different grid sequences, different discretization methods, different iteration convergence criteria, etc. and thereby produce a distribution of results with random errors for a single problem. However, this concept is not of much practical importance, in my opinion. The more practical concept is the probabilistic assessment of the accuracy for the Publication Ensemble, stated as a "gambler's bet."

5.14.2 § Implied Uncertainty of an Error Estimate

Any one continuum mathematical problem has a correct answer, and for it we have one computational answer, with (hopefully) an associated signed error estimate E_e. As is well known, one cannot perform formal statistics with a sample size of 1. Yet if we consider an ensemble of problems, we can do statistics. We *might* be able to say something like the following: the true answers for all these problems lie within the given computational solutions ± the associated error estimates, in 50% of the cases. Then we could say that we had 50% "confidence" (on this use, see footnote #34 on page 122), or a 50% confidence interval, and the absolute values of the error estimates would constitute a 50% uncertainty.

$$|S - T| \quad \leq \quad |Ee| \quad \equiv \quad U_{50\%} \tag{5.14.2.1}$$

Then for any one sample problem from the ensemble, a gambler's bet would be even odds (1:1) or 50% probability that the true answer for the sample problem would lie within the given computational solution ± |the given error estimate|.

For an ordered estimator like E_1, considering the wide range of problems in the Publication Ensemble or similar population, there is no reason to expect any preference for signs of higher order derivatives in the expansion of the error estimate. The true values will show no preference for being either > or < the ordered error estimate. Nor will this be changed by considering problems in which higher derivatives are not defined. It is also a reasonable assumption for any kind of best estimate, e.g. database values for material properties are just as likely to be high or low (unless they are "conservative" estimates vs "best" estimates). So quite generally, if δ is a signed estimate (ordered, or "best" in some sense) we can reasonably take

$$|\delta| \equiv U_{50\%} \tag{5.14.2.2}$$

For related discussion see Section 2.3.2.2 and Roache (2003a).

In computational and experimental practice, errors estimates have (consciously on unconsciously) been utilized as uncertainties of $U_{50\%}$. It has been argued (Oberkampf et al, 2004) that the grid convergence error of computations is more akin to experimental bias errors than to experimental random errors, and therefore the concept of uncertainty in computations is not appropriate. Even if the first part of the statement is granted, we assert (V&V20) that "uncertainty' in the sense of a gambler's bet is also applicable to experimental bias errors, in the same way that here it is applicable to computational errors, i.e. in the sense of an ensemble. Professional associations of experimenters (e.g. the Supersonic Testing Association) make assessments (estimates) of bias errors of their facilities. The ensemble of all these facilities can be treated as a population, and an overall estimate can be made of the state-of-the-art including an error bar or uncertainty estimate. In fact, it is impossible to *use* experimental data in any project involving Risk Assessment and Management without such an uncertainty estimate. Note that bias error estimates are not sufficient; an uncertainty estimate is required. But if only an estimate is given, without an explicit statement of probability, the only rational approach for an outsider with no additional information is to treat the absolute value of the error estimate as a 50% uncertainty. For example, Hemsch (2002) summarized a benchmark workshop organized by AIAA and NASA to Validate lift and drag predictions of a wing-body combination. 35 CFD calculations were presented. Statistical analysis indicated that the standard deviation of the solutions (even after exclusion of outliers) was 21 drag counts ($\Delta C_D = 0.0001$) while wind tunnel data is believed to be accurate to 4. (The industry goal is 1, on the order of 0.5% of total drag of a modern transport aircraft.) These results were quoted without criticism in Oberkampf et al (2002); I believe the exercise implicitly recognizes the legitimacy of probabilistic uncertainty statements for computational PDE results, and indeed the necessity of these for comparisons with experiments.

There remains a residual fuzziness in the statistical treatment of computational results, in that standard deviations are defined with respect to the mean of the data, whereas our error bars refer to deviations from the exact ($\Delta x, \Delta t \rightarrow 0$) solutions, which are not the mean.

When computationalists report an error estimate, e.g. based on RE, they are at best reporting an estimate of $U_{50\%}$, not $U_{95\%}$. The standard tolerance for experimental work is not 50% error bars, but 95%. That is, experimenters will present (graphically or algebraically) an error bar that includes 95% of the data[67]. Although experimental practice is such that a reliable $U_{95\%}$ usually is not met, it is at least honored in the breach; it is the stated goal.

Validation involves calculation of discrepancies between experimental data including their uncertainty with computational data including their uncertainty. It does not make sense to base the Validation metric on $|U_{EXP} + U_{COMP}|$ when U_{EXP} is a $U_{95\%}$ but U_{COMP} is a $U_{50\%}$, i.e. merely an |error estimator|. The use of $U_{95\%}$ means that the odds (gambler's odds) of any particular sample drawn from the data set being within the [mean $\pm$ error bar] are ~20:1. If the scatter is Gaussian, this closely corresponds to the 2-σ band (~95.44%). Note, however, that the definition of the error bar does *not* depend on the scatter being Gaussian or symmetric or even unimodal, and is unambiguously defined once some trivial details are addressed.[68] Occasionally, other tolerances are used, often corresponding to Gaussian 1-σ (~68%) or 3-σ (~99.7%). The customary 95% experimental goal has been specifically acknowledged and adopted by

[67] The reliability of the experimental error bar for random error component increases as the number of replications increases, a quality that ideally would be included in the definition of a metric for Validation (Oberkampf and Trucano, 2002; Oberkampf et al, 2004).

[68] For example, if there are only 10 data points, the error bars cannot contain 95% of the data, but either 90% or 100%. Likewise, if some of the data entries are repeat values, the error bars may be forced to include more or less than 95%. The individual statement of the error bar is readily modified to clear this up, if necessary.

several CFD practitioners (see refs. 1,4,15-29 of Roache, 2003a), although the actual practice has often been inconsistent. Note that by focusing on the experimental data set, rather than the abstract "true (physical) solution", the experimenter can meaningfully speak of a true error bar for the data, i.e. a quantity $U_{95\%}$ such that [mean $\pm$ $U_{95\%}$] includes 95% of the data. One could also conceive of theoretical models of the components of scatter (e.g., one could model position errors) and convolve these to develop a theoretical model of the true error bar, i.e. an estimate of the true error bar.

Likewise, returning to the Publication Ensemble, we may speak meaningfully of an abstract true error bar for this ensemble. That is, in principle we could re-evaluate all these problems with highly accurate numerical solutions, and establish a database of true computational solutions and establish $U_{95\%}$. Consideration would be given to normalizing across the entire data set, perhaps evaluating all metrics as % errors. If the dimensional local values are of interest, these could be normalized with a representative value for each problem, not by dimensional local values, to avoid near 0/0 indeterminacies. Alternately, if necessary for this mental exercise, we could restrict our ensemble to (say) drag calculation on airfoils. As in the experimental situation, we can now speak of a *true error bar* attainable in principle for the Publication Ensemble, and as well speak of methods for obtaining *estimates* of the true error bar.

5.14.3 § Bayesian vs. Strict Frequentist Statistics

All these issues will be familiar to any reader acquainted with the long-standing cultural divide between the Bayesian and (more traditional) strict frequentist schools of statistics. For applications of the GCI and other uncertainty estimators, common sense and engineering intuition are all that is needed, and this Section is not required reading. For those interested in pursuing the subject, three references are recommended: Kasser (2006), Wagenmakers et al (2009), and especially O'Hagan (2004). The subject is huge, the arguments are subtle, and emotions run strong. The consequences are stark, e.g. validity of interpretations of clinical tests of anti-cancer drugs. The publications bounce between declarations of boring and obvious statements to those of virtually impenetrable subtlety.

Statistics began with the frequentist approach (von Mises, 1957, first German edition 1928) in which "a quantitative probability concept must be defined in terms of potentially unlimited sequences of observations or experiments. The relative frequency of the repetition is the 'measure' of probability..." Von Mises termed these sequences "collectives," now more often called populations. This approach also limits the statistics to application not to another data set of similar type as the original population, but only to new populations derived from the original by four types of allowable "mixing" operations; the full limitations of this strict interpretation is seldom acknowledged. Von Mises stated "It is brought out repeatedly in this book that the word 'probability' has a meaning in everyday language that is different from its quantitative meaning in probability calculus." Thus we are again confronted with "mere semantics."

Opponents of the strict frequentist interpretation are not opposed categorically to re-defining everyday language terms to a more specific meanings within a technical context. The problem is that practicing engineers and scientists need *both* meanings, and the strict frequentist approach does not allow both. Here is an example taken from Wagenmakers et al (2009).

Consider a frequentist confidence interval for the normal mean μ of a given population, μ: $\mu \in \{-0.5, 1.0\}$. For a strict frequentist, the only correct interpretation is the following. When the frequentist procedure is applied many times to all kinds of possible data sets, the different intervals cover the true value of μ in 95% of the cases. This is counter-intuitive, and more importantly, irrelevant to researchers who want to learn about μ for their data. "Consistent with intuition, and consistent with what researchers want to know, this Bayesian interval conveys that there is a .95 probability that μ lies in $\{-0.5, 1.0\}$."

(Wagenmakers et al, 2009.) This intuitive and useful interpretation is a *mis*interpretation to a strict frequentist.

If the reader has accepted the interpretation of the GCI data sets given earlier, then he has intuitively adopted a Bayesian interpretation. The Bayesian approach to statistical inference involves modeling *all* unknown quantities with probability distributions. This is exactly what engineers and indeed scientists have been doing for a long time. Note, however, that the arguments given above in Section 5.14.2 based on the gambler's bet concept may have appeared naive and merely intuitive, but in fact these type of arguments are central to the most compelling arguments in Bayesian philosophy.

Why is the strict frequentist approach dominant, even though its usefulness is so limited? First of all, it is easy to teach a quantitative concept of probability in terms of relative frequencies occurring in indefinite sequences using intuitively appealing and familiar thought experiments: coin flips, dice rolls, poker hands, etc. Second, in spite of protestations by strict frequentists, people do use Bayesian inference, whether they acknowledge it or not. "Perhaps frequentist inference has survived so long because researchers translate the frequentist statistical outcomes to informal Bayesian conclusions." (Wagenmakers et al, 2009) See also Kacker and Jones (2003), who pointed out that the internationally accepted and widely referenced ISO Guide (ISO, 1995; see also Chapter 11) recommends classical (frequentist) statistics for evaluating uncertainties, but it interprets the combined uncertainty from a Bayesian viewpoint.

This should not be surprising; it is the way we all assess risks and "probabilities" in our lives. This is illustrated pointedly in the personal anecdote of O'Hagan (2004) in a widely referenced, highly readable two-page article. (The author has served on the Council of the Royal Statistical Society.)

"It was my experience, as a young statistician, of analyzing data and producing frequentist tests and confidence intervals for other scientists[69] that convinced me that the Bayesian approach is the right one for statistical analysis. I had great difficulty persuading the scientists not to misinterpret [see example above] the frequentist inferences I was giving them. And it was clear to me that this was because the correct interpretation was of no use to them. Frequentist inferences make only indirect statements about parameters, and can only be interpreted in terms of repeated sampling. Bayesian inferences directly answered the scientists' questions, making statements that were unambiguously about the parameters they wanted to learn about. Since that time (more than 30 years ago now) I have been an enthusiastic advocate and practitioner of the Bayesian approach."

The concept of relative frequency in an indefinite sequence of experiments is very valuable but it is also pertinent to Bayesian inference. It is sometimes stated that frequentists will apply the concept of probability only to objective and quantitative data, whereas Bayesians are subjective, a very dirty word. But in fact Bayesian philosophy is inclusive; it incorporates both objective and subjective probabilities, and if all the data are objective, as we would maintain for the GCI data, then that part of the analysis is objective. In developing the statistical base for the GCI (including the determination of the value for F_s) I have taken some pains to define populations. In this sense, I have tried to align with the strict frequentist philosophy as much as possible. I have not agglomerated all the studies into one data base, but each of the sampled data bases gives consistent results (bearing in mind the modest goal of *roughly* 95% coverage). But at the end, a strict frequentist would say that the GCI just covers that conglomerate data base. What we need is a statement that *probably* it applies to other cases as well. Our impression is that ultimately a Bayesian approach is required if we are ever to *use* any of such studies or analyses.

Traditional engineering classification on experimental errors (and corresponding uncertainties) distinguishes random and systematic errors. More recent terminology uses aleatory and epistemic

[69] Nuclear power engineers and scientists at the U.K. Central Electricity Generating Board (O'Hagan, 2008).

uncertainty, with aleatory (from the Latin *alea* for dice) referring to uncertainty that is at least in principle reducible (through repeated experiments) and epistemic (same root as epistemology, i.e. the study of knowledge) referring to uncertainty due to lack of knowledge. The interpretation and *application* of the distinction between random and systematic, aleatory and epistemic uncertainties, is not so neat as usually presented; e.g. see NIST TN 1297 (Taylor and Kuyatt, 1994). The strict frequentist approach is only applicable to random errors or aleatory uncertainty. A good example is wind tunnel testing. Repeated testing of the same models in the same wind tunnel will give a distribution of results, the spread of which indicates random error or aleatory uncertainty. (If done thoroughly, as in the classic study of Aeschliman and Oberkampf (1997a,b) repeats are done on different days, with removal and reinstallation of models.) Such thorough and objective data obtained from a long sequence of experiments are amenable to frequentist analysis and to frequentist *or* Bayesian inference. But the differences from one wind tunnel facility to another, and our ultimate interest in the difference between these and reality (i.e. a free flight case) cannot, practically speaking, be considered as part of a long sequence. In fact, some experimental facilities are unique. Yet the errors must be estimated and assigned probabilities or uncertainties, based on engineering analysis, experience and intuition that develops from experience - hence the "subjective" part of Bayesian inference. Once these probabilities are assigned, they are treated identically, according to internationally accepted standards like ASME PTC 19.1 (ASME 1986); see other references and discussion in Chapter 11. Intuition is not a mere new-age philosophy; it is developed from experience. (Intuition changes during education.) It is part of engineering and science, and should be incorporated into statistical analysis; statistics should not be done in an intellectual vacuum.

If the reader decides to pursue this subject, it is good to note that many arguments in the literature are beyond the level of ordinary engineering practice associated with V&V of computational PDEs, including such topics as null hypothesis testing, updating analysis with later data including the importance of prior probability assignments in Bayesian methods and some highly suspicious aspects for strict frequentist approach, discredited claims of objectivity (vs disguised subjectivity) for the frequentist approach, and the different flavors of the frequentist approach (Fisher vs Neyman and Pearson) that have become so unconsciously amalgamated that "the confusion between the two different approaches is now close to total" and "has rendered applications of classical statistical testing all but meaningless among applied researchers." (Wagenmakers et al, 2009). Other piquant observations follow.

"I also believe that everyone is born Bayesian, and only lose this state of grace by being exposed to frequentist ideas." (O'Hagan, 2008)

"We agree .... that, deep down inside, what researchers really want is to draw Bayesian conclusions. Or, in the words of Dennis Lindley, 'Inside every Non-Bayesian, there is a Bayesian struggling to get out.' " (Wagenmakers et al, 2009)

5.14.4 § High Consequence Applications

Experience has shown that the GCI with the Factor of Safety F_s determined from a fairly extensive set of studies involving many hundreds of cases produces a dependable 95% uncertainty estimator, even for some poorly behaved problems, especially if the more demanding Least Squares GCI is used. If one is presented with a new computational PDE problem that is not radically different from those already tested, one can proceed with high confidence, similar to that accruing to the ordinary standards of confidence in experimental results and engineering analyses that have led to modern engineering marvels. But if one is confronted with a very high consequence analysis, e.g. nuclear reactor performance and safety, and with unusually complex physical problems (e.g. multiscale turbulence in highly variable property fluids) it would be imprudent to proceed without further support for F_s. This is straight-forward to achieve, with

sampled calculations at high resolution performed to assess the adequacy of the F_s, but the computational workload is high. Fortunately, such high consequence programs will naturally involve a large body of multi-parameter simulations, making it easier to amortize the cost of this determination. (See Section 6.33.)

5.15 § EVALUATION OF UNCERTAINTY ESTIMATORS FROM SMALL SAMPLE STUDIES[70]

The published literature contains methods for uncertainty estimation justified on one or two cases. This is notably the case for single-grid methods (Chapter 7) but is also true for many (indeed, most) multiple-grid methods. (E.g. see Sections 6.25.2,3.) Several of the presentations and discussions in the Proceedings of the Third Lisbon V&V Workshop (Eça and Hoekstra, 2008) described success, or at least promise of success, for new or modified $U_{95\%}$ estimators, based on a few test cases. What could this mean?

If we agree that the target is $U_{95\%}$, i.e. an estimator that includes the actual error in roughly 95% of the cases, then we would need to examine at least 20 cases to expect to find one for which the actual error is outside the $U_{95\%}$ range. (Note this is not truly a "failure" because we do not want to pay the price of huge uncertainty estimates for ~100% coverage, including outliers.) Twenty cases would not be sufficiently large for dependable statistics; we would expect to need $O(100)$ cases to have confidence in the claim of roughly 95% coverage. "Cases" do not necessarily mean separate fluid dynamics problems, since the term includes multiple grid triplets for the same problem, but for a convincing evaluation of a method for general use we certainly need many fluid dynamics problems. To date, I think it is fair to say that only the only uncertainty estimation methods that have been subjected to a statistically significant database are the GCI and Least-Squares GCI, and the related ITTC Correction Factor Method in its latest incarnation studied by Xing and Stern (2009); see Section 6.25.2).

In spite of this, I believe that the small sample studies are valuable. The analysts are essentially using their intuition, based on extensive computational experience, to cautiously infer something about the $U_{95\%}$ statistical coverage from the performance on a few cases. Suppose that we have just one case, and it is in some sense representative (i.e. not a singular problem, not a terrible grid, there are some points in a boundary layer and along a backstep, observed p is reasonable, etc.). If the $U_{95\%}$ estimate is 20% while the actual error is 0.2%, we would at least hypothesize, perhaps reasonably conclude, that the uncertainty estimator is far too conservative (by ~ two orders of magnitude) even though we all know that we cannot do any statistics with a sample size of one. Conceivably, we might do 100 total cases and find that it is non-conservative in the other 99 cases. But we are not doing blind statistics uninformed by experience, and if such a study showed 99/100 cases were non-conservative and 1/100 showed the estimator was far too conservative, we would surely re-examine that special case, expecting to find that a mistake was made. Similarly, if we examined 5 cases and two were non-conservative but only by a little (say, $U_{95\%}$ estimator = 10% and actual error = 11%) we would be encouraged, even though if this pattern later proved to be replicated for 1000 cases, we would have verified not a $U_{95\%}$ estimator but an inadequate $U_{60\%}$ estimator.

Although these small sample studies, combined with well-founded intuition, are valuable, the convincing statistical evidence can come only with hundreds of cases, on some range of fluid dynamics problems. (Alternately, one could perform the evaluation only for a limited range of problems of interest, e.g. drag of ship hulls, claim only this success, and leave it to someone else to evaluate the $U_{95\%}$ estimator on another problem like airfoil lift or structures, if that is their interest.)

With such statistically significant evaluations ultimately in mind, it seems ill-advised to be introducing too many new parameters into the $U_{95\%}$ estimators and fine tuning them on small sample studies. Dr. Jay Boris noted ironically in the 1980's that there were so many variations possible on discretization algorithms

[70] From Roache (2003a, 2008b).

that nobody ever needed to use the same algorithm twice. The same is obviously true of RANS turbulence models, with all their switches and knobs, tunable for each new problem. Researchers are creative, and we are now in danger of seeing the same situation for uncertainty estimators. However, Eça and Hoekstra (2008b) have pointed out, it is also true that excessively conservative uncertainty estimates that "jump" by a factor 3/1.25 = 2.4 due to noise around observed $p \sim 0.95$ or 2.05 are disconcerting and would probably not be used. For me, this is an unsatisfying state of our present methodology, but at least it only occurs for the most difficult problems.

5.16 § ON NOT DISCARDING OUTLIERS

The discarding of outliers of statistical distributions is an accepted part of science and engineering education. The discarding can be done informally (intuitively) or formally, based on well-established (19th century) mathematics and philosophy using Chauvenet's criterion (widely used) or Peirce's criterion (better founded but more cumbersome), both of which are objective and repeatable (Ross, 2003). The latter was used by Xing and Stern (2009) to evaluate a significant data set for their uncertainty method (see Section 6.25.2).

It is my opinion that this process, whether done informally or formally, is invalid for numerical uncertainty studies unless further justification is given. As is often the case, convenient and well-established mental habits are applied reflexively without re-consideration of the underlying conceptual assumptions. We may ask, What is the conceptual basis for discarding a measurement just because it is significantly different from others in a data set (no matter that the discarding may be done "objectively" by a repeatable algorithm)? The textbook case is typified in Ross (2003) as follows. Ten pressure measurements (kPa) were recorded *at one setting* in an experiment, yielding values of:

> 101.2, **90.0**, 99.0, 102.0, 103.0, 100.2, **89.0**, 98.1, 101.5, 102.0
> (mean 98.6, sample standard deviation 5.02).

"The pressures of **89.0** and **90.0** appear suspect." These both could be discarded intuitively, or analyzed formally resulting in discard of only the 90.0 value by Peirce's criterion and none by Chauvenet's criterion (Ross, 2003). The *underlying conceptual assumption* is that there exists one correct, "true" value, which has been corrupted (perhaps by human error). The data were all taken for the same physical problem and were recorded *at one setting* in an experiment, so it is reasonable to be suspicious of any outliers.

This is not analogous to the situation of evaluating numerical uncertainty estimators for a variety of cases. We cannot assume that a better-controlled "measurement" would have produced less noisy data. There is not a single true answer that we can reasonably expect to apply to all cases. We *wish* it to be true, we *hypothesize* it to be true, and so we test it to see how often it fails (target ~5% failure rate). But it is not reasonable to assume (in the strongest sense) that failure of some uncertainty estimator (say GCI) for a small number of difficult problems (say multiphysics with mixed type equations and step changes in parameters) can be discarded because GCI gives good estimates for a large number of benign problems (say strongly elliptic equations). Likewise for studies of the same physical problem wherein outlier "cases" might cover completely inadequate grids. A legitimate analog to the experimental measurements would be a situation of ten analysts using the same computational model on the same physical problem with the same parameters and comparable gird resolutions and discretization algorithms, with noisy variations coming from (hopefully higher order) effects such as minor details of coding, iteration convergence criteria, solvers, grid details. In this hypothetical situation, an outlier would be legitimately suspect and a candidate for discarding. But in the normal case, outliers for uncertainty evaluations can be viewed suspiciously (e.g. misapplication of GCI formulas, which has occurred) but they should not be discarded automatically.

5.17 § INCREMENTAL COSTS OF GRID CONVERGENCE STUDIES: THE BLESSING OF DIMENSIONALITY

It is often heard that one cannot afford to do error estimation by grid convergence studies because of the high cost of multidimensional calculations. In fact, the incremental cost of grid convergence studies, when properly normalized, is *less* in higher dimensions. The "curse of dimensionality" is a well-known phrase and phenomenon of computational PDEs. In a 1-D problem, if we double the resolution, we double the number of grid points N, and we expect that the operation count and computer time will double, at least. In fact, for most iterative methods, the number of iterations will also increase with N, so the operation count $\propto N^a$ where $a > 1$, often $a \sim 2$. Optimal methods like true multigrid give $a \sim 1$.[71] In a 2-D problem, when we double the resolution N in each direction, the operation count increases at least by a factor of 4. In time-accurate problems, generally the time resolution must be refined similarly to maintain consistent accuracy, so doubling resolution in a time-accurate problem with 3 spatial dimension (computational D = 4) increases the operation count by at least a factor of $2^4 = 16$. From another perspective, an factor of 2 improvement in computer speed enables only less than a 20% increase in resolution.

However, for grid convergence studies, we *coarsen* rather than refine, and this curse becomes a blessing. Consider a computationally 4-dimensional problem (D = 4) and a grid refinement (obverse, coarsening) factor $r \sim 1.3$.[72] Consider a base grid of 200 cells (or time steps) in each computational dimension, and coarsen by $\sim r = 1.3$ to produce the grid sequence n = 1,2,3,4 with N(n) = 200, 154, 118, 90. With the cost of computing a solution on the base grid $n = 1$ denoted as *Cost(1)*, we have the *Cost(n)* given by

$$Cost(n) = Cost(1) \Big/ r^{D(n-1)} = Cost(1) \Big/ 1.3^{4(n-1)} \tag{5.17.1}$$

Normalizing the cost of computing a solution on the base grid $n = 1$ to *Cost(1)* = 1, the *Cost* for each grid and the total cost Σ for the 4 grid sequence is

$$\begin{aligned} &Cost(n) = 1, \quad 0.350, \quad 0.123, \quad 0.043 \\ &\Sigma = 1.516 \end{aligned} \tag{5.17.1}$$

or 52% incremental cost of obtaining solutions for a 4 grid convergence study compared to the base grid.

If one considers (correctly) that 2 grids are a minimum requirement, with a cost of 1.350, then the additional incremental (and re-normalized) costs for the third and fourth coarsened grids are (0.123 + 0.043)/1.350 = 0.123, or a 12% incremental cost of obtaining solutions for a 4 grid convergence study compared to a 2 grid convergence study. This additional 12% cost allows one to calculate observed convergence rate *p* over as many as 4 grid triplets[73] and/or use the Least Squares GCI. In practical situations, these penalties will be further reduced when amortized over time for problem definition and grid generation, and when combined with economies of sampling of input parameters (Section 6.33).

---

[71] For operation counts, see e.g. Chapter 10 of Roache (1998b) and Roache (1995b).

[72] For this example problem, we do not consider halving the resolution (which would make the argument overwhelming) since this may put the coarser grids of a sequence out of the asymptotic range. The value $r = 1.3$ is the recommended minimum in the *ASME Journal of Fluids Engineering* procedure using the GCI; see Celik et al (2008). If this minimum is accepted (but see comment at end of Section 5.10.1), then the economies calculated are worst-case.

[73] Grid triplets n = 123, 124, 134, 234.

5.18 CONCLUSION

It is recommended that results of systematic grid convergence studies be uniformly reported using the Grid Convergence Index of Eq. (5.6.10) or (5.7.2). The GCI provides uncertainty estimates or error bars based upon applying an empirically determined Factor of Safety F_s to a grid convergence error estimator derived from the theory of generalized Richardson Extrapolation. While not answering all questions involved with Verification of a calculation, this method at least enforces some uniformity in the reporting and is based upon an objective asymptotic estimate of the grid convergence error. Use of $F_s = 1.25$ should be restricted to convergence studies with a minimum of three grids that experimentally demonstrate an observed order of convergence p in reasonable agreement with theoretical p on the actual problem. For problems with noisy observed p, the Least Squares GCI of Section 5.11 is recommended.

Since the GCI will often be less optimistic than the simplistic ε of Eq. (5.4.3), especially for the all-too-popular 1st-order methods, some reluctance of authors may be anticipated. Fortunately, the formulas are simple enough to be applied *a posteriori* by editors and reviewers. It is urged that they do so in the review process to continue improving the quality of computational PDE papers. To quote Ferziger (1993), "*... the frequently heard argument 'any solution is better than none' can be dangerous in the extreme. The greatest disaster one can encounter in computation is not instability or lack of convergence but results that are simultaneously good enough to be believable but bad enough to cause trouble.*"

CHAPTER 6

APPLICATIONS OF SYSTEMATIC GRID CONVERGENCE STUDIES AND THE GRID CONVERGENCE INDEX (GCI)

6.1 INTRODUCTION

This chapter presents applications of systematic grid convergence studies and the Grid Convergence Index (GCI) in both Verification of Codes and Verification of Calculations. The examples are taken from various articles in the open literature to exemplify good work and to illustrate the details of sensitivity for particular calculations. Such Verification is performed external to the code and therefore is applicable to commercial codes, with or without access to the source code (which would be dangerous). It is the easiest to apply of all error estimation or Verification methods (it just requires methodical plodding) and is the most reliable and flexible.

6.2 TWO FURTHER EXAMPLES OF (PARTIAL) CODE VERIFICATION IN GROUNDWATER FLOW

Chapter 3 presented a detailed example of systematic Verification of code (produced by Symbolic Manipulation) using the Method of Manufactured Solutions to obtain the non-trivial analytical benchmark solution. This methodology is very general. Whether the benchmark solution is manufactured or not, or realistic or not, the Code Verification involves the same procedures. So long as the benchmark solution exercises all the aspects of the discretization, the Code Verification is convincing.

Following are two further examples of (partial) Code Verification via systematic grid convergence studies, both taken from Roache et al (1990). In this paper, we described some our of experiences with formulating and applying several benchmark test cases in groundwater hydrology computational PDE problems. The problems were all single phase Darcy law flows. Three problem categories were considered:

1. full 2-D groundwater hydrology flow codes tested on a steady problem with constant conductivity in stretched cartesian coordinates;
2. Fortran subroutines produced by computer Symbolic Manipulation for the stencil array evaluation of tensor conductivity in general non-orthogonal 2-D and 3-D coordinates;
3. particle tracking in 2-D and 3-D.

Freedom from coding errors, consistency of the discretization, and order of convergence were Verified. The first two exercises, described below, demonstrate successful Code Verification and successful error detection. The third example of an unsuccessful "false negative" test will be presented in Chapter 8.

6.2.1 Darcy Flow in Stretched Orthogonal Coordinates

Four codes were considered in this part of the study: SWIFT II (Reeves, et al, 1986), HST3D (Kipp, 1987), MODFLOW (McDonald and Harbaugh, 1988), and SECO_FLOW (Roache et al, 1995). SWIFT II is a code developed by Intera Technologies for Sandia Laboratories. HST3D and MODFLOW were developed by the USGS (United States Geological Survey). SECO_FLOW (for **S**andia-**ECO**dynamics) is a suite of codes (including 2-D and 3-D SECO_TRACKER and SECO_TRANSPORT) developed at Ecodynamics for Sandia Laboratories. SWIFT II and HST3D use primitive variables and allow for varying fluid compositions (e.g. brine, heat) whereas MODFLOW and the early version of SECO_FLOW allowed for only a pure component. (A version developed later included variable density.) SWIFT II and MODFLOW use a Marker-And-Cell (MAC, or Arakawa "C") staggered grid (e.g., see Roache, 1998b), HST3D uses a non-staggered grid (i.e., collocated variables) and SECO_FLOW has options for either. All four use fully implicit (backward Euler) time differencing, and all four appear from the problem formulations and code descriptions to be uniformly 2nd-order accurate in space and 1st-order accurate in time.

The simplest formulation of the governing equations for this problem gives a single parabolic equation in terms of hydraulic (piezometric) head *h*.

$$S_s \frac{\partial h}{\partial t} = \nabla \cdot (k \nabla h) \tag{6.2.1.1}$$

The codes considered all use stretched cartesian (i.e. planar orthogonal) coordinates, so the conductivity *k* may be tensor, provided that the principal axes of *k* are aligned with *x* and *y*; this is effectively a scalar conductivity assumption, since it is equivalent to a simple rescaling of *x* or *y*. All four use, or have options to use, harmonic averaging for the conductance (a combination of the physical variable conductivity and discretization terms), but for this first set of tests, only the constant *k* was verified. (Variable *k* tests, including full tensor *k* with discontinuities, will be described later.)

SWIFT II and HST3D treat much more complicated problems, so that many of their calculations are null for the simple problem described here. However, MODFLOW and SECO_FLOW are also fairly complex codes, owing not to the complexity of the governing PDE itself, but to the options for harmonic averaging and many *modeling* issues such as definition of aquifer properties, locally confined (artesian) or unconfined aquifer conditions, simulation of rivers, recharge, well schedules, inactive regions, etc. (All the

SECO codes have the feature of using separate grids to define aquifer properties and to perform calculations; this is highly advantageous for performing grid convergence studies.) All of these options introduce significant nonlinearities (step function dependencies on the dependent variable *h*) and coding complexities. Also, SECO_FLOW has both regional and local area grid solution capability (i.e. an elementary domain decomposition approach) and can treat discontinuous boundary definition of the general Robin type, whereas HST3D allows Dirichlet or non-homogeneous Neumann, and the other codes all use only homogeneous Neumann boundary conditions.

A benchmark (partial Verification) test case for steady flow was formulated as follows. With constant aquifer properties on a square domain of (0, *a*) × (0, *a*) and no wells, the boundary conditions are as follows. (The variable "*h*" = head for MODFLOW and SECO_FLOW, or pressure for SWIFT II and HST3D.) At $y = a$, the Dirichlet condition applies as

$$h(x,a) = c\left(\frac{\pi x}{a} + 1\right) \tag{6.2.1.2}$$

$$c = \cosh(\pi) = 11.59... \tag{6.2.1.3}$$

and on the other three sides of the domain, the homogeneous Neumann condition applies as

$$\frac{\partial h}{\partial n} = 0. \tag{6.2.1.4}$$

The analytic solution is

$$h(x,y) = \cos\left(\frac{\pi x}{a}\right)\cosh\left(\frac{\pi y}{a}\right) + c \tag{6.2.1.5}$$

The domain chosen was 40 km × 40 km, but these values, like the constant aquifer properties, scale out of the results for the test problem. The analytic solution was used to set initial conditions at all points.

SECO_FLOW uses this discretized analytic solution to set discrete boundary conditions, which means boundary values and values at the first interior cells. Thus, at boundaries with a homogeneous Neumann condition, the two-point difference equation for $\partial h / \partial n$ is not set to zero, but to the discrete values from the exact solution, which only approach zero as the grid is refined.

Two problem sets are presented. Table 6.2.1.1 presents results for a uniform staggered (MAC) grid, and Table 6.2.1.2 for a stretched MAC grid. The Tables show maximum error ERR_MAX and COEFF_MAX = ERR_MAX · (IL – 1 + mac)^2, where mac = 1 for the MAC grid, or mac = 0 for the collocated grid option. For a uniformly 2nd-order accurate solution, the value of COEFF_MAX should become roughly a constant as the grid is refined.

IL × JL	ERR_MAX	COEFF_MAX
5 × 5	1.129E–03	2.82E–02
10 × 10	4.142E–04	4.14E–02
20 × 20	1.390E–04	4.14E–02
40 × 40	2.274E–05	3.64E–02
80 × 80	4.902E–06	3.14E–02

Table 6.2.1.1. Convergence of SECO_FLOW, Uniform Grid. (From Table 1 of Roache et al, 1990.)

IL × JL	ERR_MAX	COEFF_MAX
5 × 5	5.906E–04	1.48E–02
10 × 10	5.354E–04	5.35E–02
20 × 20	4.242E–04	1.70E–01
40 × 40	1.704E–04	2.73E–01
80 × 80	4.101E–05	2.62E–01

Table 6.2.1.2. Convergence of SECO_FLOW, Stretched Grid. (From Table 2 of Roache et al, 1990.)

The results in Table 6.2.1.1 display the expected 2nd-order accuracy (somewhat better). The maximum error occurs well away from boundaries, e.g., in the 40 × 40 grid, at (I, J) = (7, 34).

Table 6.2.1.2 presents the same type results, but for a stretched grid. The grid stretching used in x concentrated (arbitrarily) the stretching at x = 20 km in the 40 km × 40 km grid, with a power-law stretching with b = 0.5. This gives a ratio of maximum to minimum x-spacing of 1.72 in the 5 × 5-cell grid, and 6.33 in the 80 × 80-cell grid. The grid stretching used in y concentrated the stretching at y = 23 km with b = 0.4. This gives a ratio of maximum to minimum y-spacing of 2.97 in the 5 × 5-cell grid, and 5.92 in the 80 × 80-cell grid.

Again, the results display the expected 2nd-order accuracy, and the maximum error occurs well away from boundaries, e.g., in the 40 × 40 grid, at (I, J) = (35, 34). Asymptotic behavior is reached more slowly, and the maximum truncation error is larger, for the stretched grid than for the uniform grid, as is to be expected for a smoothly varying solution. This is the cost for the increased resolution near the grid concentration point (20 km, 23 km).

A second similar test problem, which we had previously used on SWIFT II tests, was also exercised on SECO_FLOW. At $y = a$, the Dirichlet condition applies as

$$h(x,a) = c \cdot \cos\left(\frac{0.5\pi x}{a}\right) \tag{6.2.1.6}$$

$$c = \cosh\left(\frac{\pi}{2}\right) = 2.509... \tag{6.2.1.7}$$

At $x = a$, the Dirichlet condition applies as

$$h(a, y) = 0 \tag{6.2.1.8}$$

and on the other two sides of the domain, the homogeneous Neumann condition applies as

$$\frac{\partial h}{\partial n} = 0. \tag{6.2.1.9}$$

The analytic solution for this second problem is

$$h(x, y) = \cos\left(\frac{\pi x}{2a}\right)\cosh\left(\frac{\pi y}{2a}\right) \tag{6.2.1.10}$$

This problem was considered for SECO_FLOW only because it caused difficulties for SWIFT II. Although the solution behavior is not significantly more difficult than the previous problem, it does contain Dirichlet boundary conditions on two adjacent boundaries. This causes a local error in SWIFT II near that corner. No such difficulty arises with SECO_FLOW, as expected. In the 80 × 80 uniform grid, ERR_MAX = 6.265E−06 and COEFF_MAX = 4.01E−02, which are comparable to the errors for the first problem. Also, the maximum error occurs at (I, J) = (22, 41) indicating no difficulty near the corner, in contrast to the SWIFT II results, which were clearly indicative of a larger error. The problem was not unknown to the code authors; the SWIFT II user manual indicated correctly that Dirichlet conditions at a corner cell, where different Dirichlet values apply at each side, will be over-written so that only the second entered value is active. Thus, the two adjacent faces of a corner cell cannot have different Dirichlet values. That is, due to coding peculiarities (attributable to its age), SWIFT II needs to have

$$h\left(\frac{a - dx}{2}, a\right) = h\left(a, \frac{a - dy}{2}\right) \tag{6.2.1.11}$$

What we could not explain was that, even with the first problem (involving 3 Neumann and 1 Dirichlet boundaries), neither SWIFT II nor MODFLOW displayed 2nd-order convergence rates in the resolution ranges tested. Both of these use MAC grids. HST3D, which uses a collocated variable grid, did display 2nd-order convergence rates (as did SECO_FLOW above, also using MAC grids). However, for a related problem with specified non-zero flux boundary conditions, HST3D proved to have an error by a factor of 2 in the magnitude of the computed flux boundary value, only for a steady state calculation. The pressure solution itself was correct, but the post-processing evaluation of boundary flux was in error. It turned out that this coding error could easily be averted by changing the time differencing scheme selected for the steady state calculation. (The coding error was easily corrected by the author.)

6.2.2 Darcy Flow with Tensor Conductivity in Non-Orthogonal Coordinates

The tests described above Verified SECO_FLOW and HST3D for the simple (and unrealistic, for groundwater modeling) cases of constant properties. The next tests involved a much more general problem, and exemplify the debugging that can be accomplished with this approach of systematic grid convergence testing.

Modules for the SECO_FLOW codes were developed that use 2-D and 3-D non-orthogonal coordinates and allow for tensor conductivity. The Fortran source codes for the stencil loading were

produced using computer Symbolic Manipulation, rather than hand coding.[74] These code subroutines were verified by solving several test problems, the most difficult of which involves an ellipse shaped region on the unit square with a discontinuous jump in tensor conductivities across the boundary.

A driver code was written to test the code and theory of automatic generation of Symmetric Finite Difference Stencils in Generalized 3-D Coordinates (Steinberg and Roache, 1990). The goal of the exercise was to verify 2nd-order accuracy of the finite difference approximations to the continuum problem, the latter consisting of any second-order symmetric elliptic operator with tensor coefficients applied to a continuous function on an arbitrary connected domain in the plane. A discontinuous coefficient tensor model was included to simulate the case of abrupt changes in the permeability of adjacent geological formations.

The basic capabilities of the driver code were, therefore, to generate

i. a set of transformations from logical space to physical space in the plane,
ii. a symmetric elliptic operator, and
iii. a set of boundary conditions based on the choice of solution function.

With these inputs, the stencil-loader code was called to generate the symmetric stencils. A simple point SOR solver then used the stencils to obtain the discrete solution. Finally, the discrete solution was compared to a known exact solution to obtain the global discretization error.

The irregular domain produces a general non-orthogonal grid, the most general problem for which the stencil-loader was designed. A number of other domains (e.g., rectangles, parallelograms, trapezoids) were also used during the debugging phase of this exercise, but were superfluous in the final Verification. Likewise, separate 2-D tests were run, but only the 3-D results are shown herein. The test problems were designed by Dr. P. Knupp.

The Transformation

Two transformations were used.

A. Unit Cube:

$$\begin{aligned} x(\chi,\xi,\varsigma) &= \chi \\ y(\chi,\xi,\varsigma) &= \xi \\ z(\chi,\xi,\varsigma) &= \zeta \end{aligned} \tag{6.2.2.1}$$

B. Irregular Domain:

$$\begin{aligned} x(\chi,\xi,\varsigma) &= \chi + \frac{1}{2}\sin\left(\frac{\pi}{4}\cdot\xi\right) \\ y(\chi,\xi,\varsigma) &= \xi + \frac{1}{2}\sin\left(\frac{\pi}{8}\cdot(\chi+\xi)\right) \\ z(\chi,\xi,\varsigma) &= \zeta + \frac{1}{2}\sin\left(\frac{\pi}{8}\cdot(\chi+\xi+\zeta)\right) \end{aligned} \tag{6.2.2.2}$$

[74] For example, see Roache and Steinberg (1984), Steinberg and Roache (1985, 1986a, 1986b).

Using the bound $|\cos(x)| \le 1$, it is possible to show that the Jacobian of this transformation is strictly positive everywhere. The domain looks basically like a distorted cube.

The Operator

Three possible choices of coefficients define the Operator.

A. Laplace Operator.

$$\begin{aligned} k11(x,y,z) &= 1 \\ k12(x,y,z) &= k21(x,y,z) = 0 \\ k13(x,y,z) &= k31(x,y,z) = 0 \\ k22(x,y,z) &= 1 \\ k23(x,y,z) &= k32(x,y,z) = 0 \\ k33(x,y,z) &= 1 \end{aligned} \tag{6.2.2.3}$$

B. Arbitrary Elliptic Operator with Continuous Coefficients.

$$\begin{aligned} e1(x,y,z) &= 1 + 2x^2 + y^2 + z^2 \\ e2(x,y,z) &= 1 + x^2 + 2y^2 + z^2 \\ e3(x,y,z) &= 1 + x^2 + y^2 + 2z^2 \end{aligned} \tag{6.2.2.4}$$

$$\mathbf{D} = \begin{bmatrix} e1 & 0 & 0 \\ 0 & e2 & 0 \\ 0 & 0 & e3 \end{bmatrix} \tag{6.2.2.5}$$

$$\mathbf{P} = \begin{bmatrix} \cos(u)\cdot\cos(v) & \sin(u) & \cos(u)\cdot\sin(v) \\ -\sin(u)\cdot\cos(v) & \cos(u) & -\sin(u)\cdot\sin(v) \\ -\sin(v) & 0 & \cos(v) \end{bmatrix} \tag{6.2.2.6}$$

with $0 < u, v < 2\pi$, and

$$\mathbf{K} = \begin{bmatrix} k11 & k12 & k13 \\ k21 & k22 & k23 \\ k31 & k32 & k33 \end{bmatrix} \tag{6.2.2.7}$$

Then **D** has positive eigenvalues. If we let

$$\mathbf{K} = \mathbf{P}^{-1}\mathbf{D}\mathbf{P} \tag{6.2.2.8}$$

then **K** has the same eigenvalues. Further, since **P** is an orthogonal matrix, **K** is symmetric. The elements of **K** are as follows.

$$k11(x,y,x) = e1\cdot\cos(u)\cdot\cos(u)\cdot\cos(v)\cdot\cos(v) + e2\cdot\sin(u)\cdot\sin(u)\cdot\cos(v)\cdot\cos(v) + e3\cdot\sin(v)\cdot\sin(v) \tag{6.2.2.9}$$

$$k12(x,y,z) = k21(x,y,z) = (e1-e2)\cdot\cos(u)\cdot\sin(u)\cdot\cos(v) \tag{6.2.2.10}$$

$$k13(x,y,z) = k31(x,y,z) = \cos(v)\cdot\sin(v)\cdot[e1\cdot\cos(u)\cdot\cos(u) + e2\cdot\sin(u)\cdot\sin(u) - e3] \tag{6.2.2.11}$$

$$k22 = e1\cdot\sin(u)\cdot\sin(u) + e2\cdot\cos(u)\cdot\cos(u) \tag{6.2.2.12}$$

$$k23(x,y,z) = k32(x,y,z) = (e1-e2)\cdot\cos(u)\cdot\sin(u)\cdot\sin(v) \tag{6.2.2.13}$$

$$k33(x,y,x) = e1\cdot\cos(u)\cdot\cos(u)\cdot\sin(v)\cdot\sin(v) + e2\cdot\sin(u)\cdot\sin(u)\cdot\sin(v)\cdot\sin(v) + e3\cdot\cos(v)\cdot\cos(v) \tag{6.2.2.14}$$

For the numerical calculations, we took

$$u = \frac{5\pi}{12} \quad \text{and} \quad v = \frac{2\pi}{9} \tag{6.2.2.15}$$

C. Elliptic Operator with Discontinuous Coefficients.

The test case of discontinuous tensor conductivity was established by defining two regions, ***I*** and ***II***, with continuous tensor conductivity in each but with a discontinuity at the boundary. To model the geologic situation, normal fluxes across a geological boundary were assumed continuous.

$$\left[K\frac{\partial f}{\partial n}\right]_I = \left[K\frac{\partial f}{\partial n}\right]_{II} \tag{6.2.2.16}$$

$$\mathbf{K}\ \frac{\partial f}{\partial n} = [n1 \quad n2 \quad n3]\begin{bmatrix} k11 & k12 & k13 \\ k21 & k22 & k23 \\ k31 & k32 & k33 \end{bmatrix}\begin{bmatrix} fx \\ fy \\ fz \end{bmatrix} = nK\nabla f \tag{6.2.2.17}$$

The unit cube domain was divided into the two regions ***I*** and ***II*** wherein ***I*** was an ellipse embedded in the unit cube.

$$\boldsymbol{I} = \{(x, y, z)\ in \quad U \mid \quad r < 2\} \tag{6.2.2.18}$$

$$\boldsymbol{II} = \{(x, y, z)\ in \quad U \mid \quad r > 2\} \tag{6.2.2.19}$$

$$r = \sqrt{(1+x)^2 + 12y^2 12z^2} \tag{6.2.2.20}$$

On region ***I***, we let

$$\mathbf{D} = \begin{bmatrix} 1.5 & 0 & 0 \\ 0 & 1.0 & 0 \\ 0 & 0 & 0.75 \end{bmatrix} \tag{6.2.2.21}$$

with $u = \dfrac{5 \cdot \pi}{12}$ and $v = \dfrac{2 \cdot \pi}{9}$. Then

$$\begin{aligned} k11 &= 0.916361, \quad k12 = 0.095755, \\ k13 &= 0.139593, \quad k22 = 1.466506, \\ k23 &= 0.080348, \quad k33 = 0.867133. \end{aligned} \tag{6.2.2.22}$$

On region ***II***, we let

$$\mathbf{K}_{II} = \rho \cdot \mathbf{K}_{I} \tag{6.2.2.23}$$

with $\rho > 1$. For our tests, $\rho = 2$. We then took as the solution $f(x, y) = F(r)$, so that

$$F_I(r) = \rho \cdot \ln(r) \tag{6.2.2.24}$$

$$F_{II}(r) = \rho \cdot \ln(2) + 0.5 \cdot \ln(r - 1) \tag{6.2.2.25}$$

and the flux continuity condition reduces to

$$\left[\mathbf{K} \quad \mathbf{F}_r\right]_I = \left[\mathbf{K} \quad \mathbf{F}_r\right]_{II}. \tag{6.2.2.26}$$

This is automatically satisfied by the choice of $F(r)$. Furthermore, $F(r)$ is continuous and differentiable (even at $r = 2$), as is needed for F is to represent pressure or head.

The Boundary Conditions

The general Robin boundary condition is

$$\alpha \cdot \frac{\partial f}{\partial n} + \beta \cdot f = \gamma \tag{6.2.2.27}$$

for each (continuum) point on the boundary of domain. For each of the four boundaries, the user selected from the following choices.

A. All Dirichlet: $\alpha = 0, \beta = 1$
B. All Neumann: $\alpha = 1, \beta = 0$
C. All Robin: $\alpha = 1, \beta = 1$
D. Random: $0 < \alpha, \beta < 1$ (6.2.2.28)

Choice D uses α and β varying randomly along a boundary.

The γ was then computed, based on the user's choice of solution function, *f*.

A. $f = 1$ on all of domain ($g = 0$ for Laplace Operator) (6.2.2.29)

B. $f = \sin(\pi \cdot x) \cdot \sin(\pi \cdot y) \cdot \sin(\pi \cdot z)$ (6.2.2.30)

C. $f = F(r)$ for discontinuous tensor (6.2.2.31)

The source term *g* is also computed from *f*, using $g = Lf$. The values of *g* and *f* are computed at cell centers, while γ is computed on boundary cell faces. For the discontinuous tensor case, $F(r)$ given in (6.2.2.24–25) must be selected.

Interface Conductivity Evaluation

Finally, the user can define PDE coefficients (i.e. the conductance, a combination of the physical variable conductivity and discretization terms) at the cell interfaces either by linear or harmonic averages. For a uniform grid, harmonic averaging is equivalent to defining *k* at the cell face ($i + 1/2$) as

$$k(i+1/2) = \frac{2k(i)k(i+1)}{k(i)+k(i+1)} \tag{6.2.2.32}$$

as opposed to the more obvious linear average

$$k(i+1/2) = \frac{k(i)+k(i+1)}{2} \tag{6.2.2.33}$$

Harmonic averaging is commonly used in groundwater flow codes.

Debugging Successes

Several bugs in the 2-D codes were uncovered during the course of this Code Verification procedure.

i. The coefficient averaging procedure was not defined on the physical boundary of the problem, giving zero coefficient values there. This was easily remedied by changing the way in which the metric coefficients were computed in the stencil-loader.
ii. An error in the point-SOR routine at the corner stencils prevented 2nd-order convergence there.

iii. Two of the boundary loops used to compute the right-hand-side arrays in the stencil-loader were indexed over "*i*" when they should have been indexed over "*j*," causing incorrect zeros in the right hand side array.

iv. The corner stencil formula, e.g. at the lower left corner (*i*, *j*), was modified from

$$f(i, j) = \frac{1}{2}[f(i, j+1) + f(i, j-1)] \tag{6.2.2.34}$$

to

$$f(i, j) = f(i+1, j) + f(i, j-1) - f(i+1, j+1) \tag{6.2.2.35}$$

to preserve 2nd-order convergence there. (This also required adding a few lines of code setting the right-hand-side arrays to zero during the computation of the corner stencils). With the 2-D code fixes built into the *n*-dimensional Symbolic Manipulation code, the 3-D tests uncovered only one additional bug.

v. The right-hand-side arrays on the edges of the cube were not properly initialized to zero (being outside the needed loops).

When these items were corrected, the 2-D and 3-D stencil-loader routines were Verified to produce 2nd-order accurate stencils for problems on general domains using non-orthogonal grids, provided smooth PDE coefficients were used. Solution function (B), Eq. (6.2.2.30) was used in all the runs shown here.

The full set of 3-D problems is described in Table 6.2.2.1. Linear averaging of the conductivities was used except in Run 10. Numerical results are given here only for Runs 1, 8, 9 and 10.

The symmetry of the solution and boundary conditions made Run 1, shown in Table 6.2.2.2, ideal to check the case $\Delta x \neq \Delta y \neq \Delta z$. As seen in the Table, the truncation error is the same no matter which direction is refined, and the maximum error occurs in the 21st-cell in the direction having 40 cells. Therefore, we can safely conclude there are no coding errors related to $\Delta x \neq \Delta y \neq \Delta z$.

In another experiment, Run 1 was repeated with the PDE coefficients set to zero in the ring of cells outside the physical boundary to obtain some idea of the effect of extrapolating these coefficients. The results showed that the accuracy was degraded to roughly 1st-order.

Run 8 in Table 6.2.2.3 shows the 2nd-order convergence of the continuous operator with Robin boundary conditions on an irregular domain, exercising all the metric terms in the 3-D boundary fitted coordinate transformation.

Runs 9 and 10 in Tables 6.2.2.4 and 6.2.2.5 show the convergence of the discontinuous coefficient problem on the unit cube, for linear and harmonic averaging of conductivities.

The theoretical basis for harmonic averaging (e.g. see McDonald and Harbaugh, 1988) is well established in 1-D. We found 2nd-order convergence in 1-D provided that the discontinuity stayed on a cell interface as the grid was refined, and that harmonic averaging was used; linear averaging produced only 1st-order convergence. However, nonorthogonal grid multidimensional results in Tables 6.2.2.4-5 above are not impressive. For a discontinuous k with ratio $\rho = 2$ (which is mild by geologic standards), 3-D harmonic averaging is slightly worse than linear averaging at $N = 5$, and slightly better at higher resolution. Both certainly give a consistent discretization, but both are only 1st-order accurate. (Note the unity exponent in the last column.) The order of convergence was *not* restored to 2nd-order by aligning the grid with the discontinuity.

After the publication of Roache et al (1990), the basis for this section, Shashkov and Steinberg (1995,1996) showed how to correctly generalize the concept of harmonic averaging to 2-D. The solution is elegant but surprisingly expensive. (See also Shashkov, 1997 and Hyman et al, 1997.)

The above tests Verify the accuracy of the spatial differencing. The time differencing was separately Verified by a simple test of the decay of a single Fourier component, the solution being obtained by elementary separation of variables (Roache et al, 1995). The separation of Verification for spatial and temporal differencing is justified only by the knowledge that the algorithms did not mix space and time derivatives (unlike algorithms such as Approximate Factorization, for example). One could argue that this is cheating on the Verification process, since it is not, strictly speaking, treating the code as a "black box." As noted elsewhere, this approach amounts to partitioning the option matrix of the code, and greatly reduces the complexity of the option combination problem, if it can be justified.

This partitioning is further made palatable, even to one who is not familiar with the code structure, by the modest claim of only O(Δt) accuracy. In fact, it is difficult to imagine a non-contrived coding error that could produce reasonable time solutions and not be at least O(Δt) accurate. As noted in Chapter 3, Section 3.14 titled **"**Warnings: What the Method Does Not Verify," it is easy to make coding mistakes that do not strictly follow the algorithm as intended, yet do not affect the order of time accuracy. For example, in the fully implicit (backward time) differencing algorithm in the presently considered codes, the intention was to set the boundary conditions at the advanced time level (n + 1). If a coding mistake is made that results in some boundary values being lagged, i.e. set at time level n, stability might be affected, but the order of convergence would still test at O(Δt). The *size* of the error might be (slightly) affected, compared to the

Run	Domain	Operator	B.C.
1	Unit Cube	Laplace	Dirichlet
2	Irregular	Laplace	Dirichlet
3	Unit Cube	Continuous	Dirichlet
4	Irregular	Continuous	Dirichlet
5	Unit Cube	Laplace	Robin
6	Irregular	Laplace	Robin
7	Unit Cube	Continuous	Robin
8	Irregular	Continuous	Robin
9	Unit Cube	Discontinuous	Dirichlet
10	Unit Cube	Discontinuous–harmonic average	Dirichlet

Table 6.2.2.1. Conductivity Problem Configurations. (From Table 3 of Roache et al, 1990.)

N	xerr	at	xerr · N^2
5 × 5 × 5	0.03356	(3, 3, 3)	0.839
10 × 10 × 10	0.00796	(6, 6, 6)	0.796
20 × 20 × 20	0.00202	(11, 11, 11)	0.808
40 × 40 × 40	0.00051	(21, 21, 21)	0.816
5 × 5 × 40	0.02224	(3, 3, 21)	–
5 × 40 × 5	0.02226	(3, 21, 3)	–
40 × 5 × 5	0.02226	(21, 3, 3)	–

Table 6.2.2.2. Run 1. (From Table 4 of Roache et al, 1990.)

N	xerr	xerr $\cdot N^2$
5	0.06970	1.743
10	0.02119	2.119
20	0.00559	2.236
40	-	-

Table 6.2.2.3. Run 8. (From Table 5 of Roache et al, 1990.)

N	xerr	xerr $\cdot N^1$
5	0.05395	0.27
10	0.02808	0.28
20	0.01442	0.29
40	–	–

Table 6.2.2.4. Run 9: Linear Averaging for K.
(From Table 6 of Roache et al, 1990.)
(Note the unity exponent in the last column.)

N	xerr	xerr $\cdot N^1$
5	0.05906	0.29
10	0.02079	0.21
20	0.01301	0.26
40	0.00688	0.28

Table 6.2.2.5. Run 10: Harmonic Averaging for K.
(From Table 7 of Roache et al, 1990.)
(Note the unity exponent in the last column.)

intended algorithm, but the *order* would not be changed. On the other hand, $O(\Delta t^2)$ algorithms, especially those like Approximate Factorization that mix space and time differences, can easily be corrupted by coding mistakes to $O(\Delta t)$. More scrupulous Verification is required, as in Salari and Blaine (1996); see Section 6.24.

6.3 ISSUES IN CALCULATION VERIFICATION

The above examples involved accuracy Verification of Codes. The following examples involve Verification of individual Calculations, and assume the use of an already Verified code. This initial discussion on issues in calculation Verification is taken from Westerink and Roache (1995), "Issues in Convergence Studies in Geophysical Flow Computations."

In performing systematic grid convergence studies, there are a variety of issues which lead to a level of uncertainty in the estimate which must be taken into consideration. These issues include the following.

- Higher order truncation term interaction even in the asymptotic range
- Non-asymptotic truncation term interaction
- Time-space truncation error interaction and related superconvergence
- Effect of parameter convergence

The wavenumber distribution of the response will also affect observed convergence rates, and the response spectra will include flow features artificially generated by the numerics.[75] The computations may also experience nonuniform behavior of various error norms. All these possible factors, and the others noted earlier in Chapter 5, introduce uncertainty into the Verification process based on Richardson Extrapolation or other methods, and provide further rationale for the safety factor $F_s > 1$ in the Grid Convergence Index.

6.3.1 Formal, Actual and Observed Convergence Rates

It is a common feature of discrete solutions to PDEs to encounter convergence rates for complicated applications which are less than formal rates. It is assumed that the formal convergence rate is indicated by the leading order space and/or time truncation error terms. However, even in the asymptotic range (where discrete space and time steps tend to zero), formal convergence rates may never be achieved, leading to the definition of the *actual asymptotic* convergence rate. Even neglecting the well-recognized problems with computer round-off error, this *actual asymptotic* convergence rate may be different from the *formal* convergence rate when, for example, the formal analysis misses interaction between the leading order and subsequent order truncation terms in the error series (or when other approximations are required). This can come about due to the form of the truncation terms, which consists of products of the space step to a power and higher order derivatives of the response function. When gradients of the response function increase with increasing grid refinement, as they do in many nonlinear and/or complicated flows, the higher order terms in the truncation series continue to compete with the leading order truncation term.

Finally, the convergence rate actually *observed* in numerical experiments may be different from *either* the *formal* or *actual asymptotic* convergence rates, simply due to grid resolution not being adequate to achieve the asymptotic range. When the coarse grid Richardson Error Estimator does not equal (approximately) the fine grid Richardson Error Estimator computed for the same grid, this is an indication that the formal convergence rate is not being achieved. It is in fact common for the fine grid error estimator to be less than the coarse error estimator for the same grid, indicating that the observed convergence rate is less than the formal rate. A more complete discussion of the variety of factors that can influence the observed convergence rates will be given in Chapter 8, Section 8.1.

6.4 TWO EXAMPLES OF THE EFFECTIVE GRID REFINEMENT RATIO

6.4.1 A Posteriori Application of GCI Scaling

As an example of the effect of grid refinement ratio on the reporting of grid convergence tests, consider the following Figure 6.4.1 for local Nusselt numbers (normalized heat transfer coefficients) taken from the second-order accurate Finite Element solutions by Comini et al (1995).

The presentation of results looks converged almost to the plotting accuracy, perhaps 0.1%. However, the grid refinement is not a grid doubling, as noted in the legend. To calculate an effective grid refinement

[75] See also Hoekstra et al (2000b) for convergence study in wavenumber space.

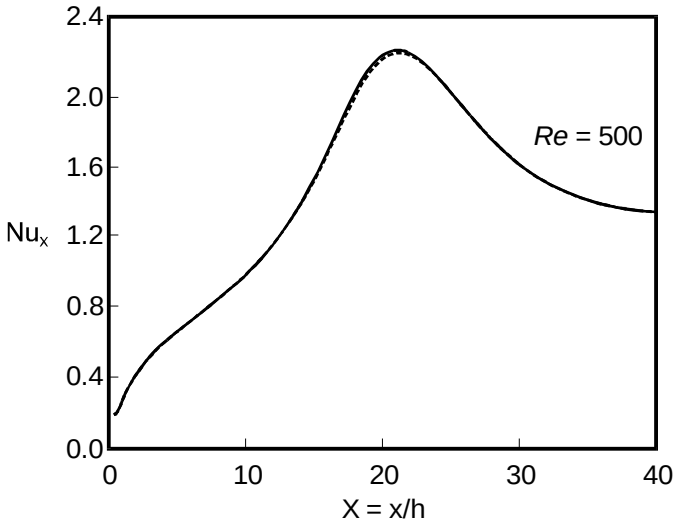


Figure 6.4.1. Grid dependence of calculated distributions of local Nusselt numbers. Pr = 0.7, *Re* = 500. Solid lines represent results from a mesh with 1024 parabolic elements and 3273 nodes. Dashed lines represent results from a mesh with 1504 parabolic elements and 4725 nodes. (From Figure 4 of Comini et al, 1995.)

ratio r as suggested in Chapter 5 for this unstructured grid, we could use either the ratio of (parabolic) elements, giving Eq. (6.4.1).

$$r_{elements} = \left(\frac{1505}{1024}\right)^{1/2} = 1.21 \tag{6.4.1}$$

or the ratio of nodes,

$$r_{elements} = \left(\frac{4725}{3273}\right)^{1/2} = 1.20 \tag{6.4.2}$$

which are equivalent within the approximation of the empiricism of the concept of effective r for unstructured grid refinement. With $p = 2$ for the parabolic elements used by Comini et al, this gives

$$\text{GCI} = \frac{3\varepsilon}{r^p - 1} = \frac{3\varepsilon}{1.2^2 - 1} = 6.64\varepsilon \tag{6.4.3}$$

With the small $\varepsilon \sim 0.1\%$ this still indicates a level of grid convergence that is totally acceptable, especially considering the conservatism of the "safety factor" of $F_s = 3$ in the GCI; yet, it is considerably less converged than suggested by the original reporting. Even if we use $F_s = 1$, so that the reported GCI equals the Richardson Error Estimator, corresponding roughly to only a 50% confidence band, we would obtain

$$GCI(F_s = 1) = 2.21\varepsilon \tag{6.4.4}$$

somewhat less impressive than just ε. If results similar to Figure 6.4.1 had been obtained with only first-order methods, the discrepancy in the reporting would be marked. For $F_s = 3$, we would obtain

$$GCI = 14.63\varepsilon \tag{6.4.5}$$

and for $F_s = 1$,

$$GCI = 4.88\varepsilon \tag{6.4.6}$$

again showing the difficulty of judging grid convergence with first-order methods.

6.4.2 § Justification of Effective Grid Refinement Ratio for Heat Conduction

The adequacy of using a simple effective *r*, as in Eq. (5.10.3.3.1), will depend on the target metric of the unstructured grid generation algorithm and the success of attaining it. If the grid generation algorithm is solution adaptive, there is no reason to expect the effective *r* to be adequate. But many unstructured FEM mesh generators allow input of user-defined normalized mesh density functions. With relative mesh density defined, the user can specify quasi-uniform mesh refinement by a factor that corresponds to *r*. True grid similarity is not possible (it is not even meaningful), but if mesh density is refined in an approximately uniform manner, we would expect the GCI based on effective *r* to provide good uncertainty estimates. Such a successful application was accomplished by Dr. K. Dowding in Section 7 of V&V20 on the heat conduction in the cooling fin of a heat exchanger. Effective *r* was successfully used in both Code Verification and Calculation Verification exercises. The four FEM meshes in the sequence contained 39, 173, 680 and 2769 elements, as shown in Figure 6.4.2.1.

Details of the particular parameters used can be found in V&V20, but these do not affect the result of interest here. The locations 1 and 2 are near the mid-point of the circular segment on the left, referred to as Surface S1. The mesh sequence was defined to have a node at both these locations in all four meshes. If the mesh sequence were not defined with a node at these locations, the code output would need to be interpolated from the nodal solution to give the solution at the prescribed locations. Table 6.4.2.1 gives the results from the Code Verification study using MMS. Apparently, the coarsest mesh (mesh 4) is more or less on the border of the asymptotic region for the local temperature in this conduction problem. The other values are fairly well behaved, even for these unstructured grid sequences.

Alternately, the convergence rate for the sequence of meshes can be estimated with standard regression on the observed ($\log(h)$, $\log(E_h)$) data over all 4 meshes. This result is listed in the last row of Table 6.4.2.1 for comparison.

Meshes	Temp Loc 1	Temp Loc 2	Flux S_1	L_2 Norm Temp
3 and 4	1.77	1.82	2.19	2.10
2 and 3	1.96	2.03	3.16	1.96
1 and 2	2.07	1.97	2.13	1.97
Reg (1- 4)	1.93	1.94	2.55	2.01

Table 6.4.2.1. Observed order of convergence p_{obs} calculated using effective *r* in an unstructured mesh refinement for a heat transfer problem with manufactured solution (MMS). From Table 7-4, V&V20.

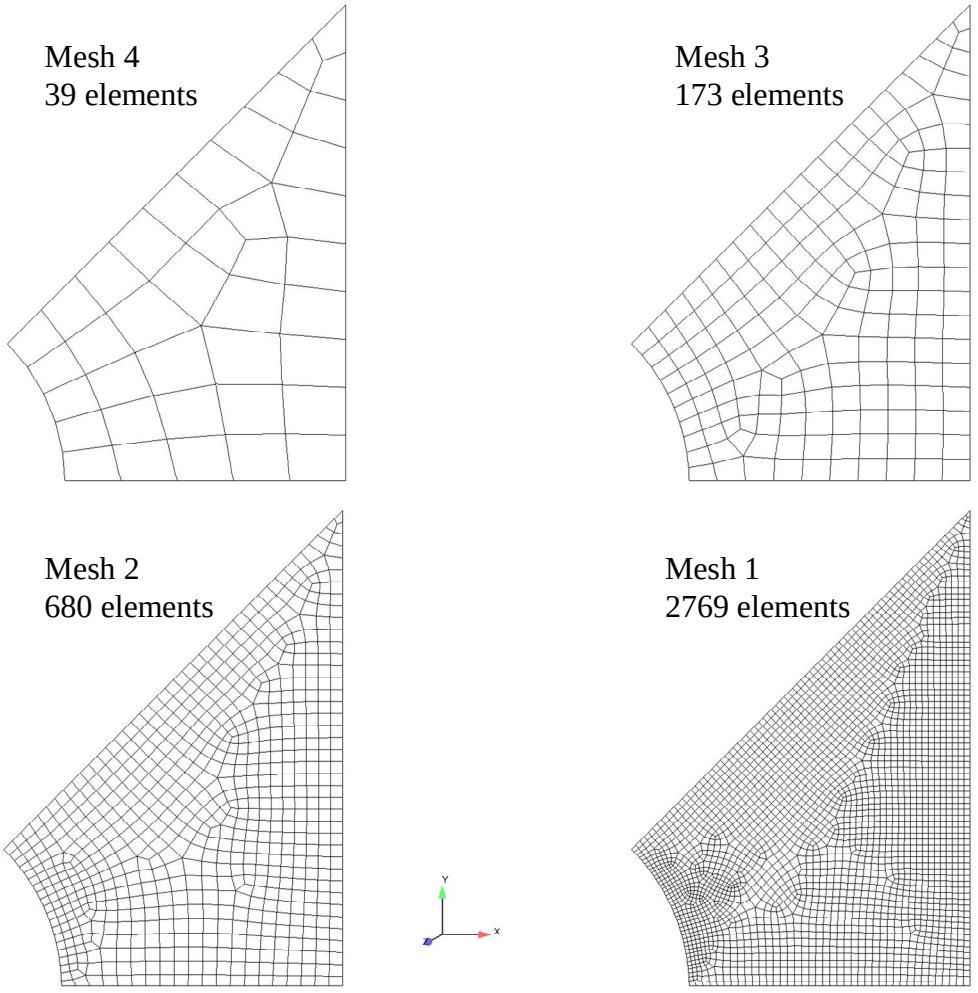


Figure 6.4.2.1 Unstructured finite element meshes used in the mesh refinement study using effective *r*. (From Figure 7.2 of V&V20.)

For this MMS problem, the exact solution is known, so the observed p can be calculated from any two meshes. For the simulation problem (which differs from the MMS in boundary conditions and source term) the exact solution is not known, so three meshes are needed to calculate observed *p*. The mesh triplet of the three coarsest meshes (2,3,4) gave observed $p = 1.99$, and the triplet (1,2,3) gave 2.01.

These results for code verification and calculation verification give further confidence in the use of an effective *r* for unstructured grids. However, heat conduction is a benign problem, compared to those with advection terms, turbulence models, etc. Also, the performance will be affected by the quality of the grid generation algorithm and its ability to approximately match user-specified mesh density distributions, which itself will be affected by the particulars of the problem geometry.

6.5 BENCHMARK PROBLEMS FOR DRIVEN CAVITY FLOW

The simple driven cavity problem has been used for decades (Roache, 1972, 1998b) as an easily reproducible CFD problem. Ghia et al (1982) produced an early benchmark solution that was often cited for comparison purposes. Although no error estimate *per se* was included, the results can still be used with confidence for debugging purposes. See Table 6.5.1. More recent higher-resolution solutions were given in Botella and Peyret (1998), Bruneau and Saad (2006), and Prabhakar and Reddy (2006). The latter removed the corner singularity with a lid velocity distribution that = 0 at the corners (see Roache, 1975b).

6.6 BENCHMARK PROBLEM FOR FREE CONVECTION

As common as the driven cavity problem is, it has a disadvantage, in that the moving lid introduces a singularity. In the upper corners, the velocity is discontinuous, and the vorticity is unbounded. (This does not prevent the vorticity from converging on the lid away from the corners; see, e.g., Roache, 1975,1995). De Vahl Davis et al (1979) devised a comparison problem specifically to remove the singularity but maintain the geometric simplicity of the driven cavity. Non-trivial motion is induced by a heated vertical wall. While requiring that the code include buoyancy terms, this problem is more realistic than the pure driven cavity, and contains no singularities. De Vahl Davis and Jones (1983) compared several contributed solutions for this problem, which showed a disturbing range of uncertainty, especially for the early FEM solutions.

De Vahl Davis (1983) produced a classic Benchmark solution of this free convection problem for comparison. He used systematic grid convergence and Richardson Extrapolation very carefully, with an experimental determination rather than an assumption of the *local* order of convergence. Seven years later, Hortmann et al (1990) recalculated this problem with modern multigrid algorithms and higher computer power. The increased resolution (up to 320 × 320 for most of the calculations, and 640 × 640 for one) of course provided more accurate answers and tighter error bands, but the original estimates of de Vahl Davis were confirmed. "For example, [de Vahl Davis] estimated the bounds for [dimensionless heat transfer rate, Nusselt number] to be 0.2%, 0.3% and 1.0% for Rayleigh numbers 10^4, 10^5, and 10^6 respectively; the difference to the present results is 0.08%, 0.06% and 0.2%." Because of complexities of normalizing schemes, Hortmann et al (1990) should be consulted directly for dependable Benchmark comparisons.

6.7 LAMINAR PLANE JET IMPINGING ON A HEATED FLAT PLATE

Pelletier and Ignat (1995) developed a simple analytical solution for the temperature field for an incompressible planar laminar jet impinging on a heated flat plate. They used the solution to calibrate and evaluate their solution adaptive unstructured grid generation methods, and to compare their Zhu-Zienkiewicz type error indicators with the GCI extended to unstructured grids. (See details in the original paper.) The following parameters are: $\alpha = x^2 y$, $a = 5000$, $\gamma = 5$, $\beta = 4$, $\delta_0 = 0.01$ for Re = 50.

$$u = 4x^2 e^{-\alpha^2 a}(1-2\alpha^2 a), \quad v = -8xye^{-\alpha^2 a}(1-2\alpha^2 a) \tag{6.7.1-2}$$

$$p = 1/[1 + C(x^2 + y^2)] \tag{6.7.3}$$

$$T = \exp\left(\frac{\gamma x}{\delta_0(1+\beta y)}\right) \tag{6.7.4}$$

129-Grid Pt. No.	y	Reynolds Number Re 100	400	1000	3200	5000	7500	10,000
129	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
126	0.9766	0.84123	0.75837	0.65928	0.53236	0.48223	0.47244	0.47221
125	0.9688	0.78871	0.68439	0.57492	0.48296	0.46120	0.47048	0.47783
124	0.9609	0.73722	0.61756	0.51117	0.46547	0.45992	0.47323	0.48070
123	0.9531	0.68717	0.55892	0.46604	0.46101	0.46036	0.47167	0.47804
110	0.8S16	0.23151	0.29093	0.33304	0.34682	0.33556	0.34228	0.34635
95	0.7344	0.00332	016256	0.18719	0.19791	0.20087	0.20591	0.20673
80	0.6172	–0.13641	0.02135	0.05702	0.07156	0.08183	008342	0.08344
65	0.5000	–0.20581	–0.11477	–0.06080	–0.04272	–0.03039	–0.03800	0.03111
59	0.4531	–0.21090	–0.17119	–0.10648	–0.86636	–0.07404	–0.07503	–0.07540
37	0.2813	–0.15662	–0.32726	–0.27805	–0.24427	–0.22855	–0.23176	–0.23186
23	0.1719	–0.10150	–0.24299	–0.38289	–0.34323	–0.33050	–0.32393	–0.32709
14	0.1016	–0.06434	–0.14612	–0.29730	–0.41933	–0.40435	–0.38324	–0.38000
10	0.0703	–0.04775	–0.10338	–0.22220	–0.37827	–0.43643	–0.43025	–0.41657
9	0.0625	–0.04192	–0.09266	–0.20196	–0.35344	–0.42901	–0.43590	–0.42537
8	0.0547	–0.03717	–0.08186	–0.18109	–0.32407	–0.41165	–0.43154	–0.42735
1	0.0000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

129-Grid Pt. No.	x	Reynolds Number Re 100	400	1000	3200	5000	7500	10,000
129	l.0000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
125	0.9688	–0.05906	–0.12146	–0.21388	–0.39017	–0.49774	0.53858	–0.54302
124	0.9609	–0.07391	–0.15663	–0.27669	–0.47425	–0.55069	0.55216	–0.52987
123	0.9531	–0.08864	–019254	–0.33714	–0.52357	–0.55408	0.52347	–0.49099
122	0.9453	–0.10313	–0.22847	0.39188	–054053	–0.52876	–0.48590	–0.45863
117	0.9063	–0.16914	–0.23827	0.51550	–0.44307	–0.41442	–0.41050	–0.41496
111	0.8594	–0.22445	–0.44993	–0.42665	–0.37401	–0.36214	–0.36213	–0.36737
104	0.8047	–0.24533	–0.38598	–0.31966	–0.31184	–0.30018	–0.30448	–0.30719
65	0.5000	0.05454	0.05186	0.02526	0.00999	0.00945	0.00824	0.00831
31	0.2344	0.17527	0.30174	0.32235	028188	0.27280	0.27348	0.27224
30	0.2266	0.17507	0.30203	0.33075	0.29030	0.28066	0.28117	0.28003
21	0.1563	0.16077	0.28124	0.37095	0.37119	0.35368	0.35060	0.35070
13	0.0938	0.12317	0.22965	0.32627	0.42768	0.42951	0.41824	0.41487
11	0.0781	0.10890	0.20920	0.30353	0.41906	0.43648	0.43564	0.43124
10	0.0703	0.10091	0.19713	0.29012	0.40917	0.43329	0.44030	0.43733
9	0.0625	0.09233	0.18360	0.27485	039560	0.42447	0.43979	0.43983
1	0.0000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

Table 6.5.1. Results for *u*- and *v*-velocity components for driven cavity flow, from Ghia et al (1982).

6.8 *K*-ε MODEL OF A FREE SHEAR LAYER

In the same paper, Pelletier and Ignat (1995) developed a simple analytical solution for an incompressible free shear layer using the *k*-ε turbulence model. As in the previous section, they used this solution to compare their Zhu-Zienkiewicz type error indicators with the GCI extended to unstructured grids. (See details in the original paper.) The solution should be of considerable interest for turbulence modelers, since it displays most of the characteristics of an experiment of Patel (1973) yet is convenient to encode. Likewise, they developed analytical Benchmarks for variants of the *k*-ε model, the *k*-τ model and for the Wilcox (1993) *k*-ω model and variants.

$$u = \left(\frac{u_1 + u_2}{2}\right) - \left(\frac{u_1 - u_2}{2}\right) erf\left(\frac{\sigma y}{x}\right) \tag{6.8.1}$$

$$\nu = \nu_0 + \left(\frac{u_1 - u_2}{2}\right) \frac{1}{\sigma\sqrt{\pi}} \exp\left(-\left(\frac{\sigma y}{x}\right)^2\right) \tag{6.8.2}$$

$$B = \exp\left(-\left(\frac{\sigma y}{x}\right)^2\right) \tag{6.8.3}$$

$$k = k_0 (c_k + B) \tag{6.8.4}$$

$$\varepsilon = \frac{\varepsilon_0}{x} (c_k + B) \tag{6.8.5}$$

$$\mu_T = \mu_{T0} x (c_k + B) \tag{6.8.6}$$

$$p = 0, u_1 = 1.0, u_2 = 0.0, \sigma = 13.5 \tag{6.8.7}$$

$$k_0 = \frac{343}{75{,}000} u_1 (u_1 - u_2) \frac{\sigma}{\sqrt{\pi}} \tag{6.8.8}$$

$$\varepsilon_0 = \frac{343}{22{,}500} C_\mu u_1 (u_1 - u_2)^2 \frac{\sigma^2}{\pi} \tag{6.8.9}$$

$$\mu_{T0} = \frac{343}{250{,}000} \rho u_1 \tag{6.8.10}$$

$$Re_l = \frac{\rho u_1 L}{\mu_l} = 10^4 \tag{6.8.11}$$

6.9 TRANSONIC AIRFOIL CALCULATIONS

Zingg (1991,1992) applied the Richardson Error Estimator to airfoil lift and drag calculations in body-fitted grids. Zingg's work demonstrated the necessity of grid convergence testing even when experimental data are available. In 4 of 7 cases, experimental agreement was better with *coarse* grid calculations than with fine (Zingg, 1992).

The results of the Viscous Transonic Airfoil Workshop (Holst, 1987) indicated that "while turbulence modeling is a significant source of error in the computation of airfoil flows, particularly for separated flows, a large portion of the error is often associated with inadequate numerical resolution and an insufficient distance to the outer boundary of the grid" (Zingg, 1991). Zingg's work convincingly brought these errors well within engineering accuracy. He used Pulliam's well-established viscous airfoil code ARC2D (Pulliam, 1986) and the algebraic Baldwin-Lomax turbulence model with the thin-layer approximation to the full Reynolds-Averaged Navier-Stokes equations on structured non-orthogonal C-type grids. Zingg presented results for 8 different flow cases and two fine base grids, each base grid being coarsened two times by removal of alternate cells in each direction; thus, $r = 2$ and $p = 2$ in the Richardson extrapolations. Here, we present the lift and drag results for just two flow cases and the A7A base grid of 497×194 points, with 401 of the 497 points on the body and 48 in the wake. Points were clustered at the leading and trailing edges with a spacing of 10^{-4} chords; the normal spacing at the body was 10^{-7} chords. The most coarse grids used still had adequate resolution of the viscous sublayer (with the largest $y^+ = 0.2$ for the first grid point, less than the commonly used value of $y^+ = 1$). The outer boundary distance was 12 chords.

The first result presented here is for *subsonic* flow, Zingg's Case 3, a NACA 0012 airfoil at $M = 0.16$ and angle of attack $\alpha = 12°$. The lift and drag coefficients are given in Table 6.9.1.

For this case, trailing edge separation was predicted on the upper surface (at about 96% of chord for the fine grid calculation.) Second order accuracy is Verified by the reduction in % changes by factors of 4 as the grid is refined by factors of 2, from coarse to medium, and medium to fine, correct to the appropriate level of precision. This is a difficult case as judged by the drag calculation, which is 5.2% in error even for the fine grid of 497×194 points, compared to the best RE (**R**ichardson **E**xtrapolation) results (using the fine and medium grids). As Zingg noted, "Richardson Extrapolation is very effective for this case," as can be judged by evaluating the results of RE from the medium and coarse grids, labeled "RE m-c" in the Table 6.9.1. These give accuracy roughly equal to the finest grid for lift, and about a 9× improvement for drag ($\%C_D = 0.6$ compared to 5.2). Zingg noted the favorable economics involved with grid coarsening (see

Grid	C_L	C_D	% C_L	% C_D
Fine	1.3081	0.01327	–0.3	5.2
Medium	1.2957	0.01526	–1.3	21.0
Coarse	1.2563	0.02299	–4.3	82.3
RE m-c	1.3088	0.01268	–0.3	0.6
RE f-m	1.3122	0.01261	–	–

Table 6.9.1. Lift and drag coefficients for NACA 0012 airfoil at $M = 0.16$ and $\alpha = 12°$. The "RE f-m" results were obtained by **R**ichardson **E**xtrapolation from the fine and medium grid solutions, and "RE m-c" from the medium and coarse grids. The %'s are changes from the RE results. (From Table 3 of Zingg, 1991.)

Section 5.17); a grid coarsened by removal of every other point gains a factor of 4 in number of points, and (roughly) another factor of 2 in number of time steps, so RE costs about an additional 1/8 or 12.5% of computer time. Applying these ratios to this case, the medium grid calculation requires about 1/8 of the fine grid calculation, and the coarse grid calculation requires another $(1/8)^2$. So the medium-coarse RE requires about 9/64 or 14% of the time of a single fine grid calculation.

Allowing a little slack for additional arithmetic, the Richardson Extrapolation method on medium and coarse grids produces about the same error as the fine grid on lift and about 9× better on drag, plus it provides an error estimate (whereas the fine grid calculation alone does not), at about 15% of the cost!

The economics are improved in 3-D by another factor of 2, so RE m-c would cost about $(1/16 + 1/16^2)$ = 17/256 or about 7% of the cost of a single fine grid solution. The cost could further be reduced by using coarse grid solutions as initial guesses for the finer grids.

Note, however, as discussed in Chapter 3, that we really require 3 grid solutions to be more confident that we are in the asymptotic range, and to evaluate the observed order of convergence *p*. Zingg found that the 3-grid solutions were well-behaved for the subsonic cases such as the above. But for transonic cases, he found that "the streamwise point spacing of the [coarse] grid is not sufficient, and shocks are excessively smeared. As a result, the extrapolated solutions [from the medium and coarse grids] generally do not reduce the error." Accordingly, the coarse grid results are not included herein for the *transonic* Case 5, a NACA 0012 airfoil at $M = 0.55$ and $\alpha = 8.34°$, shown in Table 6.9.2.

Grid	C_L	C_D	% C_L	% C_D
Fine	0.9977	0.03484	–0.4	0.2
Medium	0.9868	0.03509	–1.4	0.9
RE f-m	1.0013	0.03476	–	–

Table 6.9.2. Lift and drag coefficients for NACA 0012 airfoil at $M = 0.55$ and $\alpha = 8.34°$. "RE f-m" results were obtained by **R**ichardson **E**xtrapolation from the fine and medium grid solutions. The %'s are changes from the RE results. (From Table 5 of Zingg, 1991.)

For this case, "there is a strong shock on the upper surface at about 20% chord. There is a small separated flow region after the shock and the flow finally separates at about 90% chord." The fine grid solution is good, predicting lift and drag to within 0.4% and 0.2% respectively of the RE solution. [The GCIs for lift and drag, using the very conservative $F_s = 3$, would be 1.2% and 0.6%.] This is a high quality numerical solution for a strong shock transonic flow. Zingg noted that "the second-order artificial dissipation required at shocks scales with the grid spacing to first order. Therefore, Richardson Extrapolation can only be used if the global influence of the second-difference dissipation is small. This is likely if shocks are weak or well-resolved." [Alternately, separate Richardson extrapolation can be performed separately for the dissipation, as in Blottner (1990) and Kuruvila and Anderson (1985); see Section 6.11.] Generally, Zingg (1991) found that the error in pressure drag is larger than the error in friction drag. See the original paper for these and other additional results, and comparison with experiments. (See also comments in Section 10.5.)

6.10 Δ FAR FIELD BOUNDARY ERRORS

The one observation that can be confidently made about far-field boundary errors is that they are seldom evaluated. These errors are not ordered in Δ and therefore are not revealed in a grid convergence test. They may be considered part of the conceptual model error (like incompressibility or two-dimensionality) but this does not excuse the analyst from some attempt to estimate them. The importance is highly problem dependent, and demonstrably will differ even within the same problem for different quantities of interest.

6.10.1 Ordered Estimation of Far-Field Boundary Errors

In a later paper (Zingg, 1992) on airfoil calculations, Zingg systematically varied the position of the outer computational boundary, i.e. the far-field boundary. Inspection of his data indicates that Richardson Extrapolation can be applied to the estimation of far-field boundary errors, with the error proving to be 1st-order in the inverse of distance to the boundary, as now shown.

Zingg's (1992) Case 3 was a subsonic NACA 0012 airfoil at $M = 0.16$, $\alpha = 12°$, $Re = 2.88\times10^6$, with transition at 0.01 chords on the upper surface and 0.95 on the lower. From inspection of the data in his Table 8, "Results of outer boundary position studies," it appeared that the errors for Case 3 were approximately ordered in $1/L_b$, the distance to the outer boundary. Calculations proved to be remarkably consistent with this assumption. The Case 3 data are reproduced below in Table 6.10.1. The $\%C_L$ and $\%C_D$ were calculated as differences from the 96 chord solution.

Note that both $\%C_L$ and $\%C_D$ approximately doubled going from 24 to 12 chords, consistent with the assumption of boundary error $BE \sim 1 / L_b$ where L_b is the distance from the airfoil to the outer boundary. If we make this assumption, extrapolate to $L_b = \infty$, and re-calculate $\%C_L'$ and $\%C_D'$ as differences from this extrapolated solution (as one does with the mesh spacing results), the behavior even more accurately fits the model of $BE \sim 1 / L_b$, as now shown.

If we were doing Richardson Extrapolation from $L_b = 12$ and 24 with the assumption that $BE \sim 1/L_b^p$ and $p = 1$, it would look like the usual grid extrapolation with grid doubling and $p = 1$, i.e.

$$C_L \sim C_{L24} + (C_{L24} - C_{L12}) \tag{6.10.1}$$

However, since we are doing the analog of a grid quadrupling when we go from $L_b = 24$ to 96, the extrapolation now becomes

$$C_L' \sim C_{L96} + (C_{L96} - C_{L24})/3 \tag{6.10.2}$$

L_b (chords)	C_L	C_D	$\%\ C_L$	$\%\ C_D$
96	1.3111	0.01397	–	–
24	1.3121	0.01376	0.08	–1.5
12	1.3139	0.01349	0.2	–3.4

Table 6.10.1. Effect of outer boundary position on lift and drag coefficients of transonic airfoil calculations. $\%\ C_L$ and $\%\ C_D$ are differences from the 96 chord boundary values. (From Zingg, 1992, Case 3, Table 8.)

L_b (chords)	C_L	C_D	$\%C_L'$	$\%C_D'$
96	1.3111	0.01397	0.02289	–0.4986
24	1.3121	0.01376	0.09917	–1.9943
12	1.3139	0.01349	0.23649	–3.9174

Table 6.10.2. Effect of outer boundary position on lift and drag coefficients of transonic airfoil calculations. $\%C_L'$ and $\%C_D'$ are differences from extrapolated boundary values.

(That is, the factor of 1/3 appears either from the common doubling with $p = 2$, or from quadrupling with $p = 1$.) Applying this gives

$$C_L' \sim 1.3111 + \tfrac{1}{3}(1.3111 - 1.3121) = 1.3108 \tag{6.10.3}$$

$$C_D' \sim 0.01397 + \tfrac{1}{3}(0.01397 - 0.01376) = 0.01404 \tag{6.10.4}$$

Recalculating $\%C_L'$ based on the differences between C_L and C_L' (rather than C_{L96}), and similarly for $\%C_D'$, gives the results in Table 6.10.2. (I have kept more significant figures than justified in order to show the asymptotic behavior.)

For the assumed $BE \sim 1 / L_b$, the theoretical ratios R of successive $\%C_L'$ and $\%C_D'$ would be 4 from 96 to 24 chords, and 2 from 24 to 12 chords. The actual values are shown below in Tables 6.10.3–4.

L_b (chords)	$\%C_L'$	R actual	R theory
96	0.02289		
24	0.09917		
		4.332	4
12	0.23649		
		2.38470	2

Table 6.10.3. Extrapolations for Boundary Error in C_L.

L_b (chords)	$\%C_D'$	R actual	R theory
96	–0.4986		
24	–1.9943		
		3.9998	4
12	–3.9174		
		1.9643	2

Table 6.10.4. Extrapolations for Boundary Error in C_D.

As in the case of Zingg's grid size calculations, the extrapolations for C_L are a little noisy, since those calculations are already very accurate. The extrapolations for the less accurate C_D are remarkably consistent with the assumption $BE \sim 1 / L_b$. The need for the unrealistic retention of so many significant figures is now evident; they are required to indicate any departure from the theory! The ratios are exact for anything less than 4 significant figures.

These new error estimates are slightly larger, which is to be expected since the original reference was $L_b = 96$ to approximate $L_b = \infty$ and the convergence is monotone. The original estimate of 1.5% error for $\%C_D$ at $L_b = 24$ now becomes $\%C_D' = 2.0\%$.

6.10.2 § Importance of Far-Field Boundary Errors

Eça and Hoekstra (2009a) studied flow around a tanker at model and full-scale Reynolds numbers using three turbulence models, and systematically examined the influence of far field boundary location, independently varying the width and the length of the computational domain. Perversely, these two had opposite effects. The skin friction resistance was not significantly affected, but the pressure resistance varied by *O*(30%). (32% decrease with width, and 37.5% increase with length.) These results were not significantly affected by grid resolution. Clearly, there are many computational PDE problems that do not involve far-field boundary errors, or that are not strongly affected by far-field modeling, but this cannot always be taken for granted, notably in fluid dynamics. Another important conclusion of this study is that RMS combination of error estimates of outflow boundary errors and discretization error estimates is not conservative; rather, simple addition of absolute values is required. (See also 5.10.10.2-3 for similar restrictions on iteration error estimates.)

6.10.3 § Mapping for Far-Field Boundary Errors

A long-established approach for far-field boundary implementation is to map coordinates for an infinite domain onto a finite computational domain, and apply the conditions commonly used in analytical solutions to these boundaries exactly. It can be a very effective technique, but it is not a cure-all. Many commercial codes do not have this capability. Also, when the transformed grid spacing becomes larger than the scale length of the solution structure, anomalous solutions can result (Roache, 1998b). However, this technique has the advantage of converting the far-field error into an ordered error that can be evaluated by grid convergence tests.

6.11 ARTIFICIAL DISSIPATION EFFECTS

Blottner (1990) has used the same convergence procedure to estimate effects of artificial dissipation terms in hypersonic flow calculations. Blottner's study used the boundary fitted coordinate Navier-Stokes code NS3D in the axisymmetric option to calculate hypersonic ($M = 5$) laminar flow over a spherical nose tip at $Re = 1.8875 \times 10^6$. The code is 2nd-order in space and time, and incorporates both an implicit 2nd-order and an explicit 4th-order artificial dissipation *De*. The form of *De* used is

$$De = \frac{S}{J}\left(\Delta\xi^4 \frac{\partial^4 Q}{\partial \xi^4} + \Delta\varsigma^4 \frac{\partial^4 Q}{\partial \varsigma^4} \right) \tag{6.11.1}$$

Grid Cells	S	$P_{wall}/P_{freestream}$	q_{wall}
22 × 24	0	1.62258	7.97707
	5	1.61099	7.76188
	10	1.60598	7.97235
	20	1.58938	7.96763
	40	1.56158	7.86627
44 × 48	0	1.57535	8.27713
	5	1.57453	8.27711
	10	1.57437	8.28781
	20	1.57339	8.29849
	40	1.57137	8.31405
	80	1.56760	8.33694
88 × 96	0	1.56433	8.42643
	10	1.56427	8.42615
	20	1.56421	8.42587
	40	1.56409	8.42526

Table 6.11.1. Influence of explicit artificial dissipation parameter *S* on surface pressure and heat flux *q* [kW/m^2]. (From Blottner, 1990, Table 2.)

in which S is a user-defined weight (a smoothing parameter), J is a normalizing Jacobian, Q represents each of the conservation variables, and ξ and ζ are the streamwise and cross-stream general coordinates.

Since both the implicit and explicit artificial dissipations are ordered terms, either or both of these could be evaluated simply as part of an overall grid convergence study. However, it is also of interest to separately evaluate the explicit artificial dissipation *De*, since the effects of this term can be extrapolated to $De \to 0$ for a *fixed* grid as well. For a particular grid, one obtains solutions with two values of S, and extrapolate the solution as follows.

$$Q(S \to 0) = Q_1 + S_1 \times D$$
$$D = -\frac{Q_1 - Q_2}{S_1 - S_2} \tag{6.11.2}$$

Table 6.11.1 from Blottner (1990) shows this extrapolation for three different grids, each a multiple of two, for the pressure and heat transfer rate at one representative point, on the body surface and 90° away from the stagnation point. (The iteration convergence criteria was tight, $L_2 \le 10^{-8}$.)

Note the double convergence involved. For each grid, the extrapolation gives the corresponding solution for $S \to 0$, but the values are significantly different for each grid. For the coarse grid, there is some departure from linearity for the effect of S (see Blottner, 1990, Figure 2). For the finest grid, extrapolation to $S \to 0$ proved to be unnecessary, since the pressure is already converged to 5 significant figures for S = 20 and 10. This is consistent with the fact that the explicit artificial dissipation is also an ordered approximation. See also Kuruvila and Anderson (1985) and Zingg (1992). Salari and Roache (1990) showed that the convergence as artificial dissipation is reduced in Approximate Factorization solutions of a compressible flow code is well behaved even at low Mach numbers (M = 0.1); see also Section 6.28 below.

Blottner noted that "This result indicates that the [explicit] dissipation term introduces a significant error for the coarse grid solution, but the fine grid solution [for pressure] is only slightly in error." However, "it appears that the heat flux requires a finer grid before this type of analysis can be used effectively."

[Note also the dangers of compensating errors in the pressure solution. The pressure solution on the coarse grid with large dissipation (S = 40) is 1.56158, which differs by only 0.2% from the best fine grid solution extrapolated to S = 0, which is 1.56433. (Some intermediate value of about S = 39 would give equality.) Such a spot check, without systematic convergence studies, could be dangerously misleading, especially since the errors for the heat flux do *not* compensate, but add.]

The paper of Blottner (1990) contains many more details, including Richardson Extrapolation for the grid convergence.

6.12 SINGLE AND DUAL POROSITY CONTAMINANT TRANSPORT: SOURCE LOCATION

Grid resolution studies of radionuclide transport in fractured porous media were performed by Salari et al (1995) using a finite volume code, SECO_TRANSPORT. Transport calculations were performed for the grid convergence test using both single and dual-porosity models. (All the SECO codes have the feature of using separate grids to define aquifer properties and to perform calculations; this is highly advantageous for performing grid convergence studies.) The velocity field for the transport calculations was provided by an algebraic relation, removing any additional errors that might influence transport convergence results. The GCI was used to examine the convergence results. The grid convergence results for radionuclide transport calculations in porous media proved to be significantly influenced by the *plausible but improper* modeling of the source in a finite volume grid during the grid refinement. A source term could be modeled as a point source, or can have some physical dimensions. The distinction was shown to have major consequences for grid convergence tests.

6.12.1 Transport Code

The SECO_TRANSPORT code developed by K. Salari (Salari et al, 1992, Roache 1993; Salari and Blaine, 1996) is one of the suite of SECO (**S**andia–**ECO**dynamics) codes that perform groundwater flow, transport, and particle tracking. SECO_TRANSPORT computes multiple component solute transport in (possibly) fractured porous media using single porosity or dual porosity models. It allows for radioactive decay and generation of daughter products.

The dual porosity model is a conceptualization of the geologic system as a matrix of porous material with fractures. When no significant fractures are present, all flow and transport takes place in the rock matrix. When significant fractures are present, flow in the fracture system dominates flow in the matrix system; one may safely neglect *flow* in the matrix, but not *transport* in the matrix. The contaminant disperses from the fracture system through the interface (possibly including a skin resistance due to clay lining) and into the matrix system, where it also decays, reacts chemically, etc. As a pulse of contaminant passes a point in the fracture system, some contaminant concentration will diffuse into the matrix, then be released again into the fracture fluid (usually water) after the advective pulse has passed. One *could* try to model the system of fractures as a single-porosity system, but the behavior is qualitatively altered. (It is a question of scale.) The mathematical behavior of a dual porosity system vs. a single porosity system is distinct enough to be used to interpret field pumping experiments.

The matrix block equation of the dual porosity model can treat the matrix material and a clay lining. For the fracture-matrix system, transport in the fracture is produced by advection and the related hydrodynamic dispersion/diffusion, while transport in the matrix block is assumed to be dominated by

molecular diffusion. Thus, the governing partial differential equations contain advection-dispersion (in the fracture system only), adsorption, source, and decay terms.

These equations are solved using an efficient Approximate Factorization procedure with implicit boundary conditions. The code uses an implicit Total Variation Diminishing (TVD) scheme with three-level time differencing and directional splitting for improved accuracy and execution time. The algorithms are 2nd-order accurate in space and time. In our applications, the flow field for transport was usually obtained from the SECO_FLOW code (Roache, 1993) or equivalent.

The SECO_TRANSPORT code on a 80 × 80 mesh is about 35 times faster than the classical approach (using direct banded solvers) and requires about 6 times less memory. The solution produced is more accurate and less diffusive for low-high Peclet number cases compared to the usual 2-point upstream differencing calculation.

6.12.2 Problem Definition for Source Location

To run the SECO_TRANSPORT code, a flow field is needed and is usually generated by the SECO_FLOW code. In Salari et al (1995), the intention was to conduct the study independent of the flow

i	1	2	3
a_i	–0.046969054	–0.114209345	–0.613161177
b_i	–1.374151646	–6.544653318	–1.327978144

Table 6.12.2.1. Coefficients for Equations 6.12.2.1–2.

code so that the convergence of the flow field is decoupled from the transport. A non-trivial 2-D flow field representative of an active study was constructed using analytical relations as follows. Initially, a coarse grid representation of the desired velocity field was established, followed by crudely fitting the components of this field, u and v, with analytical functions. The velocity fields generated by these functions were examined and the following functions were chosen by R. Blaine to represent the flow field:

$$u = a_1 + a_2 x^2 \ln(x) + a_3 y^2 \ln(y) \tag{6.12.2.1}$$

$$v = \exp(b_1 + b_2 x^2 \ln(x) + b_3 y^3) \tag{6.12.2.2}$$

where $0 \le x \le 1$, $0 \le y \le 1$, and the coefficients are given in Table 6.12.2.1.

For the test problem, coordinates were scaled by 5000 m and velocity components by 2.5E−10 m/s to obtain representative values for physical domain and velocity magnitude. The maximum divergence of the flow field scaled by maximum velocity was about 1.0E−5 1/m which was adequate for the test problems. (The interest was in adequately representing the structure of the representative velocity field, rather than in using a truly divergence-free velocity field.) Since the flow field was defined analytically, the velocity field could easily be constructed on an arbitrary grid. This is an advantage for the grid convergence study.

For the grid convergence test, we used 3 uniform grids ($\Delta x = \Delta y$) of 40 × 40, 80 × 80, and 160 × 160 cells. The Δx's for each grid are 125, 62.5, and 31.25 meters. The *point* source Q is assumed to be located at 1437.5, 3837.5 meters and there are two rectangular discharge boundaries as shown in Fig. 6.12.2.1. The integrated discharges from these boundaries were used in the convergence study because they were the

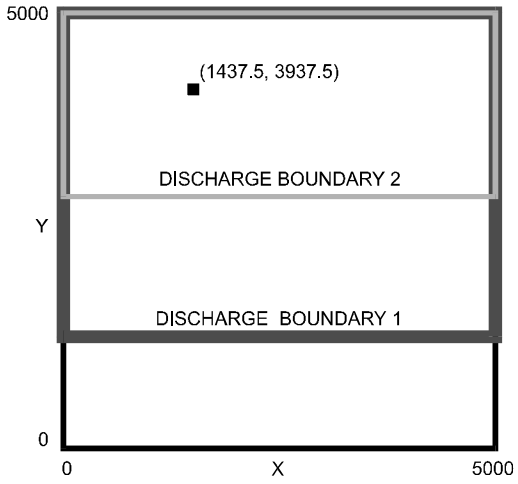


Figure 6.12.2.1. Groundwater Transport Domain Including Location of the Point Source.
(From Figure 1 of Salari et al, 1995.)

performance measures for the radioactive waste disposal site under study. For both model problems, single and dual-porosity, the parameters were as follows.

- matrix porosity = 0.16
- matrix tortuosity = 0.0001
- fracture tortuosity = 1.0
- fracture spacing, aperture = 3.5 m, 0.0035 m
- free water molecular diffusion coefficient = 1.7E−10 m^2/sec.
- longitudinal dispersivity = 100.0 m
- transverse dispersivity = 10.0 m
- transported actinide ^{233}U
- decay constant = 1.39E−13 1/sec

See Salari et al (1992) for details of the continuum equations. The time for the transport is 4500 years starting at 1000 years ending at 5500 years. Three different Δt's were used: 22.5, 11.25, and 5.625 years, which correspond to 1000, 2000, and 4000 time-steps respectively.

The time-dependence of the source function for the transport calculation was constructed using a half sine function over the pulse duration of ~3200 years with maximum amplitude of 1.0E−15 kg/s. The source release began at time 1005 years. The total mass injected (^{233}U) over ~3200 years was 3.1832E−5 kg.

One of the important aspects of setting up this convergence study was to relate the physical location of the source *Q* to the different grids. For a finite difference calculation, the computed properties exist at node points. This allows for a physical location to be preserved through arbitrary *integer* grid refinements. However, this is not the case for a finite volume grid where the computed properties are located at cell centers, and grid doubling will not preserve a physical location. (In order to retain the location of a cell center during grid refinements, the grid size would have to increase by a factor of 3.) Therefore, for grid doubling, a finite volume grid poses the following difficulties.

Assume the source location corresponds to a cell center on the coarse grid, so that the grid doubling splits this cell into four cells. Now, one has to decide which of the four new cells can be associated with the source. It should be apparent that this is an arbitrary choice and, in any case, that the continuum location is now positioned at the *edge* of a cell, rather than at the center. Since the computational representation covers the cell, the effective position is always located at the center of the cell, consistent with a second-order

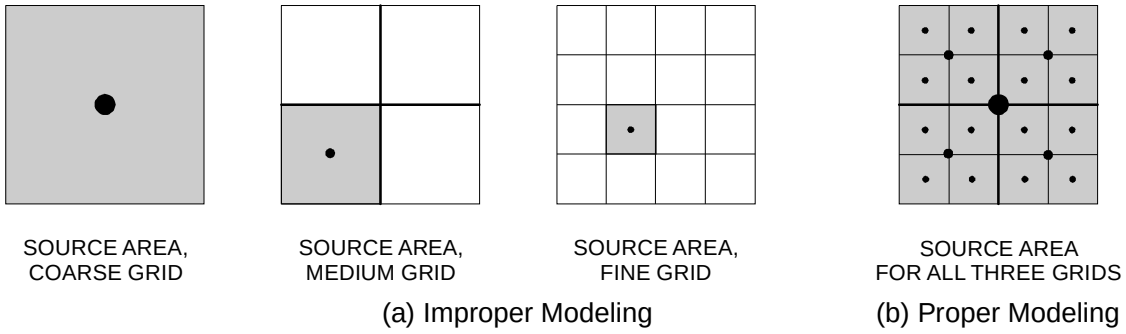


Figure 6.12.2.2. Effective Change in Source Location or Size as Grid is Refined. (a) Improper Modeling of the Point Source. (b) Proper Modeling of the Point Source (From Figure 2 of Salari et al, 1995.)

representation. Thus, the grid doubling with a finite volume representation effectively moves the point source by $\Delta x/2$ and $\Delta y/2$, introducing a first-order error in the location of the point source. Further grid doubling would introduce a similar choice. This process is shown in Figure 6.12.2.2a and is labeled as *improper* modeling of the source.

Intuition might suggest that such a small change in effective location would only affect results in the immediate neighborhood of the source, but would not affect rate of convergence of global results such as integrated discharge. Our results showed that this is not the case.

In order to remedy this shortcoming of the finite volume grid one has to physically preserve the initial cell associated with the source. This means that as the grid is refined, for example from one coarse cell to 4 medium cells and to 16 fine cells, the total injected mass should be equally distributed among the smaller cells. As a result, the area and location of the initial coarse cell are preserved as well as the total mass injected. This is shown schematically in Figure 6.12.2.2b and is labeled the *proper* modeling of the source. As shown below, only the latter approach will produce the correct 2nd-order convergence behavior.

6.12.3 Results on Source Location

The study was conducted for transport calculations using single and dual-porosity models on three different *x-y* spatial grids. Only the *x-y* spatial grid convergence tests are of interest herein, but we also examined time-step convergence, verifying adequate resolution over three different time steps. For brevity and focus, these tests are not reported herein; the results presented here were obtained with the smallest Δt. Likewise, for the dual porosity calculations; the matrix block was discretized with 15 stretched cells into the dual porosity direction. The minimum cell size (which occurs at the fracture-matrix interface) was 0.001 (nondimensional). Since the diffusion rate into the matrix is slow, there was no need to further refine the mesh at the fracture-matrix interface for finer fracture grids. In any case, the primary results of the spatial *x-y* grid convergence results presented herein would hold for any reasonable time and dual-porosity resolution.

Figures 6.12.3.1–2 show the results for the single and dual porosity calculations for all three grids. The ^{233}U concentration contours for single-porosity are shown at time = 4150 years and dual-porosity results and the breakthrough curves for the single and dual porosity calculations are shown at time = 5000 years. In both single and dual porosity calculations there are noticeable differences in the concentration contours going from the coarse to the medium grids; however, the concentration contours remain relatively the same

between the medium and the fine grid solutions. The breakthrough curves for both transport models do not show any visible variation through the grid refinements. Note that MT1 and MT2 are the discharge boundaries 1 and 2 shown in Figure 6.12.2.1.

Tables 6.12.3.1–2 present the convergence results for dual-porosity calculations for discharge surfaces 1 and 2. In this case, the source is modeled properly by preserving the initial source "cell" size and location. The theory indicates that the ratio of successive fine-grid GCIs should equal 4 for a grid doubling with a 2nd-order method in the asymptotic range. As the ratios of GCIs in the Tables show, we attained 2nd-order convergence for the integrated discharge for both surfaces.

Tables 6.12.3.3–4 present the same dual-porosity calculations as shown previously in Tables 6.12.3.1–2, with the exception that the source is now improperly modeled with a single cell on all grids. From the behavior of the ratio of GCIs, it is clear that the convergence is no longer 2nd-order. In fact, examination of the integrated discharge column shows that the convergence is not even monotone.

Tables 6.12.3.5–6 present the results for the single-porosity calculations for both discharge surfaces and the proper modeling of the source. The solutions are 2nd-order accurate as shown by the ratio of the GCIs.

Tables 6.12.3.7–8 present the same single-porosity calculations as shown previously in Tables 6.12.3.5–6, with the exception that the source is now improperly modeled with a single cell on all grids. Again, the results are similar to the dual-porosity calculations in that they are no longer 2nd-order accurate.

The level of numerical accuracy attained in these calculations is far beyond that ordinarily expected of groundwater transport predictions. The high resolution is necessary to unambiguously Verify the principal results of sensitivity to the representation of the source term. If the discharge boundaries were farther from the source, the asymptotic convergence results would have been corrupted by computer round-off error. In fact, it is somewhat surprising (and encouraging) that the asymptotic convergence rates were in such good agreement with theory even in Table 6.12.3.1, in which the variation in integrated discharge occurred only in the 5-th significant figure. However, the results would be more pronounced if the integrated discharge boundary were closer to the source and/or if the grid were more coarse. It will be worthwhile, and involves little effort, to use the proper modeling of the source Q described herein in practical modeling work.

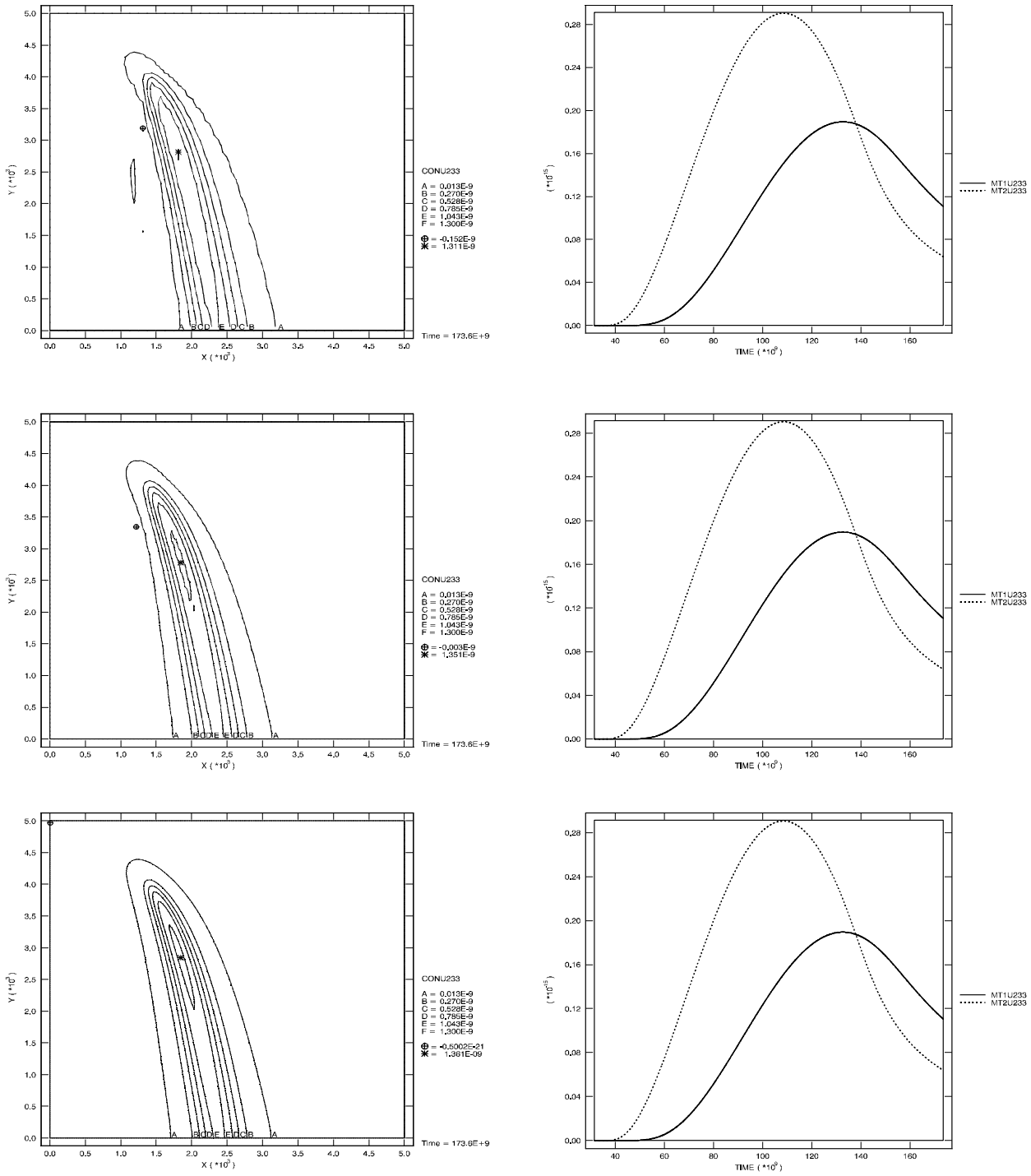


Figure 6.12.3.1. Dual-Porosity Calculations with Proper Source Modeling. Concentration contours and breakthrough curves for all three grids at time = 5000 years. (From Figure 3 of Salari et al.)

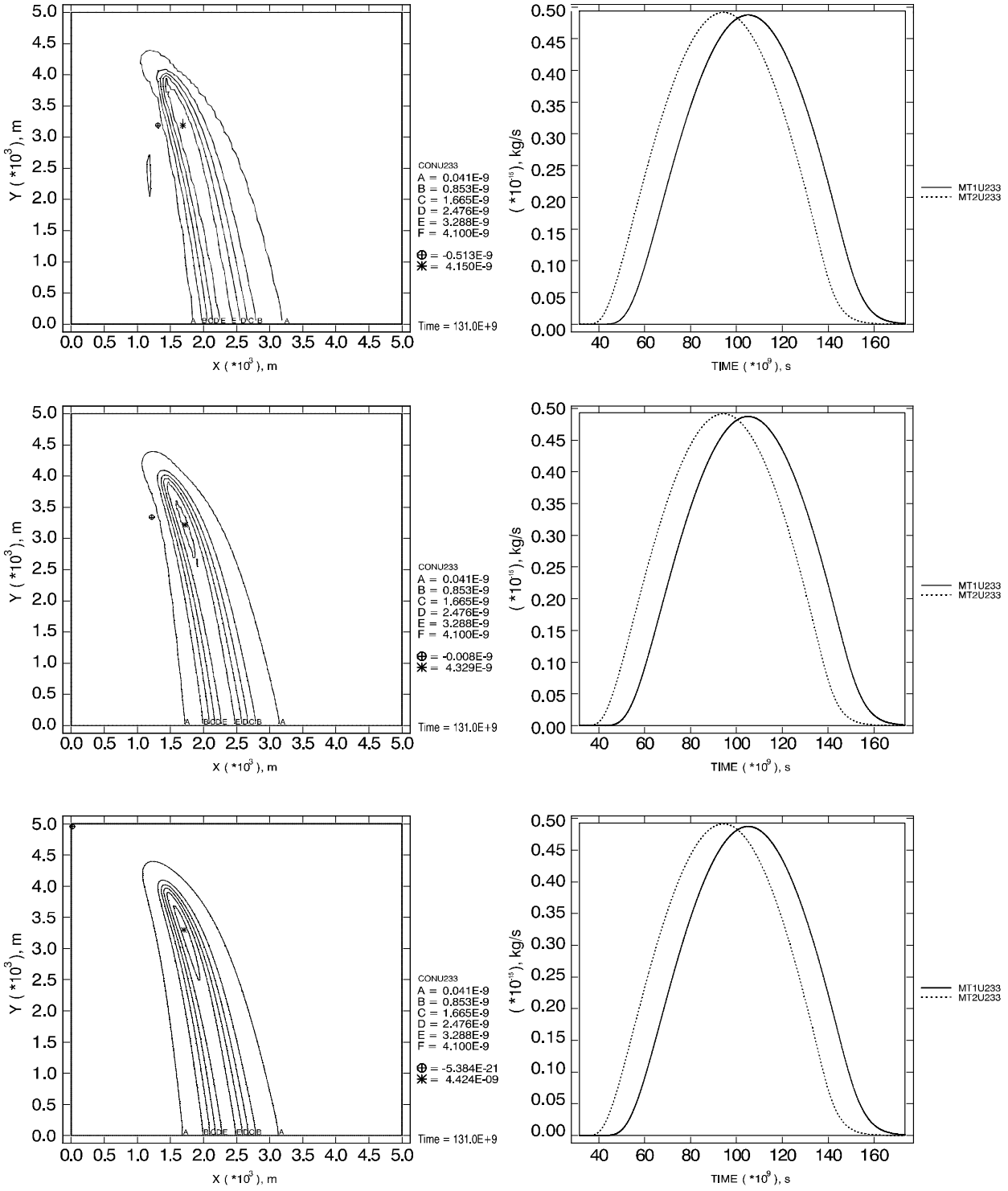


Figure 6.12.3.2. Single-Porosity Calculations with Proper Source Modeling. Concentration contours at time = 4150 years and breakthrough curves at time = 5000 years for all three grids.

Grid	Integrated Discharge	GCI_{fine}	GCI ratio
40 × 40	2.2110467E−5	–	–
80 × 80	2.2115032E−5	0.021 %	–
160 × 160	2.2116186E−5	0.0052 %	3.95

Table 6.12.3.1. Grid convergence results, dual-porosity, *proper* source modeling, surface 1. (From Table 2 of Salari et al, 1995.)

Grid	Integrated Discharge	GCI_{fine}	GCI ratio
40 × 40	1.4098546E−5	–	–
80 × 80	1.4102589E−5	0.028 %	–
160 × 160	1.4103605E−5	0.0072 %	3.98

Table 6.12.3.2. Grid convergence results, dual-porosity, *proper* source modeling, surface 2. (From Table 3 of Salari et al, 1995.)

Grid	Integrated Discharge	GCI_{fine}	GCI ratio
40 × 40	2.2110467E−5	–	–
80 × 80	2.2278868E−5	0.75 %	–
160 × 160	2.2200042E−5	0.36 %	2.08

Table 6.12.3.3. Grid convergence results, dual-porosity, *improper* source modeling, surface 1. (From Table 4 of Salari et al, 1995.)

Grid	Integrated Discharge	GCI_{fine}	GCI ratio
40 × 40	1.4098546E−5	–	–
80 × 80	1.4145352E−5	0.33 %	–
160 × 160	1.4127100E−5	0.13 %	2.54

Table 6.12.3.4. Grid convergence results, dual-porosity, *improper* source modeling, surface 2. (From Table 5 of Salari et al, 1995.)

Grid	Integrated Discharge	GCI_{fine}	GCI ratio
40 × 40	3.0177976E−5	–	–
80 × 80	3.0173064E−5	0.016 %	–
160 × 160	3.0171838E−5	0.0041 %	4.01

Table 6.12.3.5. Grid convergence results, single-porosity, *proper* source modeling, surface 1. (From Table 6 of Salari et al, 1995.)

Grid	Integrated Discharge	GCI_{fine}	GCI ratio
40 × 40	2.7125801E–5	–	–
80 × 80	2.7120203E–5	0.021 %	–
160 × 160	2.7118813E–5	0.0051 %	4.03

Table 6.12.3.6. Grid convergence results, single-porosity, *proper* source modeling, surface 2. (From Table 7 of Salari et al, 1995.)

Grid	Integrated Discharge	GCI_{fine}	GCI ratio
40 × 40	3.0177976E–5	–	–
80 × 80	3.0219098E–5	0.14 %	–
160 × 160	3.0196543E–5	0.075 %	1.81

Table 6.12.3.7. Grid convergence results, single-porosity, *improper* source modeling, surface 1. (From Table 8 of Salari et al, 1995.)

Grid	Integrated Discharge	GCI_{fine}	GCI ratio
40 × 40	2.7125801E–5	–	–
80 × 80	2.7141090E–5	0.056 %	–
160 × 160	2.7131694E–5	0.035 %	1.63

Table 6.12.3.8. Grid convergence results, single-porosity, *improper* source modeling, surface 2. (From Table 9 of Salari et al, 1995.)

6.12.4 Summary on Source Location: 1st-Order Performance with a 2nd-Order Code

This study (Salari et al, 1995) showed that the grid convergence results for radionuclide transport calculations in porous media were significantly influenced by plausible but improper spatial modeling of the source Q in a finite volume grid during the grid refinement. The most obvious approach degrades the experimental order of accuracy attained grid convergence tests to 1st-order. The proposed solution is to preserve the initial area and location of the source. The implementation of this method is quite simple and effective, as has been shown in the examples, and will obviously be applicable to source term representations for problems other than transport in porous media.

Perhaps the point of this study of most general interest is that the *code* itself had previously been *rigorously Verified* to be 2nd-order in space and time. (See the synopsis of the Verification and Confirmation exercises from Salari and Blaine, 1996 given below in Section 6.24.) Indeed, the observed convergence rate in the present example with proper source term modeling *confirms* the 2nd-order accuracy. However, even with a correct code, improper *modeling* (not coding) caused degradation to 1st-order accuracy of the *particular calculation* (rather than the code). By no stretch of imagination could this failure be construed as a fault of the code builder or the code itself; the failure arose in the *use* of the code.

6.13 Δ CONVERGENCE BEHAVIORS FOR MIXED-ORDER METHODS

The previous example from Salari et al (1995) illustrates an important aspect of the convergence behaviors for mixed-order methods. Although the code itself is demonstrably uniformly 2nd-order accurate, the inappropriate modeling of the source term Q caused an observed (an indeed, asymptotic) 1st-order convergence rate. The "method," including the inappropriate modeling for Q, is now a mixed order method, like one obtained by using 1st-order upstream discretization on advection terms with 2nd-order on diffusion terms, or like one obtained using uniformly 2nd-order on interior points and 1st-order discretization of boundary conditions. Since the magnitude of the error is ultimately dominated by the lowest order terms, one may ask what, if any, is the advantage of using higher order accuracy whenever possible, if any 1st-order discretization requires $p = 1$ to be used in the extrapolation and error estimation/banding?

The answer is shown in Figure 6.13.1 for hypothetical examples, qualitatively comparing the convergence behavior of three hypothetical methods. Method A is a uniformly 2nd-order method, as shown by the slope of ~2 in the log/log plot of error vs Δx. Method B is a hybrid method, in which the advection term is a weighted combination of 1st-order and 2nd-order discretization, the (adaptive) weighting depending on the cell Reynolds number. The weighting of the 1st-order discretization goes to zero as the grid is refined, so that ultimately the convergence rate would be 2nd-order; however, over the observed (and usually, the practical) range of Δx, some intermediate rate of convergence $1 \le p_{hybrid} \le 2$ is observed (or more complex behavior; see below). Method C is a 4th-order method except for some locally 1st-order treatment, e.g. the inappropriate Q described above, or an outflow boundary condition, or a small (non-adaptive) weighting of a 1st-order discretization of advection used to "filter" a marginally stable method (e.g., see Roache and Dietrich, 1988).

Over the range shown, hypothetical Method C is more accurate than A, in the sense that its error is smaller. Yet its convergence rate is slower, as evidenced by its slope of ~1. Note that even though C is more accurate than A, the extrapolated solution (and therefore its error estimation/banding) is correctly obtained using $p = 1$ for B and $p = 2$ for A. Likewise, the extrapolated solution for the hybrid Method B is best obtained empirically using the observed rate p_{hybrid}.

The behavior described above for the hybrid Method B is intuitively appealing and often observed in practice, i.e. $1 \le p_{hybrid} \le 2$ is observed. However, Roy (2003) has shown that this smooth behavior can be an unjustified idealization, depending upon the relative signs of competing error terms in the solution. In an elementary yet elegant analysis of model equations, Roy has shown that mixed order differencing of advection and diffusion terms can be a source of oscillatory convergence. (Other sources can contribute, of course, including shocks and other strong nonlinearities.) His analysis based on model equations is shown to be applicable to practical computation of a hypersonic flow.

6.14 GRID CONVERGENCE OF ZERO DRAG COEFFICIENT

When inviscid equations are used to model airfoils in shock-free flow, the drag should be zero. This known exact value can be used to monitor grid convergence, as in Jameson and Martinelli (1996). This variable is not only convenient, but it is deserving of attention because of its engineering importance, as noted by the authors. "In aeronautical applications the accurate prediction of drag is particularly important, and an error as large as 0.0005 is significant since the total drag coefficient of the wing of a transport aircraft (including friction, vortex and shock drag) is in the range of 0.0150." Their calculations using the H-CUSP scheme are zero to 4 significant figures on a 160 × 32 mesh, as shown in the following Table 6.14.1, which also demonstrates the roughly 2nd-order convergence rate.

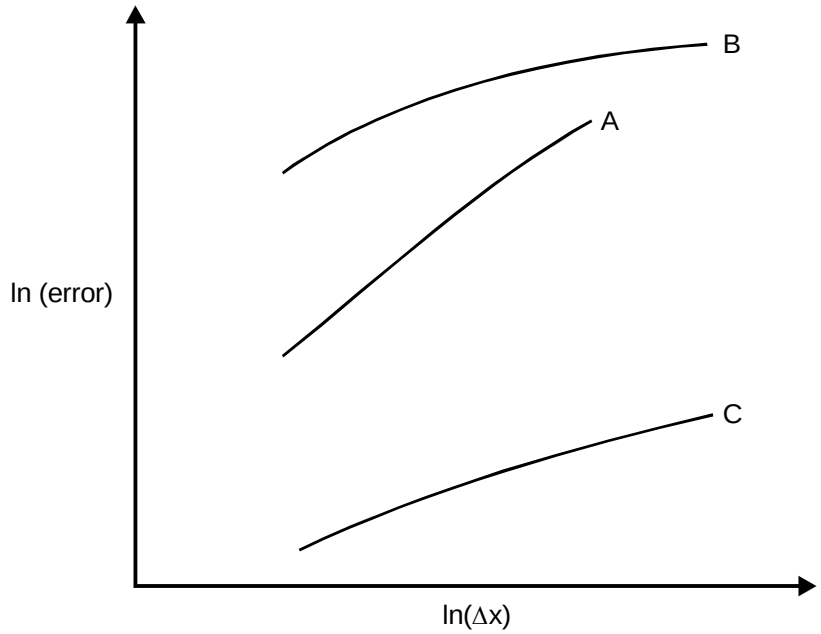


Figure 6.13.1. Qualitative convergence rate and error magnitude for three hypothetical methods. A is a uniformly 2nd-order method. B is a hybrid 1st- to 2nd-order method. C is a 4th-order method for the dominant terms, with a minor term or boundary condition discretized with only 1st-order accuracy.

For non-zero drag coefficient problems, grid convergence estimation requires at least three grid solutions to estimate *p*, as noted previously. Jameson and Martinelli also gave examples of time-dependent calculations for incompressible flow over an oscillating cylinder, obtaining a value of *p* for the maximum (over the oscillation cycle) drag coefficient *MD* (not the original notation) in three grids. In a transparent notation,

$$p_D = \log_2\left(\frac{MD_{65\times33} - MD_{129\times65}}{MD_{129\times165} - MD_{257\times129}}\right) = 1.84 \qquad (6.14.1)$$

"which is close to 2." Better Verification is obtained for incompressible vortex shedding from a stationary half-cylinder at Reynolds Number = 250, for maximum drag coefficient *MD*, maximum lift coefficient *ML*, and Strouhal number *S* (a dimensionless shedding frequency).

Mesh	RAE 2822 Airfoil Mach 0.50, $\alpha = 3°$	NACA 0012 Mach 0.50, $\alpha = 3°$	Korn Airfoil Mach 0.75, $\alpha = 3°$
40×8	0.0062	0.0047	0.0098
80×16	0.0013	0.0008	0.0017
160×32	0.0000	0.0000	0.0000

Table 6.14.1. Drag Coefficient on a Sequence of Meshes using the H-CUSP Scheme.
(From Table 2 of Jameson and Martinelli, 1996.)

$$p_S = \log_2\left(\frac{S_{65\times65} - S_{129\times129}}{S_{129\times129} - S_{257\times257}}\right) = 2.27 \tag{6.14.2}$$

$$p_L = \log_2\left(\frac{ML_{65\times65} - ML_{129\times129}}{ML_{129\times129} - ML_{257\times257}}\right) = 1.83 \tag{6.14.3}$$

$$p_D = \log_2\left(\frac{MD_{65\times65} - MD_{129\times129}}{MD_{129\times129} - MD_{257\times257}}\right) = 2.01 \tag{6.14.4}$$

"which are indeed close to 2." Note that Strouhal frequency is not a difficult quantity to predict physically nor to converge mathematically, because it is essentially an inviscid phenomenon and therefore fairly insensitive to viscous modeling, boundary layer resolutions, etc. (These calculations were all carried out with the outflow boundary located 16 diameters downstream, presumably based on some unreported sensitivity calculations.) The paper (Jameson and Martinelli, 1996) also contains an excellent example (in its Figures 30 and 31) of the radical improvement in convergence for the drag polar produced by graduating from a 2nd-order time-accurate method to a 3rd-order method.

6.15 ANOMALOUS RESULT POSSIBLY DUE TO GRID STRETCHING

Steffen et al (1995) used the GCI to report a grid convergence study on a calculation of incompressible flow in a rectangular nozzle using a well-Verified code with a k-ε turbulence model. They used GRIDGEN, a commercial code grid generation package (Steinbrenner and Chawner, 1992) to generate a 10 block, structured, generalized coordinate mesh with cells clustered towards the nozzle walls and exit plane. The base grid contained 450,000 cells, and grid refinement and coarsening of 10% was used to generate 3 grids, with the motivation of keeping the solution in the asymptotic range (see discussion in Chapter 5). The GCI values were calculated with $r = 1.1$ and $p = 2$ for two functional quantities: (a) the axial location at which the jet centerline velocity has decayed to 1/2 the value at the exit plane, and (b) the axial location at which the jet half-velocity-width along the nozzle minor axis has spread from 0.5 to 1.5.

The results were fairly convincing for functional (a). In their notation, the coarse-to-medium grids GCI(c-m:a) = 10.3%, and the medium-to-fine GCI(m-f:a) = 7.64%. These notations correspond to GCI_{23} and GCI_{12} in Eq. (5.10.5.2),

$$GCI_{23} = r^p \, GCI_{12} \tag{6.15.1}$$

Recall that approximate satisfaction of this equation indicates the asymptotic range. With $r^p = 1.1^2 = 1.21$, the above values give

$$10.3\% \approx 1.21 \times 7.64\% = 9.24\% \tag{6.15.2}$$

Considering the inherent difficulties of grid smoothness across boundaries of 10 grid blocks, this is reasonable satisfaction (~10%), and indicates that all 3 grid calculations are in the asymptotic range for functional (a).

However, the results for functional (b) were anomalous, with GCI(c-m:b) = 3.44% and GCI(m-f:b) = 4.81%, an incorrect trend. The authors noted that these results were "located in a region of the computational mesh where stretching could have affected the accuracy. This deserves further study before a solid conclusion can be reached." It would appear likely that the results are adequately converged for both functionals (a) and (b), as the authors believe, but that the assumption of observed $p = 2$ is not Verified due to grid stretching problems of the type discussed in Chapter 5, including subtleties associated with multi-block grid generation methods. No matter what the final conclusion might prove to be, this sort of candid and painstaking reporting of grid convergence tests builds confidence effectively.

6.16 NON-SMOOTH PROPERTY VARIATION: GLOBAL ERROR NORMS

The general issues of non-smooth property variation were discussed in Chapter 5. Here, we present results of a particular study, and special considerations for global error norms.

Ruge (1995) performed a thorough study of grid convergence and an evaluation of the GCI for 1-D two-phase flow in porous media with non-smooth property variation. The 1-D convergence study was performed to support 2-D studies of a geologic waste repository.[76] The 2-phase Darcy flow code uses mixed 1st- and 2nd-order spatial differencing and adaptive 1st-order time differencing, common for this class of problems. Beginning with a base case of 61 spatial cells, four levels of spatial refinement (by factors of 2, 4, 8 and 16 from the base case) were used, and the effect of timestep refinement (with the same nominal refinement factors) was also examined. In preliminary 2-D studies, Ruge found that the spatial error dominated, so that error estimates could be examined in terms of refinement of spatial step Δ alone; however, for the more extreme refinement possible in the 1-D problem, this was not true. The nominal time-step refinement factors sometimes were overcome by the solution-adaptive time-stepping algorithm, producing higher than nominal refinement. This caused some inconsistency with the temporal order inferred, compared to runs wherein the nominal (scheduled) time step refinements were followed.

This study focused on global rather than local convergence, and Ruge found that the estimates of convergence could depend strongly on the quantities compared and the norms used. A major factor was the cell alignment of faces with the boundaries of the computational domain. As already noted above in Section 6.12, in this grid configuration, doubling the number of cells requires the *location* of the center of a single cell to shift by $\Delta/2$ (See Figure 6.12.2.2.). Combined with non-smooth property variation, this can confuse grid convergence calculations. It was necessary to avoid any kind of interpolation across the coarse grid cell faces, since discontinuities in the problem coefficients can interfere with accuracy and affect error estimates. For this reason, spatial comparisons were made only *within* cells, and comparisons between two levels of discretization were made only on the coarser level, i.e. the finer mesh solution was transferred to the coarser mesh, where the non-smooth properties were defined to be piecewise constant. This allowed

[76] WIPP PA Dept. (1992), Helton et al (1995, 1996)

2nd-order interpolation between fine level cell centers (within a coarse grid cell, so that material properties are constant) to obtain fine-grid values at the coarse level cell centers. Comparisons were further complicated by the fact that the 1-D problem of interest represented a highly stretched quasi-radial mesh, so that uniform subdivision in the coordinate did not produce uniform subdivision of volumes.

Consequently, initial attempts at grid convergence studies clearly showed that it was necessary to carefully define the norms used for measuring solution differences over space and time. With S^k the space of grid functions f^k defined on the level k refinement grid, so $f^k \in S^k$, and v the volume of the ith cell, Ruge used the following volume-weighted discrete approximation to the spatial L_2 norm.

$$\left\| f^k \right\|_{S^k} = \left(\frac{\sum_i v_i (f_i^k)^2}{\sum_i v_i} \right)^{1/2} \tag{6.16.1}$$

The time norm was defined similarly, with the simplification that the solutions being compared were all stored at the same sequence of timesteps, regardless of what sequence they were calculated on (further time step refinements being skipped over) so that the level need not be specified for t. With f being assumed piecewise linear on each time interval, the normalized L_2 norm in time was defined as follows.

$$\left\| f \right\|_t = \left(\frac{1}{t_n - t_0} \sum_{i=1}^{n} \frac{f_i^2 + f_i f_{i-1} + f_{i-1}^2}{3} \right)^{1/2} \tag{6.16.2}$$

The time norm was applied to the space norm to obtain the full norm, as follows.

$$\left\| f \right\|_k = \left\| (\left\| f \right\|_{S^k}) \right\|_t \tag{6.16.3}$$

The norms as defined do not commute, but Ruge found that the order in which the norms were applied to obtain the full norm did not have much effect. The same norm order was used consistently in all the comparisons. Generally, consistency is important to the convergence study. "The norm of the difference between two functions can depend heavily on the level on which it is measured, so that when norms of several differences are involved in a GCI calculation, for example, all functions involved should be weighted to some common level." (Ruge, 1995)

With the dependent variables of interest being pressure and saturation, Ruge found the temporal order $q = 1$ in most cases. The spatial orders of convergence for the global measures of pressure were found to be approximately $p = 2$ at all refinement levels; however, the $p = 2$ result is largely an artifact of the highly stretched quasi-radial mesh, which makes most of the spatial region covered to be in virtually uniform flow. Near the modeled waste repository, where spatial variations were significant, the convergence order was only $p = 1$. For the saturation, $p \cong 2$ on the coarser levels, but decreases to $p \cong 1$ on the finer levels. [This would indicate that the coarser levels of mesh refinement are *not* in the asymptotic range of convergence, contrary to Ruge's statement.] The methods used are mixed order, so that use of the more conservative *a priori* values $p = 1$ and $q = 1$ in the GCI calculation was expected to be more conservative. Indeed, Ruge found that the GCI generally was about 4 times the actual error, compared to the factor $F_s = 3$ in the GCI definition (see Chapter 3). [Note, however, that the actual 2-D calculation of engineering interest involved much more coarse resolutions than Ruge's 1-D study, so the additional conservatism of $F_s = 3$ is probably appropriate.] Thus, Ruge (1995) concluded that "the GCI is performing well in our case."

6.17 Δ DIFFICULTIES OF SPECIAL METHODS

Special methods of discretization requires special consideration for Calculation Verification. Here we mention some aspects of special methods in CFD: discrete vortex method, LES and DNS methods.

6.17.1 Discrete Vortex Methods

Discrete vortex methods in fluid dynamics can present particular difficulties for grid convergence testing because of sensitivities. In discrete vortex methods, there is no grid. Rather, the discretization involves the selection of a finite number of vortices. The velocity field at every point needed is calculated from the Biot-Savart law, summing over all the vortices. The velocity field is calculated at each location of the vortex centers, and the velocity field is then used to move the vortices in a Lagrangian calculation. Implicit algorithms are difficult to implement efficiently, and virtually all the published methods use fully explicit differencing. Stability requires the addition of a viscous core to the inviscid vortex, which has a singularity at the center. Although physical vortices have a viscous core of size ordered by the Stokes radius, the modeled vortices can have much larger cores without destroying the accuracy. In fact, it is necessary for stability that each vortex stay within the viscous core of a neighbor; if vortices move apart so this condition would be violated, the vorticity is locally redistributed between old vortices and new vortices "injected" into the simulation. Hugo (1995) has shown that this condition corresponds to a physically intuitive requirement that the flow field at a neighboring vortex in a one-dimensional simulation is not changed, i.e. if a vortex is injected outside the core, the induced velocity at neighbors changes, evidently leading to instability.

The operation count for a fully viscous flow field (such as the driven cavity problem at low *Re*) is prohibitive. The efficiency of the method comes from the fact that vortex resolution is not required in the inviscid regions. For example, in the calculation of a shear layer, vortices are only required in the shear layer, not in the free stream region. However, the rapid increase in computer time with the number of vortices N makes convergence testing difficult. The time step convergence is straightforward. Because of the requirement for the viscous core, and the (usually) non-physical modeling of the core, it is not clear to what system of equations the method would converge in the limit of $N \to \infty$. Convergence issue are further complicated when the modeling is statistical, e.g. aero-optical phase distortion due to turbulence fluctuations; see Jumper et al (1994), Jumper and Hugo (1995), Hugo (1995).

6.17.2 § LES and DNS Methods

Large Eddy Simulations (LES) and Direct Navier-Stokes (DNS) methods are alternatives to the RANS turbulence models. (E.g. see the review of LES methods given by Madnia et al, 2006.) Although they have higher requirements for resolutions, they have a better chance of providing accurate solutions for more general problems than RANS, which are typically tuned for specific classes of flow problems, often quite limited. LES and DNS pose some challenges to the use of grid convergence for assessing code accuracy, but fundamentally the approach described in Chapter 5 can be followed. In any method, care must be taken to assure grid resolution sufficient to minimally resolve the physics of the problem, e.g. boundary layers and thermal gradients must be adequately resolved, but this is particularly important in LES. As noted in Section 2.4 of V&V20, the LES filter width is usually related to a measure of the grid resolution, and thus as the grid resolution is changed during the grid convergence study, the filter width also is changed. This means that the partitioning of energy between the resolved and unresolved scales is changing. Thus, if users are not careful as the grid convergence study is executed, they may be solving a different problem for some of the coarse-grid resolutions if the boundary between resolved and unresolved scales changes significantly

from grid to grid. The same analysis applies to DNS, in that coarser grid resolutions may not resolve the same set of appropriate flow scales adequately to qualify the simulation as DNS. A DNS simulation by definition resolves all pertinent flow scales (in the frequency domain) up to viscous dissipation.

Several new approaches for calculation uncertainty (generally, "assessment measures") for LES were developed by Celik et al (2006). The new measures that they considered are very specific to the physics of turbulence and involve (a) fraction of total kinetic energy, (b) grid size relative to Kolmogorov and Taylor scales, (c) effective subgrid/numerical viscosity relative to molecular viscosity, (d) power spectra of turbulent kinetic energy. They also took initial steps to address a subtle and important topic, segregation of numerical and modeling errors, which are closely linked in LES unless the LES filter width is defined separately from the mesh size *h*. This latter approach has significant advantages (as noted in Section 3.13); without it, there is no continuum equation that LES solves, other than full Navier-Stokes equations in the limit. So, for example, a Code Verification study using MMS could only use a manufactured solution for the Navier-Stokes equations. While the usual Code Verification process applies in the limit, the process would not have Verified the coding of LES features. Of special interest is the approach of Celik et al (2006) using 3 solutions on 2 grids. The "standard" (unperturbed) LES model is applied on two grids to estimate the purely numerical error using an assumed theoretical rate of numerical convergence. Then a modified (perturbed) LES model is used on the finer grid to estimate model error. The numerical and modeling errors are combined using absolute values to attain some conservatism, in the spirit of an uncertainty estimate. The paper is highly recommended for the presentation of the background as well as the new approaches.

6.18 OBSERVED CONVERGENCE RATES FOR EULER EQUATIONS WITH SHOCKS

This section summarizes results of Engquist and Sjogreen (1996) as presented in Yee and Sweby (1996) on the observed convergence rate for discontinuous (shock) solutions of the Euler equations (for inviscid, compressible flow) using high-resolution shock-capturing schemes such as TVD (Total Variation Diminishing) and ENO (Essentially Non-Oscillatory) schemes. The critical lesson and mathematical distinction is that the Euler equations are a *system* of hyperbolic conservation laws, and the convergence rate behavior unfortunately cannot be inferred from the study of a model *scalar* equation (i.e., for a single variable).

For a shock-containing solution of a scalar nonlinear conservation law, the characteristics point into the shock. TVD, ENO and similar schemes going back to the original FCT, or Flux-Corrected Transport algorithm of Boris (see Book et al, 1975) adapt to a low order scheme just at a shock, since the concept of high-order is meaningless across a discontinuity. According to the linear theory of Kreiss and Lundqvist (1968), dissipative schemes like these damp out the errors propagating backwards against the direction of the characteristics. Thus, it is reasonable to expect that the locally large errors at the shock will stay in the immediate vicinity of the shock. In numerical experiments with formal *p*th order methods , one usually obtains $O(\Delta^p)$ convergence away from the shock.

For systems of equations, complications arise because other families of characteristics exist, and it becomes possible that the large error at the shock can propagate out into the entire post-shock (downstream) region by following a characteristic that emerges from the shock. An example is given for quasi-1-D nozzle flow (using a model system of equations that exhibit shocks; see Yee and Sweby, 1996 for details) for an area distribution given by Eq. (6.18.1).

$$A(x) = 1.398 + 0.347 \tanh(0.8x - 4) \tag{6.18.1}$$

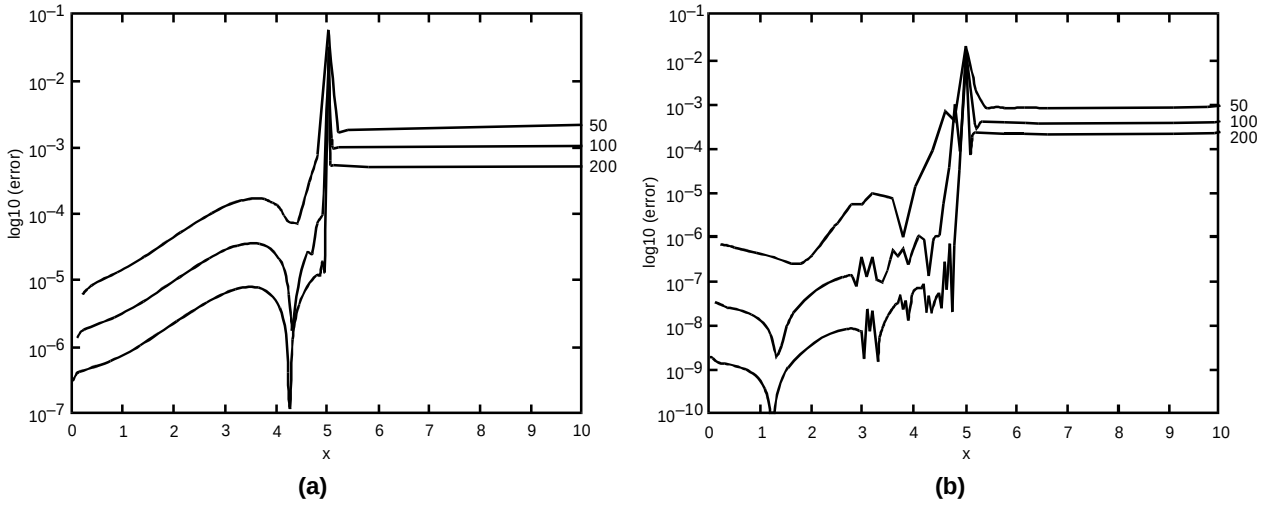


Figure 6.18.1. Effect of Grid Resolution on Error in Momentum for a Quasi-1-D Model System of Equations Exhibiting a Shock. (a) 4th-order ENO method. (b) 2nd-order TVD method. (From Yee and Sweby, 1996, Figure 4.3.)

The exact steady-state solution for the model equations used has a shock at $x = 5$. Figure 6.18.1 shows the error in momentum obtained using a formally 4th-order ENO methods (part a) and a formally 2nd-order TVD method (part b), for three grids containing 50, 100 and 200 points.

The apparently constant jumps in error between the three grid solutions is due to the log scale used; when the observed convergence rate p is calculated, the result for the ENO method (between the 100 and 200 point grids) is 3.9 before the shock, in good agreement with the formal rate of $p = 4$, However, the observed rate is only $p = 1$ downstream of the shock, indicating that the 1st-order error at the shock is being propagated downstream. Likewise for the TVD method, the observed rate upstream of the shock is $p = 2.2$, somewhat better than the formal rate of $p = 2$, but downstream of the shock it deteriorates to $p = 1.1$. Similar results are obtained for 2-D Euler solutions.

In these example calculations, the exact solution of the model problem is known, so 2 grids suffice to determine the observed convergence rate p. For non-model problems wherein the exact solution is not known, it would be necessary to perform at least 3 grid solutions to calculate p, and 4 to verify that the observed p is constant, i.e. that the calculation is indeed within the asymptotic range. Unless such thorough testing is performed, the above calculations on model problems would dictate that error estimation and banding should be done assuming the post-shock $p = 1$.

6.19 COMPLETED RICHARDSON EXTRAPOLATION

The calculations in Roache and Knupp (1993) that Verify the method of Completed Richardson Extrapolation provide a good example of the sensitivity of the grid convergence testing. See details in Chapter 3, Section 3.12.2.

6.20 TRUNCATION ERROR IN ELLIPTIC GRID GENERATION

Methods of generating curvilinear, boundary-fitted grids by solving elliptic equations are highly developed; e.g., see Thompson et al (1985) or Knupp and Steinberg (1993). The seminal paper in this area was that of Thompson, Thames and Mastin (1974) which used the following grid generation equations.

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} + J^2(Px_\xi + Qx_\eta) = 0 \tag{6.20.1a}$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} + J^2(Py_\xi + Qy_\eta) = 0 \tag{6.20.1b}$$

where

$$\alpha = x_\eta^{\,2} + y_\eta^{\,2} \tag{6.20.1c}$$

$$\beta = x_\xi x_\eta + y_\xi y_\eta \tag{6.20.1-D}$$

$$\gamma = x_\xi^{\,2} + y_\xi^{\,2} \tag{6.20.1e}$$

$$J = x_\xi y_\eta - x_\eta y_\xi \tag{6.20.1f}$$

J is the Jacobian of the transformation.

These are PDEs for the physical (x, y) coordinates in the transformed coordinates (ξ, η) and are solved in the (ξ, η) "logical" plane, with easy boundary conditions. (Note that no interpolation of boundary position is required.) An earlier method (for triangle grids) due to Winslow (1967) is recovered when the source terms P and Q are taken to be zero. It was claimed that this *homogeneous* elliptic method of grid generation has the theoretical property that the grid lines should not cross, i.e. the grid transformation generated should automatically be "one-to-one and onto." However, the present author showed by simple hand calculations (exact solutions) on a trivial 3×3 example problem that the method could easily be made to fail (Roache and Steinberg, 1985). For a horseshoe shaped region (geometrically similar inner and outer ellipses, with size ratio of 2), as shown in Figure 6.20.1, the 3×3 grid produces only one interior point. The position of this point falls on the boundary for a critical aspect ratio $A_C = 3\sqrt{2} \cong 2.1$. For aspect ratios smaller than this, the degenerate 1-point grid generation is acceptable. For aspect ratios larger than this, the point falls outside the region, i.e. the grid lines are crossed. Furthermore, crude grid resolution studies indicated that the critical aspect ratio did not diminish very rapidly with increased resolution, as shown in Figure 6.20.2.

It was far from clear whether or not this failure of the homogeneous elliptic grid generation equations indicated an error in the theorems of Thompson et al (1974). The question was later essentially resolved (with some reservations) by Knupp and Luczak (1995) who showed (a) that the failure was sensitive to the boundary parameterization, and (b) that the failure was due to truncation error. That is, in the limit of $\Delta\xi$, $\Delta\eta \rightarrow 0$, a correct grid is generated. (However, the phenomenon remains a problem at finite grid resolution.)

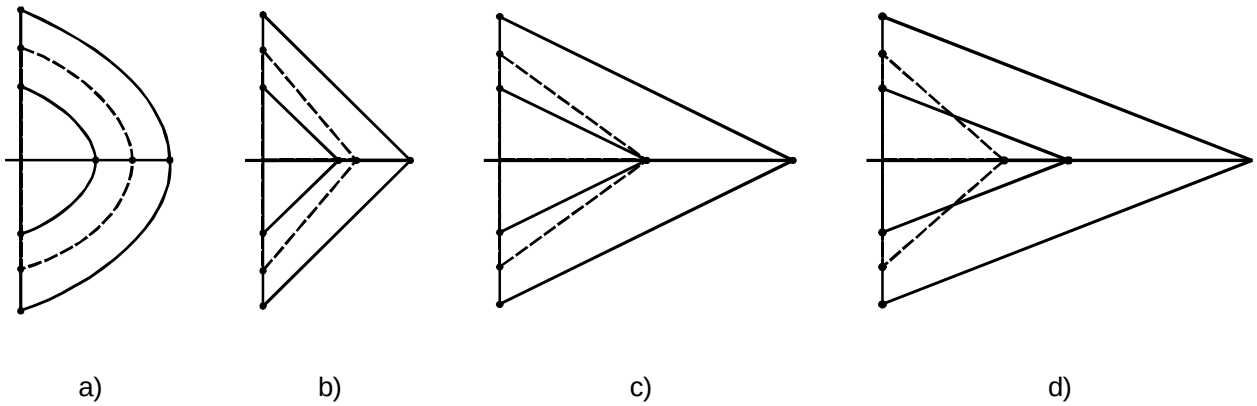


Figure 6.20.1. Horseshoe shaped region for elliptic grid generation, showing the grid folding failure of the homogeneous grid generation equations for the degenerate 3 × 3 grid. *A* is the aspect ratio of the ellipses, and the ratio of outer to inner ellipses = 2. (From Roache and Steinberg, 1985, Figure 2.)

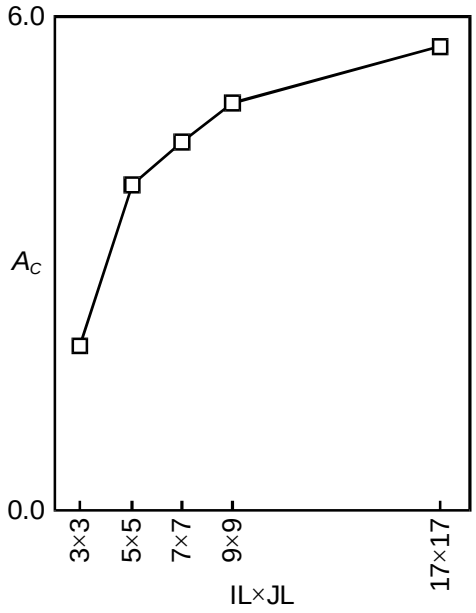


Figure 6.20.2. Variation of the critical aspect ratio A_C with mesh size. The theory would indicate *lim* ($\Delta \rightarrow 0$) $A_C = \infty$. (From Roache and Steinberg, 1985, Figure 3.)

6.21 ONE DIMENSIONAL MOVING ADAPTIVE GRID PROBLEMS

Salari and Steinberg (1994) developed a flux-corrected transport (FCT) algorithm suitable for moving, solution-adaptive grids for difficult 1-D shock propagation problems. They discussed limitations of the algorithm, as well as its significant successes. Clearly, such algorithms are much more efficient than fixed grid methods, yet there are difficulties associated with Verification of Calculations.

6.22 GCI APPLICATION IN SOLUTION-ADAPTIVE NON-INTEGER GRID REFINEMENT

Lotz et al (1995, 1997) present an exemplary application of the GCI in solution-adaptive grids with non-integer grid refinement. The subject problem was the aerodynamics calculation for viscous transonic flow over supercritical airfoils with blunt trailing edges (more specifically, divergent trailing edge or DTE airfoils) designed to create a "rooftop" pressure distribution on the upper lifting surface. The Briley-MacDonald (1977, 1980) Approximate Factorization method was used with explicit artificial dissipation, and turbulent boundary layers were modeled with a mixing-length model; the objective was to quantify the numerical uncertainties (i.e., Verification of the Calculation) rather than Validation of the turbulence model, etc. The GCIs were evaluated for several quantities: three aerodynamic coefficients, shock location, physical extent of the recirculation wake behind the blunt trailing edge, and distributions of the surface pressure coefficients.

Initial grid distributions were established with increased mesh density in the anticipated shock locations, then new grids were adapted to the initial solution. Then the new grids were refined for the grid convergence study. Three grid resolutions were used. The advantages of non-integer grid refinement (see discussion in Chapter 5) are evident; over all three grids, the overall grid refinement ratio is only $r_{13} \sim 1.68$ (compounded of $r_{12} \sim 1.27$ and $r_{23} \sim 1.32$). The relatively constant GCI over both refinements indicates that all three grids are reasonably in the asymptotic range except in the wake. Interestingly, the distribution of *local* GCI for the local pressure coefficients was almost uniform around the airfoil, except in the shock - boundary layer interaction region and at the point where turbulence transition was fixed. [This uniformity suggests that the adapted grid distribution is appropriate, i.e. near-optimum.]

The evaluation by Lotz et al (1997) of the numerical uncertainties indicates good convergence for lift, but only fair convergence for moment and drag, and indicates that drag calculations are more sensitive to numerical error. This is partially in contrast to Salari and Roache (1990), who also found lift to be the easiest to converge, but who found moment errors to be more slowly converging than drag. However, a related result in a similar vein is shown in Table 1 of Lotz et al (1997). Although the value of the GCI is larger for drag than moment, the variation in GCI (from coarse-medium GCI to medium-fine GCI) is larger for moment than drag. This would suggest that the relative error for moment is less than that for drag, even though the asymptotic behavior is not as good; this is perhaps unexpected, but not really contradictory. In any case, both the calculations of Lotz et al (1997) and Salari and Roache (1990) on viscous airfoil grid convergence studies, as well as countless other studies, make it clear that convergence of any functional cannot be assumed *a priori* to guarantee the same quantitative level of convergence for any other functional (as discussed earlier in Chapters 4 and 5; see also Chapter 7).

These studies (Lotz et al, 1995, 1997) clearly show the power and utility of the GCI in providing conservative error bands without the computational expense of successive grid doublings (see discussion in Chapter 5).

6.23 HIGH QUALITY GRID STUDIES LEADING TO A SAFETY FACTOR OF 1.25

6.23.1 Original Studies

Many studies previously cited in V&V1 taken together led to the suggestion made in Chapter 5 that the value of the "factor of safety" used in the GCI [Eqs. (5.6.1), (5.7.2–3)] be adopted as $F_s = 1.25$, but only for such high quality studies with a minimum of 3 grids to experimentally ascertain the observed rate of convergence p.[77] As noted in Chapter 5, Section 5.9, for the more common 2-grid study, we still recommend the value $F_s = 3$ for the sake of uniform reporting and adequate conservatism.

These studies included notably Lotz et al (1995, 1997) and those in the ASME publication edited by Johnson and Hughes (1995), including Chang and Haworth (1995), Salari et al (1995), Pelletier and Ignat (1995), Celik and Karatekin (1995), Leonard and MacVean (1995), and Ruge (1995). The following studies (and others too numerous to mention) published since V&V1 have confirmed this choice.

6.23.2 § Terrassa Group Results

The Heat Transfer Group at the University of Catalonia - Terrassa (Cadafalch et al, 2002) presented a uniquely far-ranging application of the GCI that allowed further evaluation of the conservatism of the GCI in finite volume computations on steady state fluid flow and heat transfer; see also Roache (2003b). The authors treated the following seven problems, any one of which would constitute a study deserving of publication:

- 2-D driven cavity (laminar), with 5 levels of grid refinement
- Variants of the driven cavity problem with 2-D inclined walls, with 5 levels of grid refinement
- 3-D driven cavity (laminar), with 4 levels of grid refinement
- axisymmetric turbulent flow (low Re k-ε) through a compressor valve, *tanh* stretching, zonal refinement, power-law advection differencing, with 5 levels of grid refinement
- 3-D premixed methane / air laminar flat flame on a perforated burner, with 7 levels of grid refinement
- free convection heat transfer from isothermal cylinder in a square duct, three zones, *tanh* stretching of body-fitted grid, with 5 levels of grid refinement
- 2-D linear advection-diffusion model problem, rotated 1-D exact solution, with 6 levels of grid refinement.

Both global and local GCI were calculated. Cartesian staggered and boundary-fitted non-staggered grids were used, with two numerical schemes (upwind differencing and SMART or power-law differencing). As recommended in V&V1 (see Section 5.9.2,3 above) when three or more grid solutions are used to calculate an observed rate of convergence p, the authors used GCI with factor of safety Fs = 1.25. They tested all nodes for monotone or oscillatory convergence (as observed over the grid set used). A global observed p was used in the local GCIs. (Using local p was erratic; Cadafalch (2002).) The authors presented local observed p and local volume-weighted GCI, deviations from global values, and fraction of nodes that were oscillatory vs. (observed) monotone. (Boundary nodes are excluded.) **As many as 1/3 of the nodes were determined to be oscillatory.** The exact solutions being unknown, the benchmark or

[77] We note again that $F_s = 1.25$ should not be used if the results from minimum three grids produce a suspicious observed p. It is imprudent to use observed $p >$ theoretical p in the GCI formula (unless there is some rare good reason to expect superconvergence).

reference solutions were those obtained using the highest order method on the finest grid. As in all such exercises, this makes the GCI evaluation less conclusive for the finest grid sequences, but the evaluations from the other grid sequences should not be significantly affected.

The authors' conclusions included the following. For the linear model problem with exact solution: "The GCI has predicted the real absolute discretization error for all the studied situations quite well." For all problems: "The certainty of the error band estimators has been checked by comparing its value to the "exact" [reference value, highest-order method on finest grid] absolute error of the numerical solutions, always obtaining very reasonable values."

In Roache (2003b) I summarized some conclusions obtained from examination of the results in Cadafalch et al (2002) which shed light on the question of the conservative (or non-conservative) character of the GCI in the engineering sense. That is, how often is the actual error greater than that banded by the GCI? The stated goal has been the 95% certainty (5% uncertainty, what would be roughly a 2σ error band if the distribution were Gaussian), the same error band typical of experimentalists, i.e. ~19 out of 20 cases produce GCI < actual error, over an ensemble of computational PDE problems, including physical problem, grid sequence, and numerical methods. The results were most encouraging for the recommended factor of safety $F_s = 1.25$ for 3 or more grids.

There are various ways to discard outliers or otherwise interpret the results. Basically, I could see only one table entry that was worrisome: Table 5, top line, UDS axial velocity, p =1.2, GCI = 1.8, error = 3.0. Not bad for 176 table entries!

The net result was 14 NC (non-conservative) of 176 entries, or 8.0%. Restricting the count to the SMART algorithm, the net result is 2 NC of 88 entries, or 2.3%. For the UDS algorithm, the net result is 12 NC of 88 entries, or 13.6%. This justifies the claim in Section 5.6 and elsewhere that lower order methods are not only less accurate, but their error estimates (and uncertainty estimates or error bands) are less reliable. Neglecting all cases with $p < 1$, which *might* arguably be considered, we would obtain a net 13/127 for 10.2%. It is seen that discarding $p < 1$ cases discards proportionately more conservative results. This probably is due to the fact that p is increasing as convergence is approached, so that the lag effect makes the estimator and error band more conservative. That is, the fine grid calculation has a p larger than the average (observed) p over the three grids.

In summary, from the limited perspective and interest of determining the conservativeness of using the GCI with $F_s = 1.25$, the results of Cadafalch et al (2002) lead to the following conclusions.

(1) Confirm that the recommended $F_s = 1.25$ used with 3-grid studies to determine the observed p is roughly compatible with the target error band (5% uncertainty).

(2) Confirm that UDS is not only less accurate than higher order methods but is less reliable, i.e. the error estimates and error bands are not as reliable.

(3) Suggest that reliable GCI may be calculated using a global observed p even though as many as ~1/3 of the node values are known to be converging non-monotonically.

(4) Suggest that there is no necessity to discard results with observed $p < 1$, probably because p is increasing as convergence is approached, so that the lag effect makes the estimator and error band more conservative. This leads to excessively conservative GCI for SMART calculations, but this is not an impediment to publication standards. These conclusions from the seven different physical problems computed in Cadafalch et al. (2002) agree with my own experiments on the Burgers equation and with other papers cited earlier, provided that those papers also use careful multi-grid studies with experimental determination of observed p.

6.23.3 § Confined Detonation Problem by SwRI

A difficult practical problem for the application of the GCI is shown in Fig. 6.23.3.1 from V&V20, Section 2.3.3 by Dr. C. J. Freitas of Southwest Research Institute. The highly nonlinear problem consists of an explosive charge (TNT) detonated in a rigid, fluid-filled box. The quantity of interest is the quasi-static pressure at various locations in the box (gray dots in left image) after a finite elapsed time in the time-dependent simulation. The right image in this figure displays the predicted value of pressure as a function of grid resolution at various measurement locations predicted by the set of simulations. In this example, the magnitude of pressure has a smooth dependence on grid resolution. The basis for the grid resolution used is the number of zones across the diameter of the explosive charge.

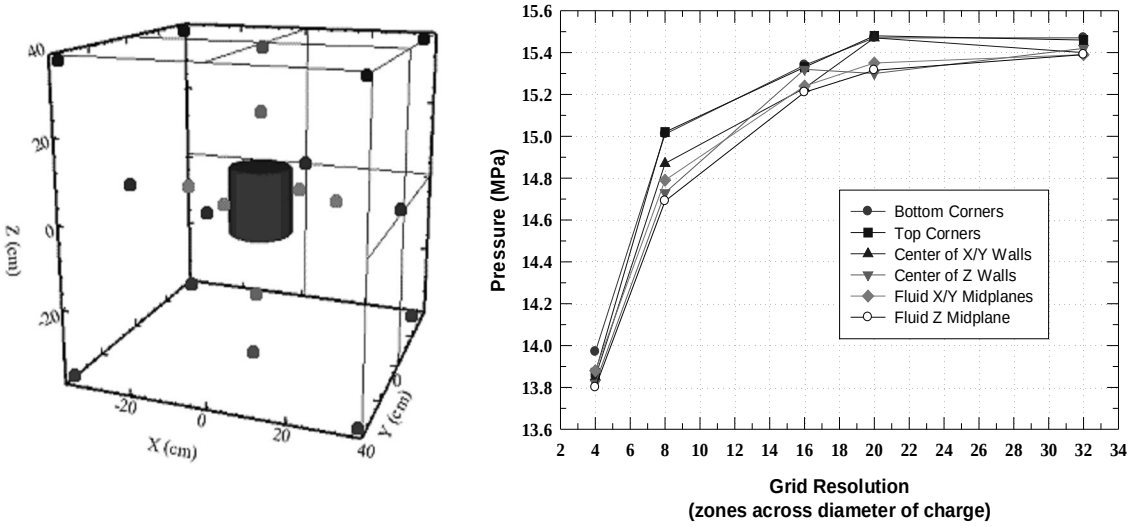


Figure 6.23.3.1 Simulation of Explosive Detonation in a Fluid-Filled Box.
(From Figure 2.1 of V&V20.)

Location of Variable	Corner	Wall	Fluid
Observed order p	1.7	1.5	1.02
GCI value (%)	1.2	1.6	3.6
Value ± U_{num}	15.34 ± 0.18 MPa	15.23 ± 0.24 MPa	15.24 ± 0.55 MPa
Fine grid prediction	15.47 MPa	15.40 MPa	15.39 MPa

Table 6.23.3.1 Simulation of Explosive Detonation in a Fluid-Filled Box.
(Adapted from Table 2.2 of V&V20.)

Table 6.23.3.1 summarizes the results of the application of the GCI to the explosive detonation problem. Pressures at three different locations are used, as indicated in Row 1: a node in the *corner* of the box, a node near the center of a box side *wall*, and a node in the *fluid* at mid-distance between the charge centerline and a box side. Row 2 presents the observed order of the method, calculated from the first four grid resolutions (with 4, 8, 16, and 20 zones across the diameter of the charge). Row 3 presents computed GCIs. Row 4 provides the range in (dimensional) pressure as predicted by the $U_{num} = U_{95\%}$ = GCI. The interval [Value ± U_{num}] is intended to band the exact mathematical solution with a 95% certainty. Row 5 presents the predicted value of pressure on the finest grid (32 zones across the diameter of the charge). The ranges displayed in Row 4 should bound the values in Row 5. They do, again demonstrating both the validity of this approach and the appropriateness of the magnitude of F_s in the GCI method. (V&V20)

6.23.4 § Other GCI Applications at SwRI

In addition to the above example of confined detonation, Dr. C. J. Freitas and his group at the Southwest Research Institute (U.S.A.) have applied the original GCI to many realistic problems with far from ideal convergence behaviors with good results. These include a range of problems including confined detonations (see above Section 6.23.3), vortex cavitation, blast-structure interactions, fluidized beds, shock propagation, structural failure, and turbulent multiphase flows, using FVM, FDM, FEM, ALE, and Riemann solvers. The following is a partial list of GCI applications at SwRI described in company Final Reports (not publicly available). (Freitas, 2009)

1. Simulation of hydrodynamic ram: a study of ballistic projectile impact of fuel filled containers using an Eulerian Hydrocode (CTH).
2. Reactive flow through porous media: a study of dissolution fronts in support of the High-Level Waste Repository using an in-house code.
3. Space Weather: code development and application to the Earth atmosphere from 50 km to 600 km altitude using an (enhanced) NCAR code.
4. Simulation of HTI ordnance in hardened structures: simulated detonation of high-temperature incendiary ordnance in under-ground structures using CTH code.
5. Rotating stall in centrifugal pump diffusers: simulated nonlinear development of rotating stall in vaneless diffusers using an in-house code .
6. Spinning disk microencapsulation: simulation of liquid film breakup dynamics on a spinning disk using an in-house code
7. Explosive hazard assessment for 2nd-generation Space Shuttle: simulation of cryogenic fuel/oxidizer releases and explosive potentials using FLOW-3D code (SwRI-version).
8. Design analysis of Peripheral Vertical Launch System for DD(X): simulation of sympathetic detonation of ordnance in a magazine using CTH .
9. Explosive hazard assessment for Aires/Orion Launch Vehicle: simulation of cryogenic fuel/oxidizer releases and explosive potentials using FLOW-3D (SwRI-version) and solid rocket motor failures using CTH.
10. Modeling of scalar transport: simulation of scalar transport from sources in the Planetary Boundary Layer using FLOW-3D (SwRI-version).
11. Simulation of flight test data for aerial refueling: simulation of aerial refueling between KC-135 and US fleet of receivers using Overflow 2 code.
12. Riemann-based solvers for shock/blast propagation: developed and applied new in-house codes for simulating external and internal blast for inclusion in Navy's ASAP code.

13. Design analysis of MLS magazines: simulation of sympathetic detonation of ordnance in a magazine and structural response using CTH for loading, LS-DYNA for structural response (a Lagrangian FEA code)

Within this broad list of applications of the GCI, several different sub-applications occurred. **"In all applications the GCI provided a very good estimate of uncertainty."** (Freitas, 2009)

6.23.5 § IIHR Compilation and LLNL Study

The compilation at the IIHR Institute at University of Iowa (Xing and Stern, 2009) examined data from a range of analytical solutions, previously published computational solutions, and new solutions for a high resolution problem (up to 8.1 million grid points). The compilation covered 17 studies, 96 variables and 304 grid convergence studies. The claimed results for the GCI were not as good as other studies cited herein, resulting in an overall "confidence interval" of 86.2% vs the target 95%. Two factors are relevant. First, the "confidence interval" reported is a more technical term than other studies (in which cases of adequate conservatism were reported by "coverage", i.e. simply counted), based on statistical analysis of reliability (Students *t*-test, etc.) that depend on assumptions of distributions. Second and more importantly, it appears that the GCI was not correctly applied, i.e. not in accordance with the Summary Recommendations of Section 5.9.2. In fact, when the maximum observed p is limited to the theoretical value[78], consistent with Section 5.9.2, the reported confidence interval is 92.1% (which I consider a success, consistent with an uncertainty target of *roughly* 95% and other studies). If observed p was in poor agreement with the theoretical value, then $F_s = 3$ should have been used, which would further increase the conservatism. See other critiques of this study in Section 6.25.2.

The smaller scale but likewise very thorough grid convergence study at Lawrence Livermore National Laboratory (LLNL) by Logan and Nitta (2006) appears to be similarly flawed in its evaluation of the GCI and minor variants[79] (see Section 6.25.1). Apparently the authors did not use the GCI *method* (as summarized in Section 5.9.2) but just the GCI formula with $F_s = 1.25$, which should be applied only to studies with a minimum of three grids to determine observed p (certainly met in this study, with seven grid levels) but only if the resulting observed p is reasonable. This is not the situation in this study, which involved a small set of problems[80] with "intentional choice of grid studies with oscillations in both exponent p and output quantity", i.e. $(f_g - f_{g+1})$ changes sign over the grid indices g. (In such cases, observed p is not meaningful or calculable without a least-squares approach.) It was on this basis that the authors concluded that "Method #1" [GCI with $F_s = 1.25$ rather than $F_s = 3$] produced closer to a 1-σ uncertainty than 2-σ, although they qualified this conclusion by stating "We realize that there is a large database of CFD solutions (perhaps mostly smooth and monotonic) that supports[81] the use of method #1." This final positive evaluation of GCI was further amplified in their summary recommendations quoted below, with which I concur.

---

[78] Termed the GCI_c method in Xing and Stern (2009) following Logan and Nitta (2006).

[79] Some of these variants consist in just recommended practice of the GCI, e.g. using in the GCI formula p = theoretical p when observed p > theoretical p or is otherwise not trustworthy.

[80] One problem is composed only of synthetic convergence data with no specified PDE. The other consists of simple beam bending with a uniform applied load solved by FDM, FVM and FEM using both shell elements and brick elements.

[81] The misrepresentative evaluation of 1-σ uncertainty was quoted by Xing and Stern (2009) without this qualification. Logan and Nitta referred to the data sets in V&V1 and Cadafalch et al (2002).

"If the analysis is assessed to be low risk, or if one is confident that the conditions for the use of the GCI are valid for the analysis, then the GCI (Method #1) with $N_g = 3$ or even $N_g = 2$ offers a fast assessment of [U_{num}] that may indicate whether more advanced methods [Least-Square] are needed." (Logan and Nitta, 2006)

6.23.6 § Lisbon V&V Workshops

Three Workshops on V&V were held in Lisbon in October 2004, 2006 and 2008. Organized by L. Eça and M. Hoekstra, the first workshop focused on Solution Verification for a realistic turbulent flow over a backstep using several RANS models, with most of the participants (six different international groups) using the GCI or the Least Squares GCI. The organizers provided the grids, including several grids that were of deliberately poor quality, chosen to stress the ability of the GCI uncertainty estimations. The second workshop expanded to include Code Verification using MMS, on both well behaved problems (in the asymptotic regime) and those with erratic convergence, as well as further Solution Verification. Seven different international groups participated. The third workshop expanded to include Validation, using the total Validation Uncertainty approach of V&V20 (Chapter 11) for the well known turbulent backstep flow experiments of Driver and Seegmiller (1985). Seven different international groups participated. Contributors to all three Workshops used a variety of discretization methods, including FDM, FVM and FEM.

The full workshop proceedings can be found in Eça and Hoekstra (2004, 2006b, 2008), with corresponding summary AIAA papers in Eça et al (2005, 2007a, 2009). Briefly, the results from these GCI users confirm again that, in the asymptotic regime, the GCI prediction is compatible with the targeted roughly 95% uncertainty using $F_s = 1.25$, even for deliberately poor grids. For more erratic convergence behavior, the value $F_s = 1.25$ performs fairly well with the Least Squares GCI, and for non-monotonic convergence the recommended value $F_s = 3$ is dependably conservative.

6.23.6 § Common Sense and the GCI Factor of Safety

As Eça and Hoekstra (2002, p.80) observed of the GCI, "Its main difficulty is the choice of the value of the safety factor, which has to be essentially based on common sense." This is also its main advantage, in my view. The authors' evaluation in 2002, that the GCI "seems to be viable and robust," has been confirmed further by the studies in this Section 6.23 and others. Their invocation to common sense is also applicable to performance measures, i.e. targeting *roughly* 95% coverage.

6.24 TRANSPORT CODE VERIFICATIONS USING THE GCI: PARTITIONING THE OPTION MATRIX

The SECO_TRANSPORT code developed by K. Salari (Salari et al, 1992, Roache 1993; Salari and Blaine, 1996) has been previously described in Section 6.12.1. The complexity of the options of the code create the difficulty described in Chapter 2, Section 2.18, which for simplicity is repeated here, with emphasis added.

Another concern in Code Verification is the number of user options in a code, especially general-purpose commercial CFD codes. This is a genuine practical problem, but does not nullify claims of Verification; it just limits those claims. The exponentially expanding complexity of the option tree does not nullify the definition of Verification of Code; it simply qualifies the definition. "Code Verification" is restricted to that combination of options claimed to be Verified. There is a gray area here, as one might expect, in the judgment of the independence of options. Some knowledge of algorithm and code structure

may be necessary to infer the reasonableness *of simplifications of option interactions (essentially, partitioning of the full matrix of option interactions)*. The easy answer is the rigid one of categorical *no*. This would usually appeal to the rigid SJ personality type (see Chapter 12) that gives Quality Assurance a bad name. A more intelligent and economical approach is possible, bearing in mind that subtle and unanticipated option interactions have occurred, especially before the acceptance of structured coding and modularity.

Here we address the "simplifications of option interactions" which essentially constitutes a partitioning of the full matrix of option interactions. We will rely on a knowledge of algorithm and code structure to guide the partitioning, but then "cover our bets" after the Verification by further Confirmation exercises to further build confidence that the partitioning is correct.

The partitioning is applied to the following aspects of the code.

- Verification of the Geologic Matrix Equations for Diffusion, Single Species
- Verification of the Fracture Equations for Advection, Dispersion, Diffusion at low Peclet Numbers, single Species
- Verification of the Coupling between Matrix and Fracture Equations
- Verification of the Radioactive Decay Equations (which are the same in matrix and fracture systems) for Multiple Species
- Verification of the TVD Algorithms (6 options) for Fracture Equations at high Peclet Numbers, Single Species
- Confirmation Exercises on a Commonly Used Analytic Solution, single Species

Code features to be Verified must include

- Time-Dependent and Spatially-Variable Velocity and Dispersivity Field
- Both Constant and Time-Dependent Boundary Conditions
- Time and Space Order of Convergence
- Discharge calculations on pre-defined closed boundaries, involving quadratures.

For many PDE codes, time and space convergence can be partitioned (e.g., the SECO_FLOW codes as discussed in Section 6.2). With the Approximate Factorization algorithm used, spatial convergence for steady state solutions can be established without considering the time differencing options (in theory), but the time accuracy is coupled through the mathematics of the Approximate Factorization, so this major option partitioning is not available for this code. The Verification was made possible by the development by P. M. Knupp of two benchmark analytical solutions for the fracture equations (see Appendix in Salari and Blaine, 1996). The first is a solution for a spatially variable dispersivity field with a time-dependent and spatially variable velocity field, but with time-constant boundary conditions. The second is a solution for time-dependent boundary conditions, a test necessitated again by the subtlety of the AF algorithm. A separate analytical solution was used for the geologic matrix equations, with spatial property variation and time-dependent boundary conditions. The time-dependent solutions were Verified separately for two 2nd-order time differencing options: 3-point backward and trapezoidal time differencing.

With the fracture and matrix equations separately Verified, the implicit coupling of the two was Verified with the previously developed solution by Tang et al (1981) for constant properties. The justification of this partitioning (i.e., not testing the coupling with spatially variable properties) depends on knowledge of the algorithm and code structure. Then the multiple species transport was Verified against the analytical solution of Lester et al (1981). No spatial variability enters into these equations. The solution is

for 3 species in a chain. The partitioning assumption here involves knowledge that the code is written for N species, and has been exercised for $N = 9$; it hardly seems necessary to test all possible numbers of species.

The only aspect of discharge calculations on pre-defined closed boundaries to be Verified involves quadratures. Knowing that these do not depend on other options of the code allows partitioning of this feature. However, since time-and space-dependent velocity fields are involved, the Verification uses the first benchmark problem. We believe this to be a high standard.

The Verification is performed with both uniform and smoothly stretched grids. As is well known, extreme grid stretching will affect accuracy of an individual calculation, but the code is Verified with moderate stretching, knowing that there is nothing special about the stretching function. (One cannot Verify a code for an infinity of stretching functions, nor an infinity of velocity fields, property variations, etc.)

Likewise, the 6 different TVD flux limited options (for eliminating oscillations in shock-like high Peclet number solutions) are partitioned from the rest of the code, since the coding does not interact with dispersion, matrix equations, radioactive decay, etc.

Needless to say, only a sampling of the Verification exercises can be presented here, in Table 6.24.1 and Figure 6.24.1. As noted in Chapter 3, such exercises require that the scientist-engineer "think like an accountant" for a while.

A later version of the SECO_TRANSPORT code included the 3-D *moving* free surface boundary conditions (developed for the single porosity case only). It was difficult to achieve 2nd-order accuracy in space and time with a moving free surface and a general non-orthogonal moving coordinate system, but it was convincingly Verified using two Manufactured Solutions in Roache et al (1996).

After these and many other Verifications of the partitioned features of the SECO_TRANSPORT code, the authors justifiably claimed Code Verification. This Verification was followed by further Confirmation exercises.

Although not logically necessary (if one really believes the Verification) Confirmation exercises can be useful for further confidence building in the user/customer. However, it is *highly* recommended that the code builder (or whoever may be Verifying the code) stand firm on distinguishing between Verification, which at least in principle is accomplished once and for all (like a theorem), and Confirmation, which can go on forever, depending on the whims of users who may be mathematically semi-literate.

In the case of the SECO_TRANSPORT code, the users wanted comparison with the Sudicky-Frind (1982) analytical solution. The problems with this solution were discussed previously (in Section 3.10.1), where it was noted that the infinite series solution has more mathematical uncertainty associated with it than does the Code being Verified. It is interesting (and from our experience, widespread) that users (and regulators, journal referees, managers, etc.) are *more impressed* with such agreements with widely accepted benchmarks in their particular discipline than they are with manufactured analytical solutions. The only real virtues seem to be familiarity and the illusion of confidence. For example, the widely accepted Sudicky-Frind solution, even when corrected and carefully evaluated, does not exercise most of the features of the code! It is based on a constant property medium and steady-state spatially-constant velocity field, yet many prefer it as a benchmark to a manufactured solution with spatially variable properties and time-and-space varying velocity field. It is often prudent to humor such irrational behavior, but only by allowing for Confirmation exercises, not by corrupting the concept of rigorous Verification. Also, managers must be warned that never-ending processes require never-ending budgets and deadline extensions.

Likewise, it is a good idea to include in Confirmation exercises some calculations on a realistic-looking problem of interest to the user. Again, this is an open-ended process. For the SECO_TRANSPORT code, this involved calculation of sample cases from the WIPP project (WIPP PA Dept., 1992; Helton et al, 1995, 1996.). Also, additional exercises of the TVD algorithm on high Peclet number cases were given (Salari and Blaine, 1996).

Grid	Δt	L_2 error	Max. error	GCI	GCI ratio
10 × 10	0.04	3.3742E−3	6.8663E−3		
20 × 20	0.02	8.2240E−4	1.7046E−3	205.360%	
40 × 40	0.01	2.0385E−4	4.2610E−4	49.779%	4.13
80 × 80	0.005	5.0512E−5	1.0590E−4	12.340%	4.03
160 × 160	0.0025	1.2426E−5	2.6112E−5	3.065%	4.03

a. Single porosity fracture equations, time-dependent flow, trapezoidal time differencing, Van Leer's MUSCL limiter, uniform grid. The solutions are 2nd-order accurate, as shown by the GCI ratio ≅ 4. (From Salari and Blaine, 1996, Table 11.)

Grid	Δx	Δt	L_2 error	GCI	GCI ratio
20 × 20	0.05	0.25	7.697E−3		
40 × 40	0.025	0.125	1.954E−4	46.540%	
80 × 80	0.0125	0.0625	4.921E−4	11.847%	3.92
160 × 160	0.00625	0.03125	1.234E−4	2.988%	3.96

b. Single porosity fracture equations, time-dependent flow, 3-point backward time differencing, central differences in space (no limiter), uniform grid. The solutions are 2nd-order accurate, as shown by the GCI ratio ≅ 4. (From Salari and Blaine, 1996, Table 14.)

Grid	Δs	L_2 error	Max. error	GCI	GCI ratio
20 × 20	0.04	4.7240E−3	1.1464E−2		
40 × 40	0.02	1.0251E−3	2.3426E−3	244.200%	
80 × 80	0.01	2.4657E−4	5.6313E−4	51.398%	4.75
160 × 160	0.005	6.0886E−5	1.3810E−4	12.259%	4.19
320 × 320	0.0025	1.5147E−5	3.4211E−5	3.020%	4.06

c. Matrix equations only, steady-state calculations, non-uniform grid (s is the spatial increment in logical space). The solutions are 2nd-order accurate, as shown by the GCI ratio ≅ 4. (From Salari and Blaine, 1996, Table 16.)

Grid	Δs	L_2 error	Max. error	GCI	GCI ratio
20 × 20	0.04	4.7240E−3	1.1464E−2		
40 × 40	0.02	1.0251E−4	2.3426E−3	244.200%	
80 × 80	0.01	2.4657E−4	5.6313E−4	51.398%	4.75
160 × 160	0.005	6.0886E−5	1.3810E−4	12.259%	4.19
320 × 320	0.0025	1.5147E−5	3.4211E−5	3.020%	4.06

d. Discharge calculations. Single porosity fracture equations, steady- state flow, central space differencing (no limiter), non- uniform grid. The solutions are 2nd-order accurate, as shown by the GCI ratio ≅ 4. From Salari and Blaine (1996), Table 16.

Tables 6.24.1. Sampled Verification tests of the SECO_TRANSPORT 2D code using Manufactured Solutions. See Salari and Blaine (1996) for parameters and details.)

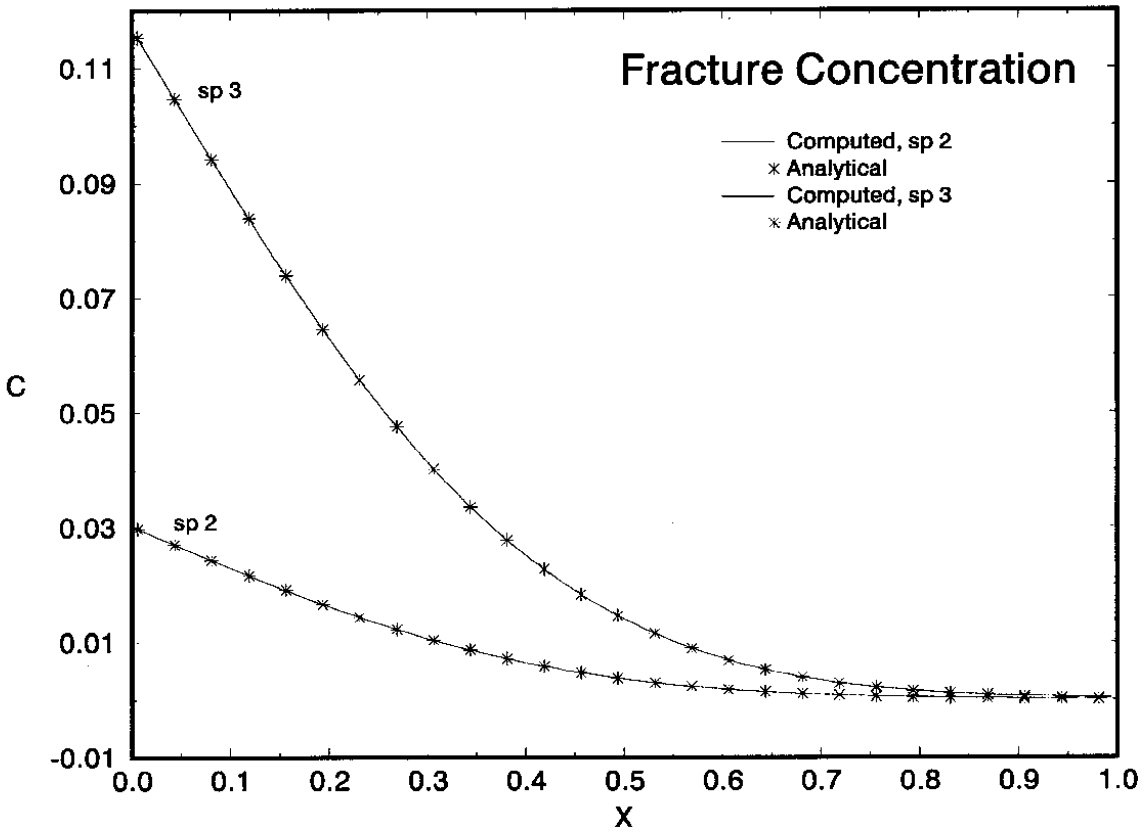


Figure 6.24.1. Sampled Verification test of the SECO_TRANSPORT 2D code using Manufactured Solutions: Multiple Species Verification. Computed concentration *C* in the fracture system, for species 2 and 3 of a 3-species chain, compared with an analytical solution. From Salari and Blaine (1996), Figure 6. See Salari and Blaine (1996) for parameters and details.

In closing, this Section, it is best to bear in mind that even with a rigorously Verified code, user errors will occur, and that *the code will be blamed*! This appears to be a fixed feature of human nature, and requires the code builder (or Sponsor, in the QA terminology of Chapter 12) to continually provide education and training to the users.

6.25 Δ OTHER UNCERTAINTY ESTIMATORS BASED ON RICHARDSON EXTRAPOLATION

Other approaches to estimating numerical uncertainty beginning with Richardson Extrapolation have been suggested.

6.25.1 Δ Celik and Karatekin Method for Turbulent Separated Flow

Celik and Karatekin (1995) applied a generalized Richardson Extrapolation to the calculation of turbulent flow past a backward facing step. The commercial code used (Phoenics), like many, utilizes "hybrid" first-order/second-order differencing for advection terms. This method is robust, but not very accurate, as noted in Chapter 1. See also Leonard and Drummond (1995) for warnings on the use of hybrid differencing. Celik and Karatekin used the standard k-ε two-equation turbulence model with wall functions for the turbulent inner layer, and non-uniform (power law) grid distributions. The hybrid method *ultimately* should be second order, but over practical ranges of grid sizes, it will appear to have some intermediate non-integer convergence rate with $1 < p < 2$. In spite of this ambiguity, by careful work Celik and Karatekin were able to convincingly demonstrate grid convergence, even with the grid refinement extending into the viscous sublayer, by using 5 grid refinements (with equal grid refinement factors in each coordinate direction), i.e. a total of 6 grids.

They used both the GCI and their alternative error estimator (discussed below). As anticipated (Roache, 1994) the GCI was excessively conservative with the safety factor $F_s = 3$ for this 6-grid sequence. (The same applies to other carefully performed grid convergence studies presented in Johnson and Hughes (1995); see Section 6.23 above.) Recall that (as acknowledged by Celik and Karatekin) the GCI was not proposed as an error *estimator* at all, but as an error *band*, equal to F_s times the error estimator, with $F_s > 1$ used to provide conservatism.

Celik and Karatekin (1995) also proposed an alternate error estimator with a built-in conservatism. Although similar to the GCI, and similarly based on a generalized Richardson Extrapolation, there are several differences. First, a relatively minor point which has caused confusion: whereas the ε in the GCI approach (absolute or relative) is based on the difference between the fine and coarse grid solutions (and is normalized by the fine grid solution for the relative GCI), Celik and Karatekin defined their relative error e_{ext} by reference to the extrapolated solution f_{ext}.

$$e_{ext} = \frac{f_{ext} - f_h}{f_{ext}} \tag{6.25.1}$$

This would be equivalent to the ε defined in Eq. (5.4.3a) (Chapter 5) but with f_1 (the fine grid solution) replaced by f_{ext}. Clearly, these two will be ordered approximations of one another; however, any noisy convergence behavior will make f_{ext} undependable and this normalization is not recommended. (It is not an essential feature of the presently considered method.) Also, the authors use the equivalent of the safety factor $F_s = 1$, which by itself would not be conservative.

A somewhat confusing point is that the paper involves two distinct Richardson-type Extrapolations: one used above to define the error estimate e_{ext}, and another used to obtain the best estimate of the converged solution. (Note the distinctions made in Chapter 5, that we may not want to actually use Richardson Extrapolation to obtain a final solution, for various reasons, but we can still estimate or band the error of the fine grid solution using the theory of Richardson Extrapolation.) In the above error estimator, e_{ext} is obtained using the value $p = 1$, which is (generally) conservative for the hybrid algorithm. However, the final solution (or best estimate of the converged solution) was obtained by using a value of p

suggested by numerical experiments, but limited. Their observed rate (for various quantities, over many grid combinations) was higher than $p = 1$, as expected; a representative value was observed $p = 1.3$. However, the point-by-point values of velocity components sometimes (locally) demonstrated non-monotone convergence behavior (incorporated into the Richardson Extrapolation by judicious use of the absolute value function); sometimes the observed p was < 1, and sometimes observed p was > 2. These excursions in observed p, if used in the Richardson Extrapolation, produced clearly erroneous values. The cure proposed was to limit the allowable p used in the extrapolation to the theoretical range of the hybrid method, which is $1 \le p \le 2$.

While this would seem to be a reasonable engineering-type approach, its limitations are clear. First of all, it is clearly non-conservative, since the actual convergence rate could be less (and was). If the same rationale were applied to anything but a hybrid or other mixed-order method, the range $1 \le p \le 2$ would collapse to a single number, e.g. $p = 2$ for a uniformly 2nd-order method. The limit then would simply ignore the experimental determination of observed p and substitute for it the value obtained from formal analysis, as one would need to do if only two grid solutions were calculated. Thus, the limit would replace experimental *determination* or *Verification* of p with a *presumption* of p for a uniformly $p = 2$ method and would defeat the purpose of using more than two grid solutions in a high quality convergence study.

Another possible confusion exists in the definition of the grid refinement factor. Celik and Karatekin use the ratio $a(i)$ which is defined relative to (usually) the finest grid in the sequence. For a simple two-grid sequence, this gives $a = r$, as used in the GCI definitions in Chapter 5. For more grids in the sequence of a grid convergence test, we have

$$r_i = \frac{a_{i+1}}{a_i} \tag{6.25.2}$$

Although not incorrect, this definition of a grid refinement factor is inconsistent with previous and common usage. Clearly, if a grid is refined from 100 to 200 grids points or elements, the refinement factor "r" between these two grids should be 2, regardless of any *other* grids in the sequence of the grid refinement study. But in their definition, the grid refinement factor a, besides being an inverse (i.e., a coarsening factor), is normalized at the finest grid in the study sequence. If the finest grid were (say) 100 in the sequence 25–50–100, then their refinement factor a for the 25 point grid would be 4, not 2. If they added another fine grid of 200 points to the study sequence, then their grid refinement factor a for the same 25 point grid would change to 8. This is an inappropriate terminology, even if it did not disagree with previous common usage. A less confusing term would be "grid index" for a.

The Celik and Keratekin error estimator e_{ext} can be expressed in terms of the closely related fine-grid GCI for $p = 1$ and $F_s = 1$, which we can denote by $\mathrm{GCI}_{1,1}$. That is, using Eq. (5.6.1) with $p = 1$ and $F_s = 1$, we have

$$\mathrm{GCI}_{1,1} = \frac{\varepsilon}{r-1} \tag{6.25.3}$$

Then (from Appendix A of Celik and Keratekin, 1995),

$$e_{ext} = \left(\frac{r-1}{r - f_2 / f_1} \right) GCI_{1,1} \tag{6.25.4}$$

In the limit of a converged solution, $f_2 \to f_1$ and $e_{ext} \to \mathrm{GCI}_{1,1}$ as expected.

For the example shown in Figure 7 of Celik and Keratekin (1995) their error estimator e_{ext} is less conservative (or less *overly* conservative) than the GCI calculated with either $p = 2$ or $p = 1$. However, as seen above, the correct comparison should be to $GCI_{1,1}$, to which e_{ext} is an ordered approximation. Unfortunately, e_{ext} will not be *reliably* less overly conservative than $GCI_{1,1}$ as evident from Eq. (6.25.4) above. If the convergence for the quantity f is approached from above in the grid convergence sequence, i.e. if $f_2 / f_1 < 1$, then $e_{ext} < GCI_{1,1}$ but if $f_2 / f_1 > 1$, then $e_{ext} > GCI_{1,1}$. Thus, in Figure 4 of Celik and Keratekin (1995) for velocity components above the base, e_{ext} is less conservative, but for velocity components below the base, e_{ext} is more conservative.

Although limited in the application in Celik and Keratekin (1995) to the hybrid method, the basic concept of their approach could be generalized to higher order methods. That is, instead of obtaining conservatism in the GCI (or closely related e_{ext}) by using $F_s > 1$, we could use $F_s = 1$ but $p <$ observed p. For example, one could use $F_s = 1$ and $p = 3$ in Eq. (5.6.1) for a verified 4th-order accurate method. Whether this conservatism is any easier to interpret than using $F_s > 1$ is problematical, but could be worth investigation. However, the neat interpretation of the error band in relation to the ε obtained for a grid doubling with a 2nd-order method would be lost. Also, there is no empirical correlation available to indicate what degree of conservatism is achieved, i.e. the uncertainty level is not quantified. Yet is a true ordered error estimator asymptotically, and is conservative for well behaved problems.

Related minor variants of the GCI were considered by Logan an Nitta (2006), such as using $p = 1$ in the GCI formula even though the method is expected to be 2nd order, and averaging the results of $p = 1$ and $p = 2$ formulas. Other candidate are obvious, e.g. using $p = 1.5$, $p = \sqrt{2}$, etc. with or without F_s. The need is not for alternative old-fashioned engineering tweaks, but large scale studies.

6.25.2 § ITTC Correction Factor Method

The International Towing Tank Conference (ITTC) is an association of worldwide organizations, with history beginning in 1932, that have responsibility for the prediction of hydrodynamic performance of ships and marine installations based on the results of physical and numerical modeling. The ITTC Quality Manual (ITTC, 2002) for CFD Uncertainty treats both Calculation Verification and Validation. The ITTC Validation approach is based on the early seminal work of Coleman and Stern (1997). This valuable work has now been superseded by the V&V20 approach described in Chapter 11.

The Calculation Verification described in the ITTC Manual covers several methods and variants. Possibilities include the GCI method itself, but most of the description is given to the Correction Factor methods in multiple variants, developed and tested by F. Stern and associates over years[82], most recently in Xing and Stern (2009).

All variants use generalized Richardson Extrapolation and are algebraically equivalent to the GCI method with a variable factor of safety F_s. (This equivalence is somewhat obscured in earlier publications.) The assumptions are that (a) F_s should be larger for grids farther from the asymptotic region, and (b) F_s should $\rightarrow 1$ as $\Delta \rightarrow 0$. Assumption (a) is certainly true in a general sense, as reflected in the recommended $F_s = 3$ of the GCI when it is not known if the grids are in the asymptotic range (see Section 5.9.2), but it is not clearly necessary to have a continuous variation in F_s. Assumption (b) may appear reasonable but is not conclusively demonstrated to remain conservative enough, i.e. to always provide 95% coverage, although this is probably not of practical importance for highly accurate solutions.

[82] F. Stern was the principal architect of the 2002 ITTC Manual. Other variants and studies that precede or follow ITTC (2002) are Stern et al (1999, 2004), Wilson et al (2001, 2004), Wilson and Stern (2002), Carrica et al (2007), Xing et al (2008), Xing and Stern (2008, 2009). For related discussion see Roache (2003a,c) and Wilson et al (2003).

The variants involve ways to adjust F_s based on how "far" a grid set for a problem is from the asymptotic region, as measured in some metric. The most obvious and fundamental metric would be the comparison of observed p_{obs} with theoretical p_{th}. This immediately suggests the first weakness of the approach, since p_{th} is not always obvious nor trustworthy (see Section 8.1.4). The metric used is not p_{obs} / p_{th} or (p_{obs} - p_{th}) but rather the correction factor CF,

$$CF = \frac{r^{p_{obs}} - 1}{r^{p_{th}} - 1} \tag{6.25.2.1}$$

(or another based on a power series expansion solved for two leading terms). As noted in the ITTC Manual, this formulation fails for the best behaved asymptotic solutions, giving CF = 0 so U_{num} = 0; the ITTC recommendation is then to use the standard GCI.[83] Use of this CF as the metric for distance from the asymptotic range is not quite consistent with using p_{obs} / p_{th} or (p_{obs} - p_{th}) because Eq. (6.25.2.1) depends also on r. Two grid triplets with different r's but ~ same p_{obs} / p_{th} will have different distances from the asymptotic range.

In later versions, allowance is made for CF > 1 (apparently superconverging grid triplets with $p_{obs} > p_{th}$) by evaluating CF by an algebraic reflection of either CF (about CF = 1) or the calculated uncertainty itself, in their "Factor of Safety" method (*not* the GCI but a later terminology in Xing and Stern, 2009). This reflection may be intuitively appealing in some qualitative sense but is generally an unjustified assumption. Other *ad hoc* parameters have been introduced to smoothly blend the various formulas, which become tortuous; e.g. see Eqs. (9-19) of Xing and Stern (2009).

In my opinion, the basic concept of variable F_s was worth pursuing. Early publications in the series were admirable in providing candid presentation of many numerical details for real and difficult engineering problems, including issues of maintaining a near-constant r with non-orthogonal boundary fitted grids and consideration of interaction of iteration convergence error with grid convergence errors. The methods were difficult to evaluate because results were based on small sample studies (see comments in Section 5.15) and the CF method failed even in some of these cases (Roache, 2003a,c). Also, the authors had the unfortunate pattern of evaluating competitor methods by comparison to their latest CF method as a benchmark rather than using true performance, e.g. the misleading statement that they found the GCI to be "over-conservative" [with respect to a CF method] rather than examining the data. In the latest variant (Xing and Stern, 2009) a good, statistically significant data set was considered (17 studies encompassing 304 grid triplets). The performance of the latest variant, now unfortunately termed "Factor of Safety method," was

[83] Regrettable confusion exists in terminology and symbols in the ITTC Manual and later related publications. Although the manual cites V&V1 and even recommends use of the GCI for cases close to the asymptotic range, it never uses the term Grid Convergence Index nor the symbol GCI but refers to the standard GCI method as a "factor of safety approach." This unnecessary change in terminology served no purpose but at least was descriptive enough, until Xing and Stern (2009) again switched the meaning to indicate the latest variant of their correction factor method, now unfortunately referred to as the "Factor of Safety method" and given the symbol "FS method," which is easily confused with the original Factor of Safety Fs of the GCI method. (For this reason, I herein avoid denoting their latest variant by their symbol FS.) They also used p_{RE} for p_{obs}. The symbol used for the correction factor was C_k in the ITTC Manual, with k referring to the k-th "input parameter," an unfortunate term since these are not continuum parameters but rather space and time discretization Δ's plus artificial dissipation terms (presumably explicit). In Xing and Stern (2009) the symbol is just CF, which is adopted herein. In their Eq. (15), the uncertainty estimate is described as an error definition. Etc.

good, reaching or exceeding the target 95% level of "confidence[84]." The cost is some additional complexity compared to the GCI method (two parameters and the calculations of CF) and the dependency on p_{th} as discussed above. Whether or not the performance is worth the additional complications is a matter of some judgment. However, the reader is advised to take the authors' comparisons to the GCI (and the variant GCI_c, see next Section) and to simpler, earlier versions of their own CF method with some suspicion. It does not appear that the GCI was correctly applied, i.e. in accordance with the Summary Recommendations of Section 5.9.2, so their evaluation is tainted.

Other questionable aspects of the study are the statistical discarding of outliers, which is questionable for such numerical uncertainty studies as discussed in Section 5.16, and the assumption, required for the statistical interpretation of confidence intervals, that there are no correlated biased errors between different studies.

Furthermore, the latest variant *fails* for the best case reported in Xing and Stern (2008), a realistic study of flow over a ship hull performed by the authors using systematic grid convergence testing up to the finest grid "1" containing 8.1 million points. The most reliable grid sets to consider are the four finest grid triplets (1,2,3), (2,3,4), (3,4,5), (4,5,6) all obtained with $r \sim 2^{0.25} \sim 1.2$. Considering the solution functional of resistance coefficient C_{TX} , the authors dismiss the numerical uncertainty U_{num} calculated by three other methods as "unreasonable [sic] small." There is no justification given for this evaluation. In fact, all three appear reasonable and well behaved in their convergence (U_{num} => 0) for the finest four grid triplets (GCI = 52.7, 4.98, 1.07, 0.58) even though the most coarse grid of the triplet (4,5,6) is out of the asymptotic range. Apparently their negative evaluation of the other three methods is not based on a comparison with the true error, which was not estimated, nor on grid convergence of U_{num} => 0, but on comparison with the authors' latest variant. For example, for the second-finest triplet (2,3,4) the other three methods (GCI, GCI_c, and CF) give U_{num} = 1.07, 1.75, and 1.95 whereas the new variant gives U_{num} = 6.64. Alas, the new variant fails, is "unfortunately invalid," for C_{TX} on the finest grid triplet (1,2,3). The authors claimed this is "caused by the contamination of the iterative error on the fine grid." This is certainly possible, but consider the following points. (a) We do not know how the iterative error was estimated.[85] (b) More importantly, Why was this not a problem for the other three methods? These all used the same data (i.e. results from the same computations). At least we must conclude that the new variant is more sensitive to noise than the other three methods. (c) Worse than that, the new variant is already misbehaving for the previous two grid triplets. The uncertainty estimate U_{num} is supposed to get sharper as the grid is refined. For example, between the grid triplets (3,4,5) and (2,3,4) the GCI estimate drops from 4.98 to 1.07, then further to 0.58 for the finest grid triplet (1,2,3). Similarly for the other two methods. But U_{num} for the new variant *increases* from 6.20 to 6.64, followed by complete failure for triplet (1,2,3). For ship motions rather than resistance, the new variant fails on the *two* finest grid triplets.

[84] This "confidence" is a statistical term, dependent on assumptions and techniques of statistical analysis, rather than just a straightforward "coverage" of counting of cases for which U_{num} was conservative or not, compared to the actual error. The claim of "confidence interval" is based on a conceptual model in which the examined data set of an individual study is taken to represent the entire population, which is inaccessible. If this data set is small, "small sample correction" techniques (like Students *t*-tests) are applicable. The simpler approach just claims (say) "89.2% coverage" for an individual study (perhaps small) with the idea that the results of these small studies will eventually be aggregated (probably informally). If small sample corrections for "confidence interval" are made to each study, later aggregation is confused (because the small sample corrections are nonlinear and non-distributive).

[85] Eça and Hoekstra (2006a, 2009b) have shown that common methods of estimating iterative error are grossly under predictive. See Section 5.10.10.2.

6.26 LEVEL OF ACCURACY ESTIMATES FROM GRID CONVERGENCE STUDIES

An important and somewhat unsettling observation about the practice of accuracy estimation or banding by grid convergence studies was made by Oberkampf et al (1995); see also Blottner (1990). In their experience using Richardson Extrapolation on aerodynamics problems, both for local values (surface pressures) and for solution functionals (force and moment coefficients), they found that one must compute solutions with accuracy in the range of 1% to 0.1% in order to demonstrate that level of accuracy, i.e. to Verify the asymptotic convergence rate upon which the Richardson Extrapolation is based. This is not a problem if one's objective is 1% to 0.1% accuracy, but often engineering analysts would be satisfied with something like 5–10% accuracy, and they understandably are reluctant to pay for the increased resolution necessary to achieve higher accuracy. Haworth (personal communication) had the same experience in modeling of internal combustion engines (Haworth et al, 1990, 1993), and Zingg (personal communication) in transonic airfoils (Zingg, 1991, 1992). The rule of thumb (1% to 0.1% accuracy) is probably problem dependent, but is probably reasonable for all but the easiest problems (e.g., Laplace equations).

So attaining the asymptotic range is necessary to give confidence to the error estimation, but this may already provide more accuracy than one would like to pay for. If error estimation is done outside the asymptotic range, the resulting lack of dependability motivates the large factor of safety (F_s = 3) in the original GCI error banding (Eq. 5.6.1). In our opinion, reliable error estimation *outside* the asymptotic range is not likely for difficult problems (those with significant scale ranges, unlike Laplace solutions).

To put the dilemma succinctly, you may only want to pay for 10% accuracy, but 10% accuracy is difficult to Verify reliably!

This harsh fact of life is softened by the realization that attaining an accuracy level an order of magnitude more than required will *not* require an order of magnitude greater grid refinement, provided that higher than 1st-order methods are used. Unfortunately, it could possibly require an order of magnitude more *cost* for multi-dimensional time-dependent problems. This is where the use of error estimation for "nearby" problems is highly recommended, bringing the cost of reliability of error estimation into the same range as the accuracy requirements.

6.27 OTHER EXAMPLES OF CAREFUL USE OF RICHARDSON EXTRAPOLATION

6.27.1 Fluid Dynamics Examples

Following Oberkampf et al (1995), we recommend four papers as examples of "careful use and estimation of error using Richardson's method" in Calculation Verification. These are Roache (1982), Shirazi and Truman (1989), Blottner (1990), and Walker and Oberkampf (1992). Also recommended is Nguyen and Maclaine-Cross (1988) for application to heat exchanger pressure drop coefficients and Caruso et al (1985).

The papers by Roy and Blottner (2000, 2001, 2003) presented exemplary application of Richardson Extrapolation to difficult problems, solving 1 - and 2 - equation turbulence models in 2-D at hypersonic speeds for a flat place and a sphere-cone. Flat plate calculations at Mach= 8 used a parabolic mesh topology to mitigate the effect of the leading edge singularity. Most results were obtained with 80x160 cells, with error estimation for nearby problems using additional grids of 40x80 and 160x320 cells. Second-order convergence was assumed, based on experience with other related [nearby] problems. Sphere-cone calculations at Mach = 20 used various grids, depending on the turbulence model, including 100x40, 200x80 and 400x160. These papers are among the few to include convincing estimates of incomplete iteration error (see Section 5.10.10.4 for a description of the method). The free flight data included virtually

no experimental uncertainty and the authors considered only error estimates rather than numerical uncertainties, yet all the error estimates (both iteration errors and grid convergence errors) were convincingly shown to be much smaller than the turbulence modeling errors.

6.27.2 § Quantum Chromodynamics Calculation in 4-D Lattice

Durr et al (2008) accomplished a fundamental quantitative confirmation of an aspect of the Standard Model of particle physics with a 4-D lattice computation of quantum chromodynamics. They extrapolated using three finest discretizations, as in Richardson Extrapolation, and the "lattice-spacing dependence of the results is barely significant statistically." The Validation was also successful, predicting from first principles (*ab initio*) the masses of quarks and gluons (ratios to other particles) to within 4%. (See also Cohen, 2008.)

6.28 PARAMETER CONVERGENCES OF A COMPRESSIBLE FLOW CODE NEAR THE INCOMPRESSIBLE LIMIT

Exercising a PDE code over a range of parameters is an activity that does not fit into the semantic categories of Chapter 2 very well, but it is nevertheless a worthwhile exercise in "confidence building", and uses similar techniques. In Roache and Salari (1990), we presented a cursory exercise of a CFD code built for compressible flow near the incompressible limit. There is no question of "Justification" here (see Chapter 2, Section 2.10.3) since the accuracy of the incompressible limit equations is well accepted. However, it is not at all obvious how well a compressible flow code will perform at low Mach numbers. With special care in the formulation of the equations, it is possible to build a code that handles very low Mach numbers, or even $M = 0$. But *most* compressible flow codes, given input parameters that give $M = 0$ (with non-zero velocities) will fail dramatically, e.g. a divide by zero. Many will fail at $M = 0.1$ or 0.2. Also, it is of interest to reveal the effects of low M and other parameters on the convergence rate, and to "justify" not the limit of $M \rightarrow 0$ in general, but to quantify the boundary of the incompressible approximation, which is problem dependent.

The code used in the study was based on the Approximate Factorization (AF) algorithms of Briley and MacDonald (1977,1980) and Beam and Warming (1976) using the "delta formulation." It uses boundary fitted non-orthogonal coordinates, ideal gas relations, constant specific heats, and the Sutherland viscosity-temperature relation. Significantly, the code treats cross-derivative terms without unnecessary approximations; however, it does use the common non-ordered approximation of $\partial p / \partial n = 0$ at walls. Although code versions included 3-D and turbulence models, and 2nd-order time and space accuracy had been Verified, the presently reported study considered only 2-D, laminar, steady flow.

Three steady-state problems were considered:

- a NACA 0012 airfoil at $\alpha = 0$
- a 2-D model of a submarine torpedo shuttleway
- the driven cavity problem.

Sample airfoil results are shown in Figures 6.28.1–2. The *grid* convergence results are given in the original study as well as Salari and Roache (1990). (In Roache and Salari, 1990, the observed spatial rate of convergence was $p > 2$, apparently an artifact of the artificial dissipation at low M.) The grid used in Figures 6.28.1–2 was 185×71. The easily attained value $M = 0.1$ gives effectively "incompressible" results. The code "works" at $M = 0.01$, but evidence of the incipient breakdown due to the singularity at $M = 0$ is beginning to show itself by the small scale pressure oscillations near the trailing edge in Figure 6.28.1.

Figure 6.28.2 presents a comparison between the compressible AF code at $M = 0.1$ and the true incompressible code of Mehta (1977) based on an entirely different mathematical formulation using the stream function-vorticity equations. The excellent agreement between the two codes was, of course, to be expected from the mathematics, but nevertheless constituted a gratifying Confirmation exercise. In fact, the original agreement was not so good, with a 15% discrepancy in pressure near the pressure minimum. The PDE solutions were good, but the incompressible code had to be modified (by I. P. Itty) in the post-processing quadrature for pressure, by using one-sided 2nd-order (3-point) differences for surface normal derivatives, in order to achieve the agreement shown in Figure 6.28.2.

Table 6.28.1 shows the behavior of the aerodynamic coefficients and the maximum dilatation $\nabla\cdot\mathbf{V}$ as M is reduced, compared to the results from the incompressible stream function-vorticity (ψ-ζ) code. The incipient breakdown at $M = 0.01$ (in pressure oscillations near the trailing edge) is evident here; although the total drag looks good, in fact this is due to a lucky cancellation, as the pressure drag has undershot the incompressible value and the friction drag has overshot it. The dilatation $\nabla\cdot\mathbf{V} = 0$ holds identically for incompressible flow. It decreases rapidly with M, and plots of dilatation (see Figure 4 of Roache and Salari, 1990) bear a striking resemblance to experimental interferograms.

The airfoil solution at $M = 0.01$ was obtained with a time step that produced a maximum advective Courant Number = 15, and full (advective + acoustic) Courant Number = O(1500), an impressive stability performance for the AF algorithm. (Note that the breakdown is beginning, but the results are still somewhat usable at this Mach number.)

The fairly complex flow in the torpedo shuttleway (see Figures 5–11 of Roache and Salari, 1990) was obtained again using $M = 0.1$ in a 250×150 grid; no evidence of oscillations occurred, even though one region of the flow had velocities 4–5 orders of magnitude less than the free stream value at $M = 0.1$.

For the driven cavity problem, the parameter value $M = 0.1$ also proved to be adequate for achieving effectively incompressible steady solutions. Comparisons were made with the benchmark solutions of Ghia et al (1982) obtained with (a different) incompressible code based on ψ-ζ variables. That study established that a uniform grid of 129×129 points was adequate, which we confirmed by also running a 65×65 grid case. A more accurate 129×129 stretched (rather than uniform) grid was used to produce Figure 6.28.3, which shows the u-velocity profile through the center of the cavity for $Re = 100$. The agreement of the $M = 0.1$ and 0.05 solutions with the incompressible benchmark solutions is excellent. The disagreement at $M = 0.01$ is pronounced, due to deterioration as the $M = 0$ singularity is approached. Unfortunately, the inaccuracy is only revealed by this convergence study (in this case, convergence as $M \rightarrow 0$) and not by any obvious qualitative breakdown in the solution. We would prefer a more precipitous breakdown, such as the beginning pressure oscillations near the airfoil trailing edge as in Figure 6.28.1. (There is in fact some evidence in a noisy quadrature for stream function near the bottom of the cavity, and poor resolution of the corner eddies.)

A more practical consideration is the effect of the artificial 4th-order explicit damping coefficient DC. All the previously described solutions were obtained with the code default value of $DC = 0.01$. We increased DC by a factor of 2, and decreased it by a factor of 50, with no significant effect on the solutions. Only when DC was further increased to 1.4 (140 times the nominal value) *and* the grid spacing near the boundary was doubled did a noticeable effect show up. (See Figure 16 of Roache and Salari, 1990.)

The agreement with the downward v-velocity component from the incompressible benchmark solution is not as good, differing by < 2.5%. (See Figure 14 of Roache and Salari, 1990.) The speculation (in Roache and Salari) that the most likely reason is the inherent error at low M in the Approximate Factorization (AF) procedure itself (see Beam and Warming, 1976) is incorrect; as pointed out by a reviewer, this error is negligible for good steady-state iteration convergence with the delta-form of AF.

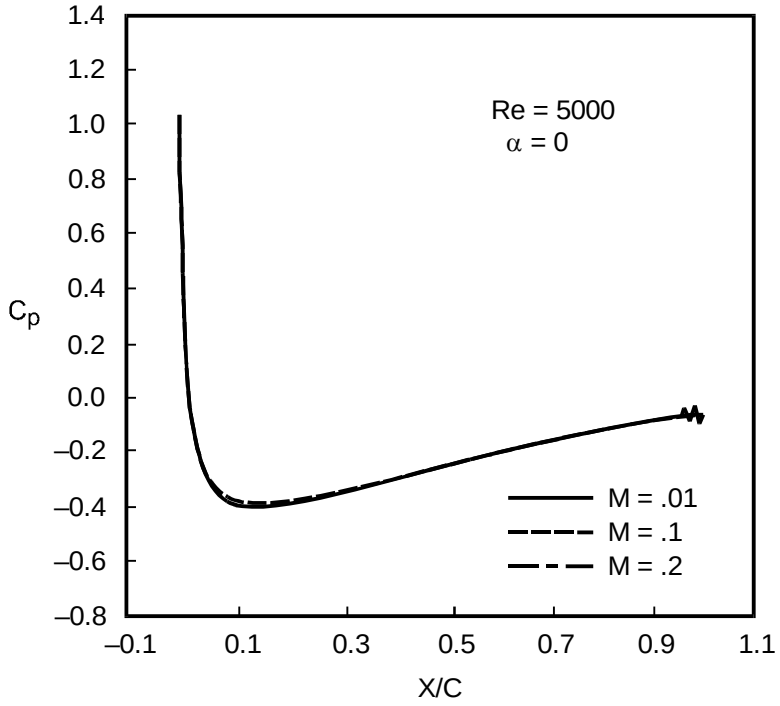


Figure 6.28.1. Pressure coefficients on NACA 0012 airfoil at three Mach numbers. Laminar flow, *Re* = 5000, $\alpha = 0$. (From Roache and Salari, 1990, Figure 2.)

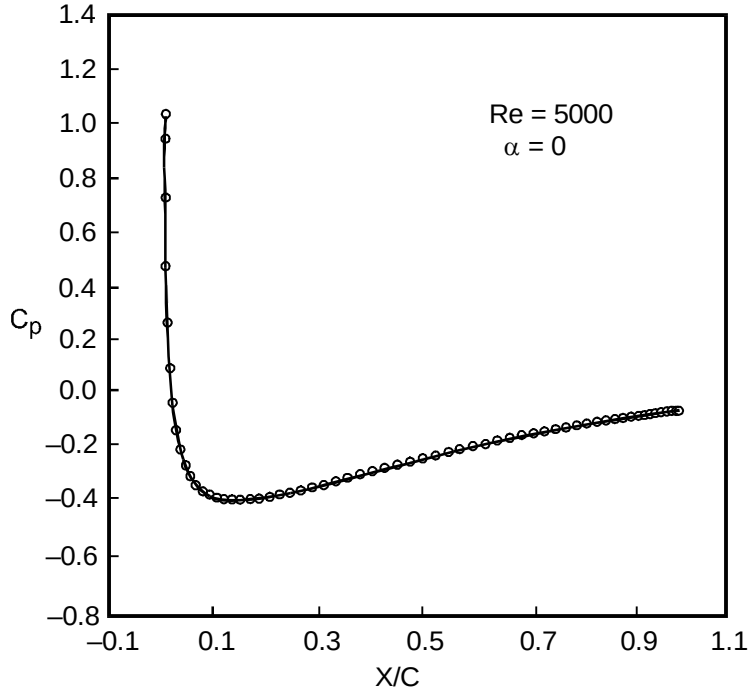


Figure 6.28.2. Pressure coefficients on NACA 0012 airfoil calculated by a compressible flow code at Mach number = 0.1 and by an incompressible code based on stream function-vorticity variables. Laminar flow, *Re* = 5000, $\alpha = 0$. (From Roache and Salari, 1990, Figure 3.)

Mach Number	Total Drag C_D	Pressure Drag C_{DP}	Friction Drag C_{DF}	Stagnation Pressure C_P	Maximum $\nabla\cdot\mathbf{V}$
0.34*					0.89*
0.20	0.05282	0.01850	0.03432	1.0448	0.145
0.10	0.05268	0.01778	0.03491	1.0384	0.041
0.01	0.05331	0.01589	0.03742	1.0302	0.004
Incompressible	0.05341	0.01864	0.03477	1.0282	–

Table 6.28.1. Aerodynamic coefficients and the maximum dilatation as *M* is reduced. NACA 0012 airfoil, laminar flow, *Re* = 5000, α = 0. The "Incompressible" results were obtained with a separate code based on a ψ-ζ formulation. (*The value of dilatation at M = 0.34 was obtained for a high *Re* turbulent flow case.) (From Roache and Salari, 1990, Table 1.)

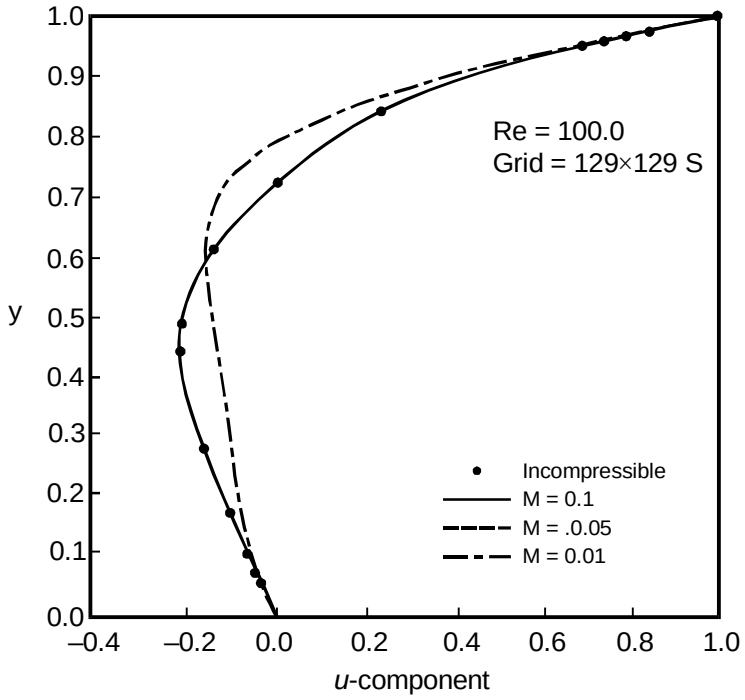


Figure 6.28.3 *u*-velocity profile through the center of the cavity for *Re* = 100.
(From Roache and Salari, 1990.)

Additional quantitative results are shown in Table 6.28.2, which displays convergence as $M \rightarrow 0$ of minimum density ρ, minimum stream function ψ (or primary vortex strength), and maximum dilatation $\nabla \cdot \mathbf{V}$. The minimum stream function is also given from the incompressible ψ-ζ benchmark solution of Ghia et al (1982). Note that the ψ evaluation for the compressible code involves additional numerical error in quadratures, as well as compounded errors in velocities and densities.

The study also investigated the effects of ratio of specific heats γ and the 4th-order explicit damping coefficient *DC*. The previously cited results were obtained with $\gamma = 1.4$ (air). At $M = 0$, the continuum results will not depend on the thermodynamic parameter γ. Consideration of the gas dynamic relationships suggested that a low γ solution would be a better incompressible approximation. Indeed, results for $\gamma = 1.1$ and 1.005 at $M = 0.1$ showed a somewhat decreased range of the dilatation, as shown in Table 6.28.2. (The use of $\gamma = 1$ causes indeterminacies in the code, as $M = 0$ does.)

This cursory study considered the effects of Mach Number, ratio of specific heats, and artificial 4th-order damping on the behavior of an Approximate Factorization code for compressible flow in the low Mach number range for steady flows. The parameter study was far from complete, but might suggest an interesting and more complete study. Highly transient solutions are expected to be more difficult.

Mach Number	γ	minimum ρ	minimum ψ	max $\nabla \cdot \mathbf{V}$
0.10	1.4	0.89028	–0.10208	0.226
0.05	1.4	0.97244	–0.10296	0.028
0.01	1.4	0.99885	–0.08531	0.00099
Incompressible	–	–	–0.10342	–
0.05	1.1	0.97977	–0.10306	0.0222
0.05	1.005	0.98218	–0.10309	0.0203

Table 6.28.2. Driven cavity solutions for minimum density ρ, primary vortex strength (minimum ψ), and maximum dilatation $\nabla \cdot \mathbf{V}$. The "Incompressible" results were obtained with a separate code based on a ψ-ζ formulations. γ is the ratio of specific heats. (From Roache and Salari, 1990, Table 3.)

6.29 JUSTIFICATION OF THE DUPUIT APPROXIMATION

The previous Section 6.28 presented an example of a numerical study of parameter convergence, an activity that does not fit the definition of Verification but uses similar techniques. Herein, we consider an example of another such activity, that of "Justification" of modeling equations. As discussed in Chapter 2, Section 2.10.3, the distinction between Verification and Validation may not be obvious when one considers a code based on some sort of simplified equation set.

Agreement between results of the simplified and complete equations is not strictly included in the term Verification, since the Verification of the simplified equation code has already been completed prior to the full equation comparison. One could say that the agreement has demonstrated that the simplified code is "solving the right equations" in one sense, i.e., it justifies the use of simplified equations. Yet to claim Validation would be over-reaching, since we have not demonstrated the adequacy of the more complete model by comparison with experiment. We have "solved the right equations" only in an intermediate sense of demonstrating that the simplified equations adequately represent the more complete equations, but not in the ultimate sense of "solving the right *physical* equations." Appealing to the other distinction between

Verification and Validation based on mathematics vs. science, it is clear that such a comparison exercise should be categorized with Verification rather than Validation, but that the categorization is somewhat inadequate because the simplified code can (should) have been fully Verified in the more usual sense before the comparison exercise was started. If the only choice is Verification or Validation, this gray area is best categorized as Verification, with the term Validation reserved for experimental comparisons. Even better is to introduce another term, Justification.

As an example of Justification, consider the Dupuit approximation, a small-angle approximation for groundwater flows that neglects any effects associated with vertical velocity. This approximation is virtually ubiquitous in groundwater flow codes, e.g. MODFLOW (McDonald and Harbaugh, 1988). The Dupuit approximation is so commonly used that a modeler can assume it with little chance of criticism in review, yet very early Bear (1972) gave the following requirements for its use (from Knupp et al, 1996).

- $(K_{hor} / K_{ver})s << 1$ (K = conductivity, s = slope of the water table)
- no recharge (so that the phreatic surface is a streamline)
- no seepage faces
- absence of no-flow boundaries on sides of domain
- steady state
- no local “pits or mounds” in the water table
- no well terms that would deform the water table

Knupp et al (1996) compared the results from MODFLOW, which uses the Dupuit approximation, with the results from SECO_FLOW_3D, which has an option to use the full nonlinear phreatic boundary conditions (Knupp, 1996). Each code was Verified to produce consistent (ordered) solutions to its own equation set. Interestingly, the development of an exact Benchmark analytical solution was easier for the full equations, and use of this full solution confused Verification of a simplified code. This provided additional motivation for development of a full equation code (Roache et al 1996) .

An example from the Verification exercises of the complete equation code is shown in Figure 6.29.1. The largest error, at the right boundary, was 0.005%. In Figure 6.29.2, the comparison of the full equation analytical solution with the result from the MODFLOW code shows (primarily) the effect of the Dupuit approximation. At worst, it is 8% in error. (See Knupp et al, 1996, for details of the test case parameters, resolutions, and fine modeling points required for the MODFLOW simulation.)

The Justification of the Dupuit approximation will obviously be problem dependent (see also Serrano, 1995). Knupp et al (1996) found that it was important to use the full phreatic condition when

- heterogeneity increased,
- recharge was higher, and
- well pumping occurred.

For some parameter combinations, the error in head was 24%. Furthermore, the true phreatic boundary condition eliminated some instances of non-unique solutions obtained with the steady-state option in MODFLOW. On the other hand, the phreatic condition code is more demanding of accurate recharge estimates (Roache et al, 1996). Errors in fluxes are more severe than errors in head. For a suite of test cases using SECO_FLOW_3D with and without the Dupuit approximation, it caused 5–10% errors in heads and water table elevations, but up to 50% error in fluxes (Roache et al, 1996), which would correspondingly affect contaminant transport calculations.

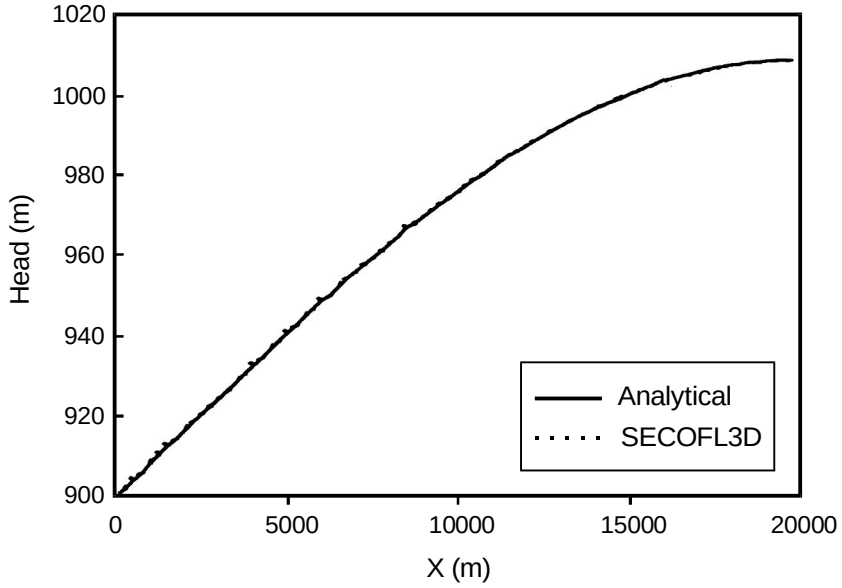


Figure 6.29.1. Water table solutions from an analytical solution of the full equations and the SECO_FLOW_3D code using the full nonlinear phreatic boundary condition. (From Figure 3-1 of Knupp et al, 1996.)

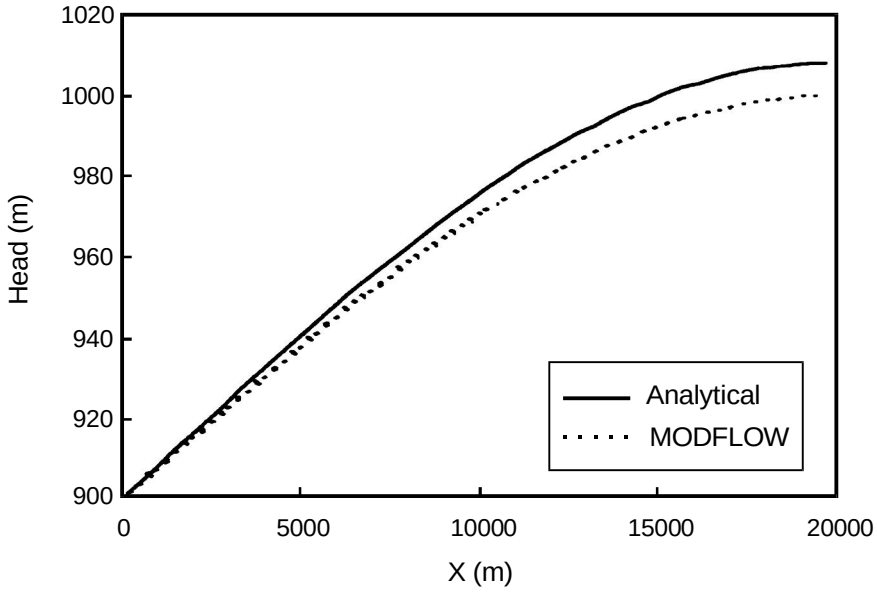


Figure 6.29.2. Water table solutions from an analytical solution of the full equations and the MODFLOW code using the Dupuit approximation. (From Figure 3-3 of Knupp et al, 1996.)

6.30 PARAMETER UNCERTAINTY AND NUMERICAL UNCERTAINTY

I offer a comment on the philosophy expressed by many flow modelers. In the field of groundwater modeling, there is a tremendous uncertainty in the physical parameters involved. This uncertainty has often been used to excuse sloppy numerical work. For example, in one of the contributions to the international HYDROCOIN project (OECD, 1988), a grid doubling in the horizontal plane caused a 40% reduction in peak value of hydraulic head; yet most of the contributors used only a single grid and a single time step; also, the various calculations produced "a very wide spread in travel times." The same excuses have been used by aerodynamicists using turbulence models, or other approximations. The basic concept is to excuse laziness in the numerical work by noting the uncertainty in the conceptual model. Often, one hears that "we only want the answer to 10% accuracy" in the case of aerodynamics, or "order of magnitude" for groundwater flow modelers.

The first and obvious response is that there should be no *preference* for doing inaccurate numerical work, and methods are available to assess the numerical accuracy. Of course, the cost trade-offs must be decided by engineering judgment, especially in a design environment, and I would not second-guess those with deadlines to meet.

The second response, especially to the groundwater flow modelers, is to note that poor (1st-order or hybrid) numerical methods for advection terms (not so much in groundwater flow itself, but in the subsequent contaminant transport calculation) can in fact introduce an *order of magnitude error*. In the 1950's, the late Senator Dirksen of Illinois commented on military budget resolution discussions by noting "a billion dollars here, a billion dollars there, and pretty soon you are into some real money." To paraphrase the Senator for the groundwater flow and transport modelers, "an order of magnitude here, an order of magnitude there, and pretty soon you are into some real errors."

Third and finally, if a groundwater transport modeler will be satisfied with order-of-magnitude numerical accuracy, I would argue that difficult computer simulations may not be necessary. Rather than go through the charade of obtaining solutions to variable coefficient partial differential equations, the modeler can obtain the needed estimates from closed form solutions (or very simple numerical solutions of Laplace equations) by assuming constant properties, using averaged value of parameters and engineering intuition. This has the virtue of exposing the level of uncertainty, rather than concealing the uncertainty with the of gloss of mathematics.

If we are going to use partial differential equations in our conceptual model, then we should solve them honestly.

6.31 § PARAMETER UNCERTAINTY AND MODEL FORM UNCERTAINTY[86]

A thorough validation study must consider input parameter uncertainty. The estimation of parametric (standard) uncertainty u_{input} is meaningful only after a set-point (nominal-valued) simulation has been completed. But note that some, or even all, of the parameters in the model formulation may be considered hard-wired values inherent to the model, and therefore not contributors to u_{input}. If *all* parameter values are considered fixed in the model, this is the limit of what has been termed a strong-model approach. (See Section 9.18.)

[86] Adapted from Appendix C, Subsection 6 of V&V20.

In addition to parametric uncertainty, *model form uncertainty* (and more fundamentally, model form error) arises when incomplete physics are incorporated into the model. The distinction between parametric uncertainty and model form uncertainty can be gray. For example, a hypersonics code can contain a chemistry package. The rate constants may be hard-wired or may be accessible to the user. Two different codes can contain exactly the same chemistry model, one with hard-wired constants, the other with user-input values.[87] In the first case, those rate constants would need to be treated as an essential part of the hypersonics model, therefore leading to model form error and uncertainty. In the second case, the analyst has the freedom to study the associated parametric uncertainty but is not required to do so. (Surely the decision to consider the values of the rate constants as part of model form or model parameters should not be dictated by code structure.) With the same model and code, the same lack of knowledge of the chemistry rate parameter could be categorized as either model form uncertainty or input parameter uncertainty. Either choice is acceptable, but the documentation for any study must be clear.

Both parametric uncertainty and model form uncertainty are generally present, and both contribute to the validation uncertainty. With or without estimation of u_{input}, neither uncertainty is ignored; their effects simply result in an overall validation uncertainty. When parametric uncertainty is completely analyzed, the validation uncertainty resulting from the comparison of experimental results with simulation results *is* the model form uncertainty.

6.32 § PARAMETER UNCERTAINTY IN VALIDATION VS PREDICTIVE ANALYSIS

It is important to distinguish between parametric uncertainty in a *validation exercise* vs. parametric uncertainty in a *predictive analysis* (e.g. Helton et al, 1995). When parametric uncertainty is quantified in a validation exercise, the remaining model form uncertainty is not ignored; rather, it is manifest in the validation uncertainty. That is, the model form uncertainty will be *evaluated* by the validation uncertainty (see Eq. 11.5.1 and 11.9.1). However, in a predictive analysis (in which the physical answer is not known), full coverage of parametric uncertainty cannot be assumed to cover all possible results because model form uncertainty is not represented. For example, in the problem of a fin tube heat exchanger in V&V20, unlimited variation of the other parameters will not reach agreement for a physical problem dominated by contact resistance if that phenomenon is not part of the model form. *Thus, even a full study of parametric uncertainty in a predictive analysis does not account for all sources of modeling error.*

Parameter Uncertainty is a fundamentally different concept for Validation experiments than it is for predictive analysis like design. Consider first Validation for a backstep flow, with a CFD simulation done cooperatively with design of the experiment, as recommended in V&V10. The backstep height b is a parameter, and the design decision is $b = 1$ cm. The experiment contains both errors and uncertainties in b, but the simulation does not. There is no uncertainty in the b used in the simulation; the input value is $b = 1$ cm. (Except for machine round-off error, which is negligible.) Likewise for other parameters. There is no uncertainty about what value of viscosity coefficient is used in the simulation. The experimental error and uncertainty will be accounted for in the Validation approach of V&V20 (see Chapter 11) and should not be counted twice. Now consider a *design analysis* involving flow over a backstep. The actual physical dimension b is uncertain (as is viscosity, etc.) and the model must be exercised to cover this uncertainty.

[87] User access to model parameters leads to serious Quality Assurance (QA) issues, and confuses the meaning of claims of a "Validated code." See further discussion in Section 12.13.

6.33 § PARAMETER UNCERTAINTY AND THE ECONOMICS OF GRID CONVERGENCE STUDIES

It is often heard that one cannot afford to do error estimation by grid convergence studies because of the high cost of multidimensional calculations (addressed in Section 5.15). Likewise, statements are often heard like "We cannot afford to do grid convergence studies because we are not academics, we are in an industrial environment, we have hundreds or thousands of *parametric studies* to run." First, the cost of grid convergence studies ought to be normalized by the base cost of the problem set, including parametric studies. If (as in Section 5.15) the incremental cost of doing a 4 grid convergence study using grid coarsening with $r \sim 1.3$ on a 3-D time accurate problem is ~52%, then that % penalty would apply no matter how many parameter values were used. But in fact, the % penalties can be radically improved if we can justify sampling the parameter space for grid convergence studies. If the parameter values chosen are close enough that the variation in solution variables of interest is fairly well resolved, then it is likely that some of these problems are "nearby" in the sense that numerical error and uncertainty estimates are close.[88]

Suppose we judge (from numerical experimentation or just good engineering judgment) that grid convergence testing is required for only every other parameter value; we will include end points to avoid extrapolation. For example, 7 parameter values to be run in simulations might require only 4 grid convergence tests (underlined).

Parameter Values = A B C D E F G

For a single parametric variable (i.e. a one-dimensional parameter space) this obviously reduces the required grid convergence test sequences by the factor 3/7 or 43%. (The 52% penalty for performing a 4-grid convergence study would be reduced to 0.43 × 52% = 22.4%.) But, as in the case of spatial dimensions for PDEs (Section 5.17), parameter space higher dimensionality is a blessing. Two parameters similarly sampled give a factor of $(3/7)^2 = 0.184$. In a 5-dimensional parameter space[89] with 7 values in each parameter and only 4 of the 7 used for grid convergence studies, the factor is a trivial $(3/7)^5 = 0.0145$.

In conclusion, the common industrial situation of high dimensional (space and time) problems and high dimensional parameter space (extensive parameter lists) make systematic grid convergence testing relatively inexpensive, when the cost is properly normalized, and enables reliable and defensible estimates of errors and uncertainties.

---

[88] See Sections 5.6, 5.8, 5.9.1, 5.10.5 for some discussion. One should also consider interpolation of uncertainty estimates in the parameter space; see Section 11.12.

[89] For example, for a study of parameterized wing shapes, one might consider the 5-dimensional parameter space = {thickness/chord, camber, wing aspect ratio, angle of attack, Reynolds number}.

CHAPTER 7

SINGLE GRID ERROR ESTIMATORS

This Chapter[90] covers methods of error estimation and error banding that do not involve systematic grid convergence testing. These methods are in Categories B, C, and D of the following taxonomy (previously given in Chapter 4) for sources of additional information for error estimation.

B. Additional Solution(s) of the Governing Equations on the Same Grid
   B.1 Higher Order Accuracy Solutions
   B.2 Lower Order Accuracy Solutions
C. Auxiliary PDE Solutions on the Same Grid
D. Auxiliary Algebraic Evaluations on the Same Grid; Surrogate Estimators
   D.1 Non-Conservation of Conservation Variables
   D.2 Non-Conservation of Higher Moments
   D.3 Zhu-Zienkiewicz and Wiberg-Type Estimators
   D.4 Convergence of Higher Order Quadratures

Grid generation can be problematical and multiple grid generation required for grid convergence studies is always troublesome. Thus, single-grid error estimators are very much of interest. As noted in Chapter 4, these methods (unlike Category A methods, grid convergence studies) cannot be used for Verification of Codes, but can be used with Verified Codes for the Verification of individual Calculations. Although they require no additional grid generation if it is assumed that the order of convergence is known, they involve significant additional algorithm and code development, beyond that required for the basic solution code.

[90] Taken primarily from Roache (1997), "Quantification of Uncertainty in CFD" and from Pelletier and Roache (2002).

Unlike the methods described in Chapter 5 and 6, the single grid error estimators are not described herein in sufficient detail to implement them.[91] It would not be appropriate to go into such detail herein (nor am I qualified to do so). The reason is that these methods are closely related to, and in some cases an intrinsic part of, the solution algorithms. Whereas the methods of Chapter 5 and 6 are applied externally to the code that produces solutions to the PDEs, the single grid methods are either part of that code (the usual situation), or could be post-processing codes whose structure resembles the PDE code. To use a common description, the single grid error estimators are "code intrusive." For example, a user of a commercial code might request the vendor to include a single grid error estimator (at considerable cost) but could not apply it on his own *a posteriori*, as he could the GCI, for example. By contrast, the methods of Chapter 5 and 6 do not require access to the inner working of the PDE code but are applied externally, often with a simple spreadsheet or even hand calculation.

7.1 ERROR ESTIMATION FROM HIGHER OR LOWER ORDER ACCURACY SOLUTIONS ON THE SAME GRID (CATEGORY B)

7.1.1 Higher Order Accuracy Solutions (Category B.1)

Category B1 methods estimate the accuracy of a base solution by comparison with "higher order accuracy solution(s)." (Again, note the somewhat abusive but common terminology.) Richardson (1910) again scooped modern error estimation papers by inventing Category B.1, error estimation from higher order solutions on the same grid, noting that the difference between a 2nd-order accurate solution and a 4th-order accurate solution is itself an ordered error estimator.

The higher order accuracy solution might be obtained via a new solution of higher-order discretizations (FDM, FVM, FEM, etc.) or by deferred corrections, compact differences (again by direct or deferred corrections), etc.

For the technique of error estimation from higher order solutions on the same grid, much the same advantages and limitations apply as with the grid convergence technique, i.e., it applies to all point values and functionals (lift, drag, etc.); x and t errors may be estimated independently or coupled; the error estimate includes nonlinear coupling. These methods are not commonly used because they require additional code capability, unlike grid convergence tests (Category A). On the other hand, these Category B methods do not require additional grid generation.

The development costs of the additional code capability may be reduced by noting several points. If we were intending to use the higher order solution itself, many restrictions and requirements could apply, such as full iteration convergence, strict conservation, etc. However, if our only use of the higher order solution is to estimate the error of the base solution, these considerations are not so important. The point is that error estimation of (say) a 2nd-order solution using 4th-order methods is less demanding numerically than obtaining a 4th-order solution for direct use. Also, though not generally recognized, directional splitting works (Roache, 1997). (The following development has not been completed for cross derivative terms, which in any case may require some careful formulation for higher order stencils and should be rigorously verified.)

We use a notation similar to that above for multiple grid solutions, with f denoting the exact solution, f_2 the 2nd-order accurate solution, and f_4 the 4th-order accurate solution. C_{2x} are the coefficients of the Taylor's theorem expansion for the 2nd-order solution in the x-direction, etc., and R_{2x} are the remaining terms in the complete 2nd-order expansion in x, etc. Then, in 2-D,

[91] The time error estimators are described in sufficient detail.

$$f = f_2 + C_{2x}\Delta x^2 + R_{2x} + C_{2y}\Delta y^2 + R_{2y} \tag{7.1.1.1}$$

$$f = f_4 + C_{4x}\Delta x^4 + R_{4x} + C_{4y}\Delta y^4 + R_{4y} \tag{7.1.1.2}$$

Defining the error of the 2nd-order solution E_2 to be

$$E_2 = f - f_2 \tag{7.1.1.3}$$

and substituting for f from Eq. (7.1.1.2) we obtain

$$E_2 = f_4 - f_2 + O(\Delta^4) \tag{7.1.1.4}$$

This is the basic (Richardson, 1910) result that the difference between the 2nd- and 4th-order solutions on the same grid is itself a 4th-order error estimate for the 2nd-order solution, a somewhat obvious result. Likewise obvious, if we define the error of the 4th-order solution E_4 to be

$$E_4 = f - f_4 \tag{7.1.1.5}$$

and substitute for f from Eq. (7.1.1.2) we obtain

$$E_4 = f_2 - f_4 + O(\Delta^2) \tag{7.1.1.6}$$

for a 2nd-order error estimator for the 4th-order solution.

The somewhat less obvious result is for the directional splitting of the higher order solutions, which can be much easier to implement than a fully directional higher-order solution. Let f_{4x} denote the solution obtained with 4th-order discretization in x and 2nd-order in y, and, f_{4y} denote the solution obtained with 4th-order discretization in y and 2nd-order in x. The C_{2x}, etc., coefficients are unchanged from the previous definitions. Then

$$f = f_{4x} + C_{4x}\Delta x^4 + R_{4x} + C_{2y}\Delta y^2 + R_{2y} \tag{7.1.1.7}$$

$$f = f_{4y} + C_{4y}\Delta y^4 + R_{4y} + C_{2x}\Delta x^2 + R_{2x} \tag{7.1.1.8}$$

We now estimate the difference between f_{4x} and f_2, using Eqs. (7.1.1.7) and (7.1.1.1) respectively. (If we dropped the higher order remainder terms at this point, we would show the resulting estimator to be 2nd-order accurate, but by retaining these terms presently, we will show that the estimator is 4th-order accurate.)

$$\begin{aligned} f_{4x} - f_2 &= f - C_{4x}\Delta x^4 - R_{4x} - C_{2y}\Delta y^2 - R_{2y} \\ &\quad - (f - C_{2x}\Delta x^2 - R_{2x} - C_{2y}\Delta y^2 - R_{2y}) \\ &= C_{2x}\Delta x^2 + R_{2x} - C_{4x}\Delta x^4 - R_{4x} \end{aligned} \tag{7.1.1.9}$$

Similarly, we estimate the difference between f_{4y} and f_2 using Eqs. (7.1.1.8) and (7.1.1.1) respectively.

$$\begin{aligned} f_{4y} - f_2 &= f - C_{4y}\Delta y^4 - R_{4y} - C_{2x}\Delta x^2 - R_{2x} \\ &\quad - (f - C_{2x}\Delta x^2 - R_{2x} - C_{2y}\Delta y^2 - R_{2y}) \\ &= C_{2y}\Delta y^2 + R_{2y} - C_{4y}\Delta y^4 - R_{4y} \end{aligned} \tag{7.1.1.10}$$

Adding (7.1.1.9) and (7.1.1.10) gives

$$\begin{aligned} f_{4x} + f_{4y} - 2f_2 = C_{2x}\Delta x^2 + R_{2x} + C_{2y}\Delta y^2 + R_{2y} \\ -C_{4x}\Delta x^4 - R_{4x} - C_{4y}\Delta y^4 - R_{4y} \end{aligned} \tag{7.1.1.11}$$

Comparing to Eq. (7.1.1.1), the first 4 terms on the RHS of (7.1.1.11) are identically equal to (f - f_2) without approximation. (That is, the remainder terms R_{2x} and R_{2y} are included.) The last 4 terms on the RHS of (7.1.1.11) are O(Δ^4). With the definition (7.1.1.3), we have the error of the 2nd-order solution E_2 estimated as

$$E_2 = f_{4x} + f_{4y} - 2f_2 + O(\Delta^4) \tag{7.1.1.12}$$

Thus the estimate for the 2nd-order solution can be obtained to 4th-order by directionally split 4th-order solutions.

7.1.2 Lower Order Accuracy Solutions (Category B.2)

Certainly a lower-order solution on the same grid could be used to estimate the error of the higher order solution (Category B.2), in the same way that grid coarsening can be used rather than grid refinement, but We know of no such applications.

7.2 Δ AUXILIARY PDE SOLUTIONS ON THE SAME GRID (CATEGORY C)

The methodology of Category C (Auxiliary PDE Solutions on the Same Grid) does not simply involve a local evaluation of something. The key aspect here is that errors are transported - advected, diffused, etc. [Babuska et al (1994,1997) refer to this phenomenon as “error pollution.”]

7.2.1 Δ Error Transport Equations

Most of the papers using the approach of error transport equations have appeared in the FEM literature, but the concepts are equally applicable to FDM and FVM. An early development for FDM limited to cartesian grids was presented by Schonauer et al (1981). Here, we follow Van Straalen et al (1995) and introduce the concepts with the simple non-conservation form of the steady-state 2-D linear advection-diffusion equation with a source term,

$$V\nabla\Phi - \nabla(\Gamma\nabla\Phi) = q \tag{7.2.1}$$

where V is the advection velocity, Φ is the concentration of the transported scalar, Γ is the diffusivity, and q is a source of Φ (per unit mass). We write a simplified form of (7.1.13) using Ψ to represent the entire partial differential equation operator.

$$\Psi(\Phi) = q \tag{7.2.2}$$

Let ϕ_{approx} be some *continuum* approximate solution of Eq. (7.2.1) obtained numerically. It will not be sufficient to consider only point-wise (or node-wise) numerical solutions; rather, ϕ_{approx} must be defined in the operator domain of Ψ, i.e., for this example problem, it must be a 2-D continuous function with a least second degree derivatives. (How this continuum approximate solution will be generated from nodal FDM or FVM solutions, which we denote by ϕ_{nodal} for the sake of emphasis, will be discussed momentarily.) Then the error function is defined as $\varepsilon \equiv \Phi - \phi_{approx}$ and from the properties of linear operators we can define the so-called "*Exact Operator Residual*" as

$$R(\varphi_{approx}) \equiv \Psi(\varphi_{approx}) - q \tag{7.2.3}$$

The terminology of "*Exact Operator Residual*," though common, is somewhat misleading, in my opinion, because it is "exact" only for a given continuum ϕ_{approx} but is not uniquely defined for any ϕ_{nodal}. That is, if the ϕ_{nodal} we obtain from a code is all we have, we cannot define an operator residual until we extend ϕ_{nodal} to ϕ_{approx} and this procedure is somewhat arbitrary. For example, generation of ϕ_{approx} by piecewise linear interpolation (i.e., basis functions) will not be adequate for the method as described so far, because derivatives of this ϕ_{approx} will generally not exist at cell interfaces, and such ϕ_{approx} will not be in the operator domain of Ψ.

Once we have defined a ϕ_{approx} from ϕ_{nodal} and defined $R(\phi_{approx})$, we note that the error ε is given *exactly* by the solution of the error equation,

$$\Psi(\varepsilon) = -R(\phi_{approx}) \tag{7.2.4}$$

The numerical solution of this equation provides the "Auxiliary PDE Solutions on the Same Grid" that constitutes Category C of error estimators.

The auxiliary error equation (7.2.4) utilizes the exact operator residual R as a source term just as the original PDE (7.2.2) contains the source term q. This provides an instructive conceptual model for the error propagation process; the exact operator residual R is a "source" for the global error distribution, which is transported by advection and diffusion. Clearly, the solution of the error equation is non-local.

Following Van Straalen et al (1995), we also distinguish the Exact Operator Residual (7.2.3) from the *Approximate Operator Residual* r_L which also appears in the numerical analysis literature, defined as

$$r_L = L(\Phi) - q \tag{7.2.5}$$

where L is the discretized operator that one has solved, i.e. that approximates Ψ. Study of the residual r_L leads to a discussion of *order*, since r_L is typically expressed as a Taylor-series expansion about the exact solution. For example, see Ferziger (1988).

The numerical solution of Eq. (7.2.4) provides the approximate global solution for the exact error ε. There are two decisions to be made on the approximations: extending ϕ_{nodal} to ϕ_{approx} and then a

discretization of Ψ (ε), which is (virtually) the same decision one had to make for the original problem. (The differences are that boundary conditions for the error equation will be homogeneous, and any algorithmic aspect affected by the magnitude of the solution, e.g. nonlinear flux limiters, will be triggered differently.) Thus, many variations on this methodology are possible. By considering an integral form of the error equation (7.2.4), Van Straalen et al (1995) eliminate the requirement for smoothness in ϕ_{approx} and use piecewise linear functions. This is a methodology from FEM theory, including the ubiquitous integration by parts, adapted to FVM. Note, however, the following universal point in all these methods. It is essential that the discretizations used to define ϕ_{approx} and the operator in the error equation be distinct from those used on the original PDE, otherwise the method becomes a meaningless circular definition. In Van Straalen et al (1995), the error estimation works only because the original PDE is discretized using 1st-order FVM for advection, and advection is the dominant error source, and the error equation is solved using 2nd-order methods.

The procedure for error estimation in Category C (Auxiliary PDE Solutions on the Same Grid) then involves selection of an alternate functional form for the inter-cell (inter-element) solution and evaluation of the (differential) residual. It is possible that the derivatives may not exist at element boundaries. Whether one selects a C^2 smooth function to begin with, or one blends (interpolates) the differentials obtained at element (or cell) centers to produce smooth functions over the entire regions, or puts off such questions to the quadrature rules, etc. is not really very significant. The residuals are then transported in an error equation for Category C methods; alternately, they could be used as local error estimators (as in Category D methods discussed below) which is indeed a significant distinction. The major point (for both Category C and D methods) is that the residual evaluation must involve a discretization rule different from that used in obtaining the solution. In this aspect, these methods are, after all, not so distinct from category B methods, which use higher (or lower) order stencils to evaluate the error. In category C, a higher order stencil is used to obtain a higher-order solution, but if the stencil were used only locally to obtain a local residual, it would be like Category D. The approach in which a rigorously iterated solution is replaced with a less completely converged solution is intermediate, the "globalness" of the error estimate now being dependent on the amount and type of iteration.

For earlier examples of the error transport equation approach in FDM and FVM, see Schonauer et al (1981), Ferziger (1993), Van Straalen et al (1995). The FEM literature is extensive, e.g. see Mills (1987), Babuska et al (1994,1997), Strouboulis and Oden (1990), Ewing (1990), Padra and Larreteguy (1995). More recently, Celik and Hu (2003) have re-visited the idea of a single grid error estimator based on numerical integration of a transport equation for truncation error. In all cases, the error transport equation will be some linearized version of the governing equations. It will be somewhat cheaper to solve than the original fluid dynamics equations (e.g., full Navier-Stokes) but, as expected, will be a less accurate error estimator near boundary layer separations, etc. This approach is truly global, but not as reliable as Category A or B methods. Among other attributes, this approach holds out the possibility of ordered error estimation for Lagrangian methods, which are in a primitive condition.

7.2.2 § Adjoint Equations

Another approach to single-grid error estimation that fits into Category C is that in which the auxiliary PDEs are the adjoint of the governing time-dependent PDEs. (The governing time-dependent PDEs solve the effects of a local point value, e.g. a perturbation, on all other locations, while the adjoint equations solve the effect of values at all other locations on the local point value.) This concept can be adapted to both error estimation and sensitivity analysis. Pernice (2007) and colleagues have developed several codes that provide error estimation for multiphysics simulations. Ragusa (2008) has developed single grid error

estimation and grid adaptation methods for higher-order multiphysics computations that can be applied as external subprograms to legacy codes, a remarkable and practical accomplishment.

7.3 Δ AUXILIARY ALGEBRAIC EVALUATIONS ON THE SAME GRID: SURROGATE ESTIMATORS (CATEGORY D)

Category D methods are described generically as Auxiliary Algebraic Evaluations or in shorthand, AAE. What they have in common is their modus of application; they all involve *local* algebraic processing of a single grid solution. A recommended overview of the theoretical development of AAE is given by Ainsworth and Oden (2000) who refer to this category simply as "*A Posteriori* Estimators." AAE methods require only one grid generation *if the order of convergence is assumed to be known*, are relatively cheap, and (sometimes) use no significant dynamic memory. If the order of convergence is not known (or if the theoretical order is not trusted) these "single grid" methods are better described as "one-less grid methods", i.e. they require one less grid than classical gird convergence testing. The two broad categories within AAE are residual-based methods and recovery methods, the most widely known of which are the Zhu-Zienkiewicz family (ZZ).

AAE can also be described as "error indicators" or "surrogate estimators" rather than error estimators, because the energy norm metric on which they are based is not usually of any direct engineering or scientific interest. (This metric is of interest in meteorological and ocean calculations.) These indirect AAE are useful for engineering use only if correlated by experience with quantities of direct engineering interest. But remarkably, Ainsworth and Oden [41] have shown how the AAE may be extended from merely the energy norm (which has fundamental theoretical significance) to functionals like Nusselt number, drag coefficient, etc., which they refer to generically as "quantities of interest." The major part of the book involves linear strongly elliptic problems, with the last six pages covering quantities of interest for nonlinear systems and Navier-Stokes equations. No demonstration calculations are given. The limitation of the theory to "small data" probably is similar to existence requirements; it may restrict the theory to Galerkin methods without stabilization (low Re) and avoidance of some pathological cases, but may not signal practical inapplicability. As usual, a strong theoretical foundation may be expected to lag methods which may nevertheless work. See also Babuska and Strouboulis (2001). Hay and Pelletier (2007, 2008) have had success with AAE using Wiberg error estimation (see Section 7.3.3 below).

7.3.1 Non-Conservation of Conservation Variables (Category D.1)

Category D.1, non-conservation of conservation variables, applies only to codes which do not use fully conservative algorithms for "conservation variables." For example, mass is not identically conserved in most old boundary layer codes, in many FEM codes, in codes based on collocation methods (including the very accurate wavelet methods, e.g. Vasilyev et al, 1995; Vasilyev and Paolucci, 1996) as well as other codes. The error estimate then involves numerical evaluation by quadrature for the erroneous loss or gain of mass in the computational domain. If the coded algorithm is consistent, this error will $\rightarrow 0$ but only in the limit of $\Delta \rightarrow 0$ for a non-trivial problem. (Even for a nominally mass-conserving full Navier-Stokes code, the satisfaction of mass conservation will usually depend on the degree of strict iteration convergence achieved, but this gives no indication of the discretization error of the solution, which is our interest here.) Momentum, vorticity, or internal energy are other possibilities. Note that the evaluation of this error depends on the accuracy of the quadrature, which probably should be consistent with the algorithm for solving the PDEs (see also discussion of D.4 below) but this does not appear to be a strict requirement. Conservative codes and algorithms are generally preferred, but non-conservative codes are seen to offer a readily evaluated and understood (although still surrogate) error measure.

In earlier Sections (e.g. 5.9.2) we noted that, if there is any suspicion that the grid resolution is not in the asymptotic range, three grid solutions are necessary to verify (or determine) the rate of convergence and thereby estimate the error, when the exact solution is not known. The qualifier used, "when the exact solution is not known," may seem redundant when the context is a realistic problem, but there is an important and useful distinction to be made in regard to Category D.1 and D.2 methods that are based on conservation errors. Although the conservation imbalance may not be of direct engineering or scientific relevance, it has the advantage that the exact solution of this quantity is known, namely zero. Thus, in these surrogate error measures based on conservation errors, one requires only a single grid solution to calculate the surrogate error, and only two grid solutions to extract the observed convergence rate p. The same is true if other functionals have known values, e.g. the drag coefficient = 0 for symmetric airfoils in inviscid flow, and can be used as an exact solution value to monitor convergence (Jameson and Martinelli, 1996).

7.3.2 Non-Conservation of Higher Moments (Category D.2)

Category D.2 involves evaluation of conservation errors for higher-order moments. In a typical turbulent (Reynolds-Averaged) Navier-Stokes code, fully conservative discretization may be used for the "primitive" variables of mass, momentum, and internal energy. However, turbulent kinetic energy is typically not identically conserved. An evaluation of its global and local conservation by quadrature, including carefully evaluated boundary inflow terms and dissipation (Haworth et al, 1993; Chang and Haworth, 1995, 1997), then gives a surrogate indication of general discretization errors. In Chang and Haworth (1995, 1997) it is also used to reliably guide local grid refinement for solution adaptive grid generation.

The fully detailed balance equations for linear momentum, angular momentum, mean flow kinetic energy, and turbulence kinetic energy have been presented by Haworth et al (1990, 1993). For clarity, following Chang and Haworth (1995, 1997), we restrict attention in this example to steady, laminar, incompressible flows, but this is not a limitation of the methodology. In laminar flow, the turbulence kinetic energy is of course zero, and we drop the distinction between mean and instantaneous quantities. The total kinetic energy in an arbitrary volume V is K.

$$K \equiv \int_V \frac{1}{2} \rho U_j U_j \, d\upsilon \tag{7.3.2.1}$$

The kinetic energy budget for an arbitrary volume V with bounding surface S can be written symbolically as

$$\dot{K} = FLUX + PRES + SHEAR + DISS \tag{7.3.2.2}$$

The terms on the right-hand-side represent the rate at which kinetic energy is advected out of V through S, (*FLUX*), the rate at which pressure forces on S extract kinetic energy from V, (*PRES*), the rate at which viscous stresses over S extract kinetic energy from V, (*SHEAR*), and the rate at which viscous stresses convert kinetic energy to sensible energy (heat) over the interior of the volume V, or viscous dissipation, (*DISS*).

In the FVM methods used by Haworth et al, mass and momentum are conserved at the cell level, but kinetic energy is not conserved. The left-hand-side $\dot{K}$ represents imbalance in kinetic energy resulting from discretization of the momentum equation. For any convergent discretization, $\dot{K}$ approaches zero in the limit as the grid spacing approaches zero, and departures from zero are suitable surrogate error measures.

The full evaluation of $\dot{K}$ for turbulent flow requires a fairly complex code development in itself, and the correlation with useful engineering measures of accuracy must be established by suites of computations

for any new class of problems. However, this approach has some significant advantages. As noted above in Section 7.3.1, in these surrogate error measure based on conservation errors, one requires only a single grid solution to calculate the error, and only two grid solutions to extract the observed convergence rate *p*. Also, Haworth (1993) has noted that, in his experience with turbulent combustion problems, when the grid refinement is sufficient to clearly exhibit asymptotic behavior, the numerical accuracy is already higher than he would like to pay for. Other CFD practitioners have noted the same experience in other turbulent flow problems; see previous discussion in Section 6.26, especially Blottner (1990) and Oberkampf et al (1995). While systematic grid convergence tests are reliable and self-contained (i.e., they do not depend on establishing correlations with some other accuracy study) they often do not become reliable (i.e., exhibit clear asymptotic performance) until the accuracy is somewhat excessive for engineering purposes. In Haworth's experience, the kinetic energy imbalance provides a more economical indicator.

7.3.3 Δ Zhu-Zienkiewicz and Wiberg Type Estimators (Category D.3)

Category D.3, the Zhu-Zienkiewicz type Estimators, were developed and intended primarily to for solution adaptive grid generation and arguably (Hay and Pelletier, 2007, 2008) are ideally suited for driving mesh adaptation. Zhu-Zienkiewicz (ZZ) estimators are not a single method, but a rather extensive family of methods.[92] Conceptually, they are post-processing or *a posteriori* methods, but practically they are built into the computational PDEs solver, so are not applied in a "black box" methodology as are the grid convergence methods. ZZ are in the category of recovery methods and involve post-processing of solution gradients.

Hugger (1997) gave the following general description of the "equilibrium" or "equilibration" method. "First postulate an error estimator (possibly justified by physical arguments); then prove that the estimated error is close (upper and/or lower bound in an *a priori* given norm, generally the energy norm) to the exact error under certain regularity assumptions." These methods allow the global energy norm to be well estimated (asymptotically exact for elliptic problems) and often give good evaluation of local errors (and provide the local estimate of stress accuracy, certainly important for structures problems). More relevant in the present context of the Quantification of Uncertainty, these estimators can be used as surrogate error estimators, and are often cheap compared to direct solution of FEM equations using inefficient solvers. However, when extended to FDM and FVM (or to efficient FEM solvers) their cost, when amortized only over the more efficient solver, is not insignificant (Pelletier, 1996). They still have the significant advantage compared to grid convergence studies of not requiring additional grid generation and multiple grid runs.

The ZZ family of estimators can be described in broad terms (Pelletier, 1996) by reference to the following 4 steps described for the simple non-conservation form of the steady-state linear advection-diffusion equation with a source term, this time written simply in 1-D for constant diffusivity.

$$V\frac{d\Phi}{dx} = \Gamma\frac{d^2\Phi}{dx^2} + q(x) \tag{7.3.3.1}$$

[92] See Zhu and Zienkiewicz (1990), Zienkiewicz and Zhu (1987, 1992), Wu et al (1990), Hetu and Pelletier (1992), Pelletier and Ilinca (1994,1997), Ilinca et al (1995), Pelletier et al (1995), Pelletier and Ignat (1995), and other FEM methods that have a similar flavor such as those of Strouboulis and Oden (1990), Oden et al (1993), Babuska et al (1994,1997), Ewing et al (1990), and Hugger (1997).

where V is the advection velocity, Φ is the concentration of the transported scalar, Γ is the diffusivity, and q is a source of Φ (per unit mass). Assume that the solution is obtained using linear elements, so that the continuum approximate solution Φ is obtained with linear interpolation between nodes.

Step 1. Perform a least-squares local projection of derivatives in the form $d\Phi^*/dx = ax + b$.

Step 2. Evaluate the projected derivatives at the nodes.

Step 3. Evaluate the error estimate on the elements.

$$\left\| e_i \right\|^2 = \int_{x(i)}^{x(i+1)} \left(\frac{dC^*}{dx} - \frac{dC}{dx} \right)^2 dx \tag{7.3.3.2}$$

Step 4. Evaluate the global error estimate by summing over the *NEL* elements.

$$\left\| e_{total} \right\|^2 = \sum_{k=1}^{NEL} \left\| e_k \right\|^2 \tag{7.3.3.3}$$

Each of these 4 steps allows a variety of implementations, leading to the extensive family of ZZ estimators.

Although developed for FEM, the Zhu-Zienkiewicz approach is adaptable to FDM and FVM (Pelletier, 1996) using the concepts discussed above in Section 7.2 on residual evaluation. Pelletier and Ignat (1995) have shown a good correlation of their ZZ estimator with the GCI (Chapter 4) for unstructured grid turbulent flow problems. ZZ methods naturally provide error estimates for the derivative of the solution, but sometimes can be structured to provide error estimates for flux or for the dependent variable itself (Pelletier, 1996).

The Zhu-Zienkiewicz type methods share the shortcoming of all surrogate indicators for fluid dynamics; other than guidance for grid adaptation, there is little inherent engineering or scientific interest in the error measure as defined. Therefore, unless the only interest is mathematics for its own sake, it is necessary to establish a correlation of the Zhu-Zienkiewicz indicators with an error measure of interest.[93] This can only be accomplished by expensive numerical experimentation for a given class of problems. For example, it would be unlikely that a correlation based on numerical experiments for internal combustion engine modeling would provide any guidance for external aerodynamics, nor even for a large range of flow parameters for geometrically similar problems. However, when one is involved in extensive suites of calculations, one may build up such correlations by experience, and arrive at a practical and relatively inexpensive single-grid error estimator. The ZZ family does not account for transport of errors; although Step 4 above gives a "global" error estimate, this evaluation is simply a summing of local error terms, and does not account for advection, diffusion, etc.

One can just as well use the various weight functions designed to guide solution adaptivity for single-grid estimators, e.g. Lee and Yeh (1994a,b). These are loosely related to Zhu-Zienkiewicz methods and

[93] This need for establishing a correlation in error measures is also true if one direct error measure (say, obtained by a grid convergence study) is to be used as a surrogate for another. For example, in an airfoil calculation, does a 1% error estimate on C_L insure a 1% (or 5%, etc.) error estimate for C_M? The correlation providing the (fuzzy) answer will only be accurate for a restricted range of parameters such as angle of attack, Reynolds number, Mach Number, and airfoil class.

carry the same caveats. But, as noted previously, this task of grid adaptation as typically practiced has little connection to the Quantification of Uncertainty for a final calculation with a useful error measure, and the success of these local "error" estimators in guiding grid adaptation must not be taken as demonstrations of their efficacy for the Quantification of Uncertainty. See discussion in Section 4.2.

The inherent limitation of the energy norm itself, the basis of Zhu-Zienkiewicz and similar error indicators, is demonstrated by the following counter-example (MacKinnon, 1996). Consider the following simple 1-D constant coefficient advection-diffusion-source equation, contrived to produce a quadratic solution.

$$-f_{xx} + cf_x = g, \quad 0 \le x \le 1 \tag{7.3.3.4}$$

$$f(0) = 0, \quad f(1) = 1 \tag{7.3.3.5}$$

$$g = -2 + 2cx \tag{7.3.3.6}$$

These produce the quadratic solution

$$f = x^2 \tag{7.3.3.7}$$

so that use of a 2nd-order discretization (say, centered differences with Dirichlet boundary conditions and the usual node-on-boundary configuration) produces the exact answer. In spite of the answer being exact, the commonly used energy norms are *not* zero. (They do, however, converge toward zero as the grid is refined.) By contrast, an error estimate based on grid convergence like Richardson Extrapolation will produce the correct zero error estimate. However, there is a disputed aspect of interpretation involved. Pelletier (1999) has pointed out that in the FEM interpretation (or philosophy) the discrete "solution" does not consist merely of the point values but of the continuous solution obtained by evaluating the basis functions everywhere. In this view, the piecewise linear FEM solution between the exact node values is not exact, so the non-zero energy norm is appropriate. We take the viewpoint that the discrete solution provided exists only at discrete points, a viewpoint supported by the common practice of presenting FEM results with interpolation based not on the FEM basis functions but on merely convenient interpolations, treating the solution discrete values as any other discrete data, e.g. experimental values. So when all node values are exact for all discretizations, yet the error measure shows non-zero values, something is amiss. The interpretation hinges on a long-standing difference of approach between FDM and FEM

Another AAE error estimator that has a unique property is the Wiberg estimator. Unlike ZZ estimators, the Wiberg estimator (Li and Wiberg, 1994) gives, with limited success, error estimates in quantities of direct engineering interest. Hay and Pelletier (2007, 2008) used it in an L_2 least-squares reconstruction, i.e. projecting the solution itself rather than its derivatives. They approximated the exact field over an element by a polynomial of 2 degrees higher than the FEM basis functions. Then a pointwise error estimated was computed as the difference between this L_2 reconstruction in each element and the FEM solution. They used ZZ to drive their very successful solution adaptive grids, but used Wiberg to estimate the solution error. For a 2-D RANS problem using wall functions, the Wiberg estimator appears to produce efficiency factors f very close to 1 as resolution increases (f => 1 indicates ordered or asymptotically exact error estimates). For a calculation without wall functions, the error estimate was accurate only to an order of magnitude (Hay and Pelletier, 2008) but in the context of the solution adaptive grid generation this has practical value because the computational PDE accuracy is so high. One could take the next higher decade for an error estimate that is not sharp but is conservative and often << experimental uncertainty.[94] In fact, the

[94] With wall functions, the Wiberg estimator gives $f \sim 1$ and the solution adaptive remeshing produces U_{num} = 0.027%, three orders of magnitude < experimental uncertainty (Hay and Pelletier, 2008).

distinction between error estimation and uncertainty estimation is not even important in such cases of very high accuracy.

7.3.4 Convergence of Higher Order Quadratures (Category D.4)

Category D.4 is the simplest, involving convergence of higher order quadratures. For example, quadrature (numerical integration of a known function) for pressure and shear force on the surface leads to evaluation of a drag coefficient C_D. For non-trivial problems (i.e., problems with significant solution structure), 2nd-order quadrature will give a different answer than 4th-order accurate quadrature. As the discretization refines, these two quadratures will converge to each other. For coarse grids, the difference may be used as a surrogate error estimator.

The same philosophy has been used point-wise. The simplest approach ultimately reduces to comparing the calculated pointwise value of some variable of interest with the value interpolated between the neighbors of the point. Indeed, these will converge as the solution converges, and therefore can (and have) been used as surrogate error estimates. However, the poverty of the concept is shown by the fact that the "error estimate" is obtained without recourse to the governing PDEs or the order of convergence. These "error estimators" would be better named simply "resolution indicators."

7.4 TIME ACCURACY ESTIMATION

Estimation of the numerical error of the time discretization can be performed in the same manner as the spatial errors, but it is also possible to use simpler methods. In fact, it is relatively straightforward to estimate the temporal error as the calculation evolves, and to build a code with a solution adaptive time step to control the temporal error to a predetermined level, even when only 1st-order time differencing is used.

Consider the following in expensive temporal error estimator (Oden et al, 1993; Roache, 1993a,b) for a fully implicit (backward) time differencing method. We write the backward time method for a general equation system in terms of an operator L (not necessarily linear) as

$$\frac{f_{i,j}^{n+1} - f_{i,j}^{n}}{\Delta t} = L^{n+1} \tag{7.4.1}$$

Generally, the solution would appear to involve an expensive matrix solution for all f^{n+1}. The inexpensive time error estimator uses the difference between backward and forward time integration, implemented as an extrapolation. The method is very cheap to implement because it does not require another implicit matrix solution, nor even another explicit stencil evaluation. It includes the effects of time-dependent boundary conditions and source terms.

In the current time step, Eq. (7.4.1) is advancing the solution for f from time level n to time level $(n + 1)$ with increment Δt using fully implicit (backward) time differencing, so that the operator L is being evaluated at $(n + 1)$. In the previous time step, the solution was advanced from $(n - 1)$, relative to the current indexing, to n with increment Δt_{OLD} and the L evaluated at n. We *could* explicitly evaluate L^n and make a separate, parallel estimate of the values f^{n+1} with an explicit step, as in

$$\frac{F_{i,j}^{n+1} - f_{i,j}^{n}}{\Delta t} = L^{n} \tag{7.4.2}$$

where F^{n+1} signifies the new value predicted by the explicit algorithm. (Since L^n involves values of f only at the location (i, j) but not at the neighboring locations $(i \pm 1, j)$ and $(i, j \pm 1)$ the equation is explicit, i.e., it does not involve a matrix solution of f at all values of i and j simultaneously.) The difference between the new value of f^{n+1} predicted by the implicit algorithm and F^{n+1} predicted by the explicit algorithm, both of which predictions are $O(\Delta t)$ accurate, is itself an error estimator of accuracy $O(\Delta t^2)$ for the time discretization error for that time step.

Explicit evaluation of L^n would not be expensive compared to the computer time necessary for the implicit matrix solution, but it does involve coding storage penalties and complexities (storage of old values of boundary conditions, source terms, etc.) An even more economical and elegant approach is to recognize that the L from the *previous implicit* step is identical to the L for the *present explicit* step. Thus, the L for the present explicit step can be evaluated from the knowledge of the previous change in f, requiring only the temporary storage of previous solution arrays. The explicit solution for f at $(n + 1)$, F^{n+1}, is then obtained by simple linear extrapolation of previous solutions. For constant Δt,

$$F_{i,j}^{n+1} = 2 f_{i,j}^{n} - f_{i,j}^{n-1} \tag{7.4.3}$$

For the more general case of variable Δt,

$$F_{i,j}^{n+1} = f_{i,j}^{n} + \frac{\Delta t}{\Delta t_{old}} (f_{i,j}^{n} - f_{i,j}^{n-1}) \tag{7.4.4}$$

The pointwise difference $F^{n+1} - f^{n+1}$ is then a pointwise temporal error estimator of $O(\Delta t^2)$.

Usually, a user would be interested in the maximum over the spatial domain of the percentage error, so one would evaluate the error estimator E_{ET} for the single time step as

$$E_{ET} = 100 \cdot \max_{i,j} \left(abs(F_{i,j}^{n+1} - f_{i,j}^{n+1}) \right) / f_{range} \tag{7.4.5}$$

E_{ET} is thus calculated as the % maximum deviation (or L^∞ norm) of the absolute value of the difference between the new f values f^{n+1}, predicted by the fully implicit algorithm, and F^{n+1}, predicted by the explicit algorithm, normalized by f_{RANGE}, which is the total range of f^{n+1}. In practice, ghost point evaluations with Dirichlet boundary conditions tend to exaggerate the error, so error estimates and f_{RANGE} should be calculated only over interior points.

Note that the explicit calculation is used only as an error estimator within a time step of an implicit method, not as the solution algorithm, i.e., its effects do not accumulate. Thus, stability limitations and/or conservation issues of explicit time stepping as a solution algorithm are irrelevant.

Although the extrapolation procedure is equivalent to explicit time-stepping, the extrapolation cannot be started until there are two time levels. Also, in the event that the initial conditions are set arbitrarily by the analyst (without setting initial conditions as a steady-state solution), it is likely that the initial conditions are incompatible with the boundary conditions applied at the first time step. This means that the change in boundary values during the first time step is virtually fixed, i.e., does not depend on the time resolution. Consequently, the first time step would not provide a meaningful estimate of $\partial f / \partial t$ and the error estimator would be inaccurate. Thus for this common situation, the time error estimation by extrapolation cannot be applied until the fourth time step.

E_{ET} can readily be used (internal to the code) as the basis for a solution-adaptive time-stepping algorithm, adjusting Δt so that the error estimate E_{ET} is acceptable, either by re-calculating the previous time step (preferred) or by simply adjusting the next time.

7.5 § UNCERTAINTY ESTIMATES FROM SINGLE GRID ERROR ESTIMATORS

The computational community will follow all these developments on single grid error estimators with interest, but a general point is that they all basically provide error estimates, but ultimately for Validation exercises we want the Calculation Verification to include error bars or uncertainty estimates, i.e. 95% certainty rather than the 50% (at best) intrinsic to error estimates. If any two ordered error estimators provide estimates for the same quantity, the same F_s should apply, at least in the asymptotic limit. This is especially clear for the *p*-methods, since RE itself is a multiple-grid *p*-estimator. If F_s has been empirically determined by grid convergence studies for (say) point values, it will not necessarily be accurate for (say) the energy norm estimates using ZZ estimators, but in lieu of empirical evidence from corresponding studies, the same F_s could be used provisionally. There is some evidence from limited studies (Pelletier and Roache, 2006; Hay and Pelletier, 2008) that $F_s = 1.25$ is adequate, and even more conservative than targeted for the very high accuracy achieved by solution adaptive remeshing. Again, in lieu of empirical evidence, the same F_s could be used provisionally for results from spectral and pseudo-spectral methods. However, the very high numerical accuracy achieved by spectral methods and by well-implemented solution adaptive methods often make the value of $F_s > 1$, and indeed the distinction between error and uncertainty estimations, unimportant relative to non-numerical uncertainties (experimental and parametric).

7.6 § COMPARISON OF GCI AND SINGLE-GRID UNCERTAINTY ESTIMATORS[95]

We first repeat a minor point to avoid confusion of terminology. As noted previously, the GCI is not an error estimator but an uncertainty estimator, equal to F_s times an error estimate. Thus, to compare likes, we must compare the GCI not to single-grid *a posteriori* error estimators but to those multiplied by a similar F_s. Setting aside this fine point, what are the pros and cons of the two approaches?

GCI (or more generally, a grid convergence study) is applicable to FDM and FVM as well as FEM, and involves such simple mathematics that the description given in Eq. (5.6.1) may be regarded as complete. AAE have been developed within the theoretical framework of FEM. Pelletier (1996) has extended the theory for ZZ to FVM, and other extensions of AAE methods to FVM and/or FDM may be possible, but at present they are not ready for "off the shelf" application. The detailed description changes with each variation of FEM. Although the evaluation is local, the cost may not be insignificant when amortized over the most efficient solvers.

GCI is the most reliable approach. While requiring no additional code, it does necessarily use multiple grids. ***If* one is taking a minimalist approach** to Calculation Verification, by assuming that the base grid is in the asymptotic range, then single grid AAE methods are much more convenient to use (once the algorithms are incorporated into the code). We still heartily recommend their inclusion in all commercial codes. However, at present they have not been conclusively demonstrated for quantities of engineering interest (such as heat transfer) in nonlinear problems. Until such theoretical approaches are demonstrated, one must establish correlations between the energy norm tolerances and those quantities of interest for a class of problems. This is a highly worthwhile area of research, because of the great convenience of

[95] From Pelletier and Roache (2006).

working with a single grid, especially for unstructured grids. By contrast, GCI applies to all quantities of interest.

We emphasize that we do not recommend either minimalist approach, i.e. one grid for AAE methods like ZZ, or two grids for GCI, unless justified by studies of nearby problems.

Also note that application of AAE to time-dependent problems is more difficult and is an open issue (i.e. requires additional theoretical work) at this time, whereas the GCI is straightforward. The GCI is usually applied in an approximate partitioned way by separately calculating a GCI for the temporal error. If this is reduced, by reducing Δt (perhaps using automatic adaptive time-step selection, to a level much smaller than the more difficult spatial errors) then the approximation is good. A more accurate way is to combine the temporal and spatial grid convergence. If both time and space discretization have the same order (e.g. $p = 2$), the formula for GCI is unchanged. If time is $p = 1$ and space is $p = 2$, the grid refinement ratios are changed accordingly, e.g. spatial grid doubling and time step quadrupling.

AAE lose accuracy near boundaries (Ainsworth and Oden, 2000; Pelletier and Trepanier, 1997) precisely where we often are most interested in the solution and the error bands. This is not a problem for GCI.

It is well recognized that singularities cause difficulties to AAE methods, through the mechanism of enhancing the non-localness of the errors, a phenomenon simply in keeping with behavior of continuum PDEs and referred to as "pollution errors" in the AAE literature (e.g., Ainsworth and Oden, 2000). (This behavior is clearly manifest in a grid convergence study, but only if more than 2 grids are used.) Strong nonlinearities are also blamed. It is perhaps less recognized that simple advection terms - not necessarily non-linear, nor even variable coefficient - are strong contributors to "pollution errors" simply because discretization errors themselves are advected and diffused.

Also, any stabilizing methods (e.g. flux limiters, SUPG FEM) destroy the theoretical basis for some AAE and degrade the actual performance as well (Ainsworth and Oden, 2000; Pelletier and Trepanier, 1997). The ZZ are immune because they do not rely on the PDE to construct the estimator.

If one is not taking a minimalist approach, but instead requires verification that the asymptotic range has been achieved, the advantages of the AAE are reduced. It is not possible to determine whether the grid is adequate (e.g., if convergence really is $p = 2$) by doing a single grid calculation. Order of convergence is verifiable only by multiple grid calculations. AAE methods still retain some advantages, in that they require one less grid than conventional GCI, at all levels. To be specific: for a minimalist approach assuming a known convergence rate p, GCI requires two grids, AAE requires one. However, each of these is dangerous, unless one is working on a suite of nearby problems so that one has confidence that one is working in the asymptotic range. To actually calculate an observed p, GCI requires 3 grids, AAE requires 2. To verify that p is constant, GCI requires at least 4, AAE at least 3. While it is simpler to generate 3 grids than 4, the same issues arise, i.e. the importance of strict grid similarity, noisy p, etc. (As noted in Sections 7.3.1-2, the exact answer for conservation checks - namely, zero - and observed p for the mass balance may be calculated from just two grids.)

The restrictions of the GCI (and all extrapolation-based estimators for uncertainties or errors) to calculations in which the grids are in (or close to) the asymptotic range, where the lowest order truncation terms dominate and observed $p \sim$ theoretical p, are well established, well known and often cited. What is seldom acknowledged is that all the single grid error estimators have the same restrictions. **AAE methods will not be accurate far outside the asymptotic range, and this condition is not detectable without multiple grid solutions.**

Finally, as noted in the introduction of this Chapter, the single grid error estimators are "code intrusive" whereas the GCI is applied externally.

7.7 § VERIFICATION WITHIN SOLUTION ADAPTATION

The powerful application of AAE occurs when they are used to drive solution adaptive grid generation. Here, the error estimate can arise without additional penalty. We are very much in favor of such methods. As a practical matter, the numerical error can be driven to small and surely acceptable levels. However, strictly speaking, the error estimate obtained by the adaptive AAE algorithm is not *always* an *ordered* error estimate of the final solution, as noted earlier. The ZZ methods in a solution adaptive grid sequence do provide such an ordered estimator. For non-ordered AAE methods, a quantitative error estimator can be obtained with systematic grid convergence (coarsening) of the final adapted grid, i.e. a separation of adaptivity and grid convergence. Using only the non-ordered AAE to guide solution adaptivity (and the truth is, almost anything intuitive works for guiding solution adaptivity) it is problematical to translate the adaptivity criterion into a reliable quantitative final error estimate, especially for functionals like Nusselt number and other "quantities of interest." However, if a correlation between the ordered AAE criteria and the results of grid convergence tests are established for a class of problems, one can proceed with confidence without requiring grid convergence testing separate from the solution adaptation for every problem.

As noted previously, the difference between any two solutions is at least qualitatively indicative of an error estimator. However, most of these are not quantifiable (and in fact most are undependable and grossly optimistic.) For example, a "feature-based" adaptation (e.g. increasing resolution in boundary layers or near shocks) is effective for improving accuracy but does not provide quantifiable error estimation. A proven approach is based on ZZ estimators.

The power of ZZ (and similar) single-grid error estimators is in fact exhibited not in a single-grid calculation, since this minimal approach can give no indication of observed order of convergence. The power of ZZ is most evident when combined with solution adaptation, which indeed was its original motivation. This procedure can produce quantified error estimation and therefore Verification, at least for global energy error norms. This approach uses ZZ but does not depend on the accuracy of ZZ for a single grid. These error estimates can be extended to uncertainty estimates via a factor of safety.

Significantly, numerical experiments consistently show that ZZ is ordered, or "asymptotically exact" (Turgeon et al, 2000). However, experience (Pelletier and Roache, 2006) demonstrates that the ZZ error estimator is not dependably conservative, as expected. E.g., Ilinca et al (1997) showed consistently *unconservative* estimates for a turbulent shear layer, Ignat et al (1998) showed consistently *conservative* estimates for turbulent flow over a heated backstep, and for a turbulent shear layer, consistently (except for the coarsest grid) *unconservative* for velocities but consistently *conservative* for turbulent diffusivities and temperatures. This lack of dependable conservatism is *not* a criticism, only an observation; the same is true for Richardson Extrapolation. But it does suggest the need for a factor of safety F_s applied to ZZ, whether used alone (in a single grid calculation) or within an adaptive grid simulation, to calculate an Uncertainty. In the solution adaptive work, the efficiency index [or affectivity index (Ilinca et al, 1997a), defined as the error estimate / true error] tends to unity asymptotically. (This is likewise true for Richardson Extrapolation.) The F_s determined by empirical correlations is more conservative asymptotically. However, at any particular resolution, some $F_s > 1$ is still necessary, no matter how accurate is the calculation. This is especially obvious when the ZZ estimator is always non-conservative in a grid sequence. Clearly this corresponds to an uncertainty worse than 50%, regardless of the accuracy. Note again that uncertainty and accuracy are distinct concepts.

The ZZ approach also allows error estimates to be made directly for *parameter* uncertainty values (Turgeon, and Pelletier, 2002). As might be expected, these have larger % errors than the primary quantities. Also note that the ZZ estimators are not as reliable near boundaries. This does not, of course, imply that the FEM itself is necessarily less accurate near boundaries, only that the dependability of the

error estimator is diminished near boundaries. More seriously, the ZZ inaccuracy near boundaries might lead to inadequate mesh adaptation there, and thus to diminished accuracy. This also occurs in hyperbolic problems of interface tracking where the local upwinding or other smoothing algorithms can misdirect the ZZ estimator into inadequate resolution. However, this shortcoming would not appear to be unique to ZZ adaptation.

Evaluation of an adequate factor of safety F_s for a solution adaptive mesh sequence appears to be more fuzzy than the GCI experience. Examination of these cited solution-adaptive remeshing studies shows that the particular adaptive grid strategy employed is so effective that the finest grid resolutions always correspond to a required $F_s < 1.25$. (All dependent variables contribute to the error, and adaptivity is based on the minimum over all variables of the mesh size predicted; the examples cited here use primarily 7-node triangular elements.) At the other extreme, when the coarse grids are also considered, results using an earlier version of the adaptive algorithm showed that the required F_s is sometimes above 3. Considering all the results, even disregarding mesh resolutions $\sqrt{N} < 20$ (i.e. roughly equivalent to a 20×20 mesh) $F_s = 1.25$ gives between 5 and 10% nonconservative estimates. The small sample and the restriction to one particularly effective solution adaptive method make determination of F_s, and perhaps the simple factor of safety approach, questionable. Until something better is developed, we still recommend $F_s = 1.25$ with the understanding that it is a rough value but that some $F_s > 1$ is generally required.

On the other hand, the solution adaptive ZZ approach offers another tempting simple error indicator: the difference ε between the last two meshes in the adaptive cycle. Although the work of Turgeon et al (2000) for a limited set of problems indicated that this was consistently conservative, it is certain that this ε would not be a reliably conservative estimator for just any adaptive scheme, e.g. feature adaptation such as available in many software packages. Even for this effective method, the conservativeness surely depends on the selection of the grid adaptivity level used, i.e. reducing the estimated error by a factor of $\varsigma = 2$ in each cycle by Turgeon et al (2000), or by ς between 0.4-0.8 in Hay and Pelletier (2008). This ε is definitely not an ordered error estimator in general, but behaves close to one for a combination of factors including selection of ς (Roache, 2008b). A smaller adaptivity factor would slow the convergence rate on successive grids, making the error estimator less conservative. Neither simple feature adaptation, nor redistribution refinement proportional to solution gradients or curvatures, would dependably give an ordered error estimator. But both theory and computational experiments indicate that the performance is not restricted to the 7-node triangular element formulation.

We strongly recommend such an adaptive verification approach, with the addition of a factor of safety, for steady state problems, provided that grid convergence is monitored to establish that the grids are in the asymptotic regime, at least for nearby problems. This approach certainly avoids the difficulties of multiple grid generation of systematically refined grids, especially when unstructured and/or multiblock grids are appropriate. Of course, this approach is applicable only to a specific class of algorithms, but once implemented, the process of (global) error estimation and Calculation Verification becomes relatively robust and painless.

7.8 § CONCLUDING REMARKS ON SINGLE-GRID ERROR ESTIMATORS

Although single-grid error estimators are not as reliable as classical grid convergence studies, they all have a significant advantage. Because they require only a single grid (if and only if it is assumed that the order of convergence is known), they can more easily be incorporated into codes for users other than the code developer, e.g. commercial codes. We strongly believe that some single-grid error estimator should be included in *any* commercial code, and that the error estimate output should *not* be an option, but an automatic hard-wired feature of all calculations (i.e. non-circumventable by the user).

Also, although our own experience and preference has always been for systematic grid convergence studies (e.g., Chapters 4, 5, 6), they too have their difficulties. As Haworth (1993) has pointed out to me, "The difficulties of grid generation and of ensuring that one lies within the radius of convergence cannot be dismissed lightly. Unstructured hex-cell grid generation remains a manually intensive process." For his work on internal combustion engines (Haworth et al, 1993; Chang and Haworth, 1995, 1997) the "minimum allowable mesh density is determined not only by accuracy considerations but also by the need to resolve geometric features" and by the "requirement that mesh integrity...be maintained as the grid deforms to accommodate valve and piston motion." Although automatic unstructured grid generation has been claimed in the FEM literature for decades, the experience of Haworth (and many others) remains that unstructured grid generation for such moving boundary problems requires weeks, even months. "Until robust auto-hex generation for arbitrary configurations is a reality, these applications are unlikely candidates for multiple solutions on multiple grids." [Of course, the time-step convergence test is cheap, and involves no problem of spatial grid generation.]

Mathur et al (2006) stated that for industrial CFD it is not unusual for geometry creation and mesh generation to take 50-80% of the overall solution time. This would not mean that this time is doubled if two grids are required, since the geometry creation and initial grid are the most difficult aspects. The ultimate solution may well be meshless methods, but these are in a relatively early stage of development and no systematic and reliable methods for Calculation Verification are available; see the review by Pepper (2006).

From an algorithmic viewpoint, *automatic* grid convergence testing is also possible. Some code developers would claim that this is already accomplished by solution-adaptive grid generation (of the *h* and/or *p* types) but, as already noted, the error upon which adaptive gridding is based is usually not a kind of measure of accuracy of interest to engineering or science code users. Automatic grid convergence testing is straightforward conceptually, but unfortunately the data structures of scientific programming languages make such codes difficult to develop and maintain.

Very impressive progress has been achieved in single-grid error estimators, and their use is highly recommended, especially in commercial codes. A non-fundamental but still real shortcoming is that most of the evaluations of single-grid error estimators have been small sample studies. (See Section 5.15.) The more fundamental limitations are (a) most single-grid error estimators are not applicable to quantities of direct scientific and engineering interest, although recent progress has been made here also, and (2) they require a single grid *only if the order of convergence is assumed to be known*. Experience with difficult problems such as RANS calculations shows that this assumption is not always justifiable. Multiple grid solutions are still required to determine the asymptotic range, which often involves considerable effort (e.g. Eça and Hoekstra, 2008). If the order of convergence is not known (or if the theoretical order is not trusted) these "single grid" methods are better described as "one-less grid methods", i.e. they require one less grid than classical gird convergence testing.

CHAPTER 8

HARD STORIES

This Chapter covers more features of Verification which can be troublesome. Examples are given of difficult problems that require special care, and can involve ambiguities, judgment calls, and subtleties.

8.1 FACTORS INFLUENCING CONVERGENCE RATES

The following discussion of a variety of factors that can influence the observed convergence rates in a discrete solution is taken from Westerink and Roache (1995) and Roache (1997). For the sake of continuity and completeness in this section, the following paragraph repeats the distinctions already given in Section 6.3.1 between formal, actual and observed convergence rates.

Formal convergence rates are always (by definition) assumed to be indicated by the leading order space and/or time truncation error terms. However, even in the asymptotic range (where discrete space and time steps tend to zero), formal convergence rates may never be achieved, leading to the definition of the *actual asymptotic* convergence rate. Even neglecting the well-recognized problems with computer round-off error, this *actual asymptotic* convergence rate may be different for a particular problem from the *formal* convergence rate for reasons to be discussed shortly. Finally, the convergence rate actually *observed* in numerical experiments may be different from *either* the *formal* or *actual asymptotic* convergence rates, simply due to grid resolution not being adequate to achieve the asymptotic range. It is not unusual for the fine grid Richardson Error Estimator to be less than the coarse error estimator for the same grid, indicating that the *observed* convergence rate is less than the *formal* rate.

8.1.1 Higher Order Truncation Term Competition

In establishing the formal convergence rate, it is assumed that the leading order truncation error dominates all other terms in the truncation series and hence the exponent of the discrete space or time step is taken as the order of the method. However, subsequent terms in the truncation series may be competitive with the leading order truncation term for several reasons. For nonlinear flows that cascade energy to the highest resolvable wavenumber range, the derivatives of the response function continue to steepen as we refine the grid. This leads to competition of the leading and subsequent order truncation terms as the grid is refined. *Even in the asymptotic limit*, the actual convergence rate will not necessarily match the formal convergence rate given by the leading order truncation term.

Another common cause of higher truncation term interaction occurs with relatively coarse discretizations. The resulting interaction of these error terms leads to non-asymptotic convergence behavior and typically causes the observed convergence rate to be less than the formal and/or actual convergence rates. We note that this situation often arises in unstructured graded finite element meshes which attempt to provide high grid resolution in regions where response functions (i.e., the solutions) vary rapidly (with corresponding high derivatives) and low grid resolution in regions where the response varies slowly (with corresponding low derivatives). Although the overall local error in the low grid resolution regions is modest due to the low derivatives of the response functions, the leading order spatial truncation error may be of the same magnitude as the next spatial truncation error, leading to differences between the observed and formal convergence rates.

Also, note that in general 2-D and 3-D problems, spatial cross derivative truncation terms arise that can profoundly affect the character and magnitude of the solution. These terms have been studied in the context of upwinding schemes for convection dominated transport equations but are in general not well studied or understood for complicated flow scenarios (e.g. see Cantekin and Westerink, 1990).

A minor point on fluid dynamics terminology is appropriate here. The term “convection-dominated flow” is often a misnomer. Consider the 1-D steady convection-diffusion (or advection-diffusion) model equation,

$$-u_{xx} + Re\, u_x = 0 \tag{8.1.1}$$

Such problems are usually described as “convection dominated” when the parameter range is such that *Re*, the Reynolds number (or more generally, the Peclet number) is large, $Re >> 1$, since this parameter multiplies the convection derivative. However, for this *steady* equation, convection *never* dominates diffusion. In fact, both convection and diffusion are everywhere in perfect balance, the convection term $Re\, u_x$ being exactly equal to the diffusion term u_{xx}. (If we tried to define the convection term as being u_x rather than $Re\, u_x$ we would find that the diffusion term u_{xx} is dominant, by the factor *Re*.) The more appropriate term is simply “high Reynolds number flow.”

8.1.2 The Effect of Space-Time Truncation Term Cancellation and Superconvergence

It is a common feature of discrete solutions to PDEs to encounter conditions in which various space-time truncation error terms cancel. For example in the Forward-Time-Centered-Space solution (see e.g. Roache, 1998b) to the 1-D constant coefficient diffusion equation, if

$$\Delta t = \frac{\Delta x^2}{6\mu} \tag{8.1.2.1}$$

where μ is the diffusion coefficient, then the leading order time and space truncation terms cancel each other, and the method becomes 2nd-order accurate in time and 4th-order accurate in space. Also, in the 1st-order upwind solution to a pure 1-D convection equation, the choice

$$\text{Courant Number } c = \frac{V\Delta t}{\Delta x} = 1 \tag{8.1.2.2}$$

leads to the cancellation of all space and time truncation errors and yields an exact solution at all nodes. Also, generalized wave-continuity equation (GWCE) solutions to the shallow water equations are subject to space-time truncation error cancellation as *c* ranges between 0.5 and 1.5 (Luettich and Westerink, 1994). This latter case corresponds to the phase error distribution changing from a dominant phase lead error to a phase lag error, passing through the zero phase lag axis.

These three examples point to the fact that space and time truncation errors certainly interact, modify or even eliminate each other. This cancellation of truncation terms can decrease *or increase* the effective convergence rate *p* depending on the problem and how the space/time-step ratio is maintained in the convergence study. Finally we note that mixed space time truncation errors (e.g., the Lax-Wendroff family) even further complicate the convergence behavior (Roache, 1998b). This category includes characteristics-based methods, e.g. Roache (1992).

8.1.3 Effect of Physical Parameter Resolution on Grid Convergence

As the level of grid resolution is increased, the level of physical parameter resolution can also change. The most common instance is geometry definition. As an example, consider a convergence study of tidal computations within the Western North Atlantic ocean using a shallow water equation based model (Westerink et al, 1994; Luettich and Westerink, 1994). The scale of definition of the bathymetry (i.e., the ocean floor) depends on the grid resolution. As the grid is refined, we can either interpolate the coarsely resolved bathymetry onto the finer grid or increase the level of resolution of bathymetry by re-interpolating actual available bathymetry data onto the finer grid. The results are not identical, of course. These types of grid studies indicate that under-resolution in the *grid* leads to *over*-predictions in the response, while under-resolution in the *bathymetry* leads to *under*-predictions in response. Thus, under-resolution in the grid and the inherent under-resolution in bathymetry lead to a partial cancellation of errors, which corresponds to a cancellation of truncation errors from the grid and bathymetric terms. Again, this type of truncation error cancellation can lead to observed convergence rates being higher than the formal convergence rates.

Similar questions will arise whenever geometry (or other parameter) resolution changes with grid refinement. For complex mechanical systems such as under-the-hood heat transfer in automobiles, coarse grid simulations will lump together some components into some idealized shape. As the grid is refined, the modeler must decide on whether to redefine the components with smaller scale features, or to simply increase the resolution of the idealized shape. This is a subtle modeling question.

A related and more fundamental question (repeated here from Section 5.10.9) arises when geostatistical methods are used to generate particular realizations of grid-block property variations with specified statistical parameters. Only the statistical results are of interest, not the solutions of the individual realizations. The question is then, should the grid refinement studies be performed separately from the geostatistical realizations? That is, should the solution of the partial differential equations be converged on finer grids with the assumed continuum property variation fixed at a geostatistically generated coarse-grid distribution, or should the geostatistical generation also change as the grid is refined? This is not a trivial question, and although definition of a fixed continuum problem for the grid refinement studies is conceptually easier, it is clear that substantial computer savings could accrue to the combined convergence

approach. In either case, the grid increments should be less than (be able to partially resolve) the correlation length of the property variation.

The question may arise as to where in the taxonomy should parameter uncertainty be included. Specifically, should parameter uncertainty be part of Verification or Validation? The answer is clear after a little thought, i.e. this is not a "gray area" at all. The answer is, in Validation. The determinant is the working definition of Validation, deciding whether or not one is "solving the right equations," which equations obviously *must* include the parameters of the equations. The exercise may look "gray" in that the calculational burden required to add the error bar of parameter uncertainty to model calculations yields an "error bar" similar to that of a grid convergence test. But it really is a "model" test rather than a "code" or "calculation" test, and therefore must be part of Validation.

Purely mathematical studies on the *sensitivity* of solutions to parameters are neither Verification nor Validation, but just results. (As already noted, these studies do require Verification of both Code and Calculation in order to be trustworthy.) Indeed, this type of study is one of the most powerful uses of simulation, performing "virtual experiments" that would be difficult or impossible to accomplish physically. Of many possible examples, an excellent one is the study of Oldenburg and Pruess (1995) on groundwater-brine systems showing the surprising sensitivity of gross flow patterns to molecular diffusivity. Another purely mathematical modeling exercise showed the effect of ratio of specific heats on the numerical behavior of compressible flow simulations as the incompressible limit is approached (Roache and Salari, 1990; see also Section 6.28).

8.1.4 Summary of Formal vs. Actual Asymptotic vs. Observed Convergence Rates

We have noted that *observed* convergence rates for complicated geophysical and other flow problems are often less and sometimes higher than the *formal* rates associated with a simple analysis of the leading truncation error terms, and have presented some of the possible origins of this problem. Some of these issues can be taken into account by carefully designing numerical experiments that define convergence. Whether grid size dependence is influencing the observed convergence rate for a problem can only be checked by going to a sufficiently fine level of resolution. This will produce the *actual asymptotic* convergence rate (i.e., the rate observed for asymptotically fine resolution without round-off errors) which may or may not equal the *formal* rate. Space-time truncation error dependence can be analyzed by performing a thorough study of the space-time error interaction found through Taylor and Fourier series analysis. It may be desirable to isolate the effects of the spatial errors by carefully keeping time steps small enough such that time errors do not interact with the space error, and by refining the time step with the space step in an appropriate fashion. (For some problems this would consist of linear variation of the time step with the spatial grid size, for others quadratic.)

However, it actually may be advantageous to work with time steps that incur space-time truncation error cancellation since they lead to overall lower errors for a given discretization even though convergence rates may not reach formal or actual asymptotic rates (depending how one sets up the convergence testing). For example, for solutions of the GWCE based shallow water equation, the best phase propagation properties are achieved with Courant numbers which lead to space-time truncation error cancellation (Luettich and Westerink, 1994).

Finally, parameter convergence can be separated out by freezing the parameter definition while refining the grid (Luettich and Westerink, 1994). We can then use the finest discretization and sequentially refine the level of resolution with which the parameters are defined.

Despite one's care in separating out these issues, *observed* convergence rates may still be less than *formal* or *actual asymptotic* convergence rates. In fact, it may not even be beneficial to try to separate them out when making an assessment of the overall error in an operational model. Therefore we should account

for these uncertainties when applying Richardson-based error estimators (or others). It is certainly justified to account for this type of problem in the GCI by including the safety factor $F_s > 1$.

8.1.5 Other Considerations in Defining Convergence Rates

Geophysical flows are very rich in their scale content and the temporal and spatial complexity of the associated flow features. Wave lengths in a geophysical flow range from the basin scale to the dissipation scale, which cannot be resolved except in the lowest Reynolds number computations. Furthermore, the features of a geophysical flow can be quite intricate. For example, a tidal flow includes local response minima for harmonic response constituents that include very high gradients in space in both amplitude and phase. (In fact, phase lines come together at the center of the amphidrome; Westerink et al, 1989.) The wavenumber range and complexity of the flow will make the task of assessing convergence and errors of such flow computations quite difficult. This observation also applies to DNS calculations of aerodynamic turbulence.

8.1.6 Defining Wavenumber Dependent Convergence

Due to the scale richness of a geophysical flow, it appears very reasonable to look at convergence on a wavenumber- or frequency-selective basis. This is routinely done in tidal studies where tidal frequencies are separated out and examined individually. As grid refinement increases, more wavenumbers and therefore more frequencies appear. We note that the effect of new frequencies will be reflected in other constituents as the convergence study proceeds. This process works well, although we must keep in mind that the accuracy of a specific constituent does not necessarily reflect the accuracy of all wavenumbers in the spectrum. Note that, for Verification of a Code, rather than Verification of a realistic Calculation, it is advisable to develop a model problem which uses the same governing PDEs but does not exhibit the strongly resolution-dependent flow features, as in Dietrich et al (1990). See also Hoekstra et al (2000b) for convergence study in wavenumber space.

8.1.7 Artificial Flow Features

As we examine any converging geophysical flow, we must carefully scrutinize the response functions as to their physical relevance. Specifically, not all flow features are converging functions which incur a certain relative percentage of error. In fact, artificial flow features can appear and disappear at a given wave number. Common artificial features are the $2\Delta x$ waves (see e.g. Roache, 1998b) which appear due to phase lag errors, folded dispersion relationships, and nonlinear cascading of energy into the higher wave number range where there may not be sufficient physical dissipation to eliminate this energy. The $2\Delta x$ waves are relatively simple to identify as non-physical flow features, since this energy moves to the highest resolvable wavenumbers as we refine the grid. However, there are other artificial flow features that can appear in the long-wave physically relevant portion of the response spectrum.

In particular, boundary placement and boundary condition implementation can influence the generation of artificial flow features. For example, in hurricane storm surge computations, placement of an “open ocean” boundary adjacent to a resonant basin such as the Gulf of Mexico can excite an artificial 18 hour period mode with significant amplitude (Blain et al, 1994). The generation of this artificial mode is very strongly dependent on where the open ocean boundary is placed and on how the boundary conditions are implemented. This again points to the importance of performing convergence studies on domain size (i.e. boundary placement), as well as boundary condition implementation and non-ordered approximations on the boundary in general. Analogous situations are common in aerodynamics and mechanical engineering

flows with outflow boundary conditions (Roache, 1994, 1998b). The key feature here is that these errors (perhaps better classified as modeling errors than numerical errors) *are not ordered* in grid increment and therefore will not be disclosed in grid convergence studies. However, as noted previously, these errors may prove in numerical experiments to be ordered in $1/l$ where l is the distance from the region of interest to the "open" boundary, and this error may be estimated (or banded, as in the GCI) by the same methods used for the grid convergence studies. (See Roache, 1994, and Section 6.10.)

8.1.8 L_2 and L_∞ Norms as Error Indicators; CAFE Curves

Different fields of endeavor have different requirements for the variables and functionals of interest. To generalize too much, engineering fields (mechanical, aerospace, chemical) often require functionals of the solution such as lift coefficient, heat transfer rates, mixing rate, etc. It is less common that practical interest is in accuracy of the entire field calculation; one example that comes to mind is aero-optical propagation (Truman and Lee, 1990). However, meteorological and ocean / river modelers are usually interested in accuracy of all the field variables. In these cases, it is meaningful to emphasize global accuracy of the entire computational field, and difficulty arises in attempting to characterize global accuracy with just one or a few numbers.

Typically, L_2 or L_∞ error norms are used as error indicators. However, questions arise whether these error norms are always reliable, and more significantly how they can be interpreted. In aerodynamics flows, L_∞ is a demanding but often practical measure, but not so in ocean modeling. For example, in a tidal convergence study, L_∞ norms are poor indicators of error due to the fact that shifting amphidromes, associated with very high gradients in response amplitude and phase, could cause extremely high localized errors, which do not converge smoothly or at the same rate as an L_2 norm. Furthermore, there is the general problem of interpreting how representative an L_2 or L_∞ norm is to the response as a whole.

The concept of Cumulative Area Fraction Error (CAFE) curves introduced by Luettich and Westerink (1994) presents domain errors in a more complete and meaningful way. These curves plot the fraction of the total domain that exceeds a particular error level (y-axis) against that error level (x-axis). Under- and over-prediction are indicated separately. Assuming that under- and over-prediction are distributed approximately evenly throughout the domain, median under- and over-prediction errors correspond approximately to a cumulative area fraction of 0.25.

As an example of the cumulative area fraction error curves, consider some results of a grid intercomparison for tidal flow computations as reported by Westerink et al (1995). Three grids were examined: Grid SS4, a regular 24255 node grid; Grid T1, a graded 11712 node grid with increasing resolution in shallower waters; and Grid T2, a graded 28889 node grid that is based on grid T1 with added resolution over steep bathymetric gradients. Results from all three grids are compared to those from a 95999 node regular grid. Figure 8.1.8.1 plots the three CAFE curves and clearly indicates that the grid T2 outperforms the other two grids over the entire distribution of error levels. Note that the error curves are continuous and that every error level in the domain is incorporated. Furthermore, extreme L_∞ errors correspond to the end points of the error curves. For a perfect solution the under- and over-prediction curves would collapse to a line on the zero error level. Finally, note that these CAFE curves provide a direct relationship between any error level and the fraction of the domain which exceeds that error level.

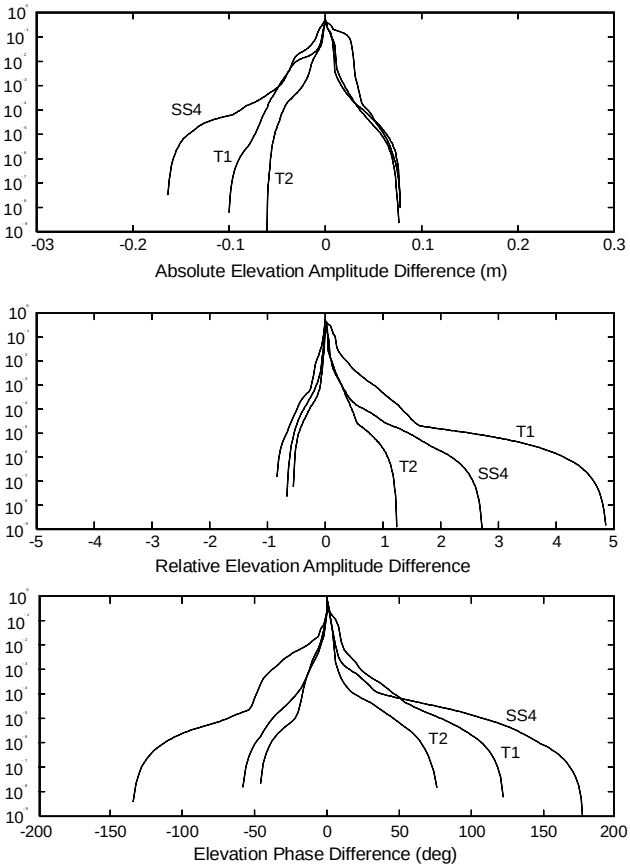


Figure 8.1.8.1. Example of Cumulative Area Fraction Error (CAFE) curves of Westerink et al (1995). (From Figure 2 of Westerink and Roache, 1995.)

8.1.9 Summary of Other Considerations in Defining Convergence Rates

We have noted that the flow spectrum must be carefully considered in any convergence study. Certainly looking at convergence behavior in a frequency or wavenumber specific way appears very reasonable. However, the overall spectrum should be considered as a whole and related to the level of convergence of a specific frequency. Furthermore, the response spectrum must be carefully scrutinized for artificial flow features such as $2\Delta x$ waves and resonant waves artificially set up through boundary placement or boundary condition implementation. The Cumulative Area Fraction Error (CAFE) distribution concept gives a complete and meaningful picture of the error distribution; the CAFE curves can be particularly useful in grid convergence studies and grid intercomparisons.

For other examples of grid convergence studies in geophysical ocean and lake modeling, see Johns et al (1983a,b), Le Provost and Vincent (1986), Bennett and Campbell (1987), Dietrich and Roache (1991), Dietrich et al (1990), Lardner and Song (1992), Piacsek and Allard (1993), Luettich and Westerink (1994), Westerink et al (1994, 1995).

8.2 BEHAVIOR OF QUASI-HIGHER-ORDER METHODS

Leonard and MacVean (1995) performed a grid convergence study that is valuable for covering different algorithms in a consistent framework, allowing comparison of error estimation across methods. The benchmark chosen was a commonly used 2-D unsteady meteorological problem consisting of an initial elliptical blob of cold air aloft that descends under gravity towards a ground plane. As it spreads laterally, a strong (nonlinear) density current is formed, with a sharp leading-edge front and a number of internal fronts associated with Kelvin-Helmholtz rotors. Although there is no exact solution available, numerous fine-grid Benchmark calculations have been reported previously; see Straka et al (1993). Figure 8.2.1 shows the initial conditions and Benchmark (fine grid) solutions for perturbation potential temperature θ.

The authors considered 7 different methods, 3 of which are flux-limited schemes. Only data on the 4 non-limited schemes were presented to allow evaluation of the error estimators. These 4 are as follows.

i. LF, a 2nd-order leapfrog method (e.g., see Roache, 1998b)
ii. UTOPIA, Leonard's nominally third-order flux-integral method (Leonard 1991)
iii. Q5, a quasi-5th-order scheme in which higher-order cross-differences are neglected and 2nd-order methods are used for diffusion (Leonard and MacVean, 1995)
iv. U5, a higher-order reference method (Leonard and MacVean, 1995)

The authors performed a thorough grid convergence study over 5 grids with 4 doublings, with mesh increments (in meters) of Δ = 400, 200, 100, 50, 25. Sample results for the 200 *m* mesh are shown in Figure 8.2.2.

Importantly, the convergence rates p (in the present notation) were determined experimentally. The authors' experience was that determining p values at every grid point was unworkable, leading to a wide range of different p values at different grid points "due in large part to slight phase shifts near strong gradient regions." They then used the global L_1 error norm,

$$L_1 = \frac{1}{N}\sum\left|\theta_{computed} - \theta_{reference}\right| \tag{8.2.1}$$

which normalizes the error over all N cells, calculated for all but the coarsest grid. $\theta_{reference}$ was calculated by Richardson Extrapolation using experimentally observed values of p (solved by graphical iteration). The L_1 error norm vs. mesh size is shown in Figure 8.2.3.

The "rather less expensive" Q5 results are graphically indistinguishable from those of the U5 reference method in Figure 8.2.3, but the error estimates will be distinguishable.

The relative error RelErr for each numerical solution was calculated from the reference solution. The relative error on grid h_1 was also estimated as E_1 calculated by the (generalized) Richardson Extrapolation (Equation 5.4.2, Chapter 5) using the observed values of p for each method and grid pair. Also, the GCI value using the original "factor of safety" = 3 was used, i.e.

$$\text{GCI} = 3E_1 \tag{8.2.2}$$

(Note the change in notation from Leonard and MacVean, 1995, whose "ε_1" corresponds to E_1 in the present notation.) The results are given in Table 8.2.1.

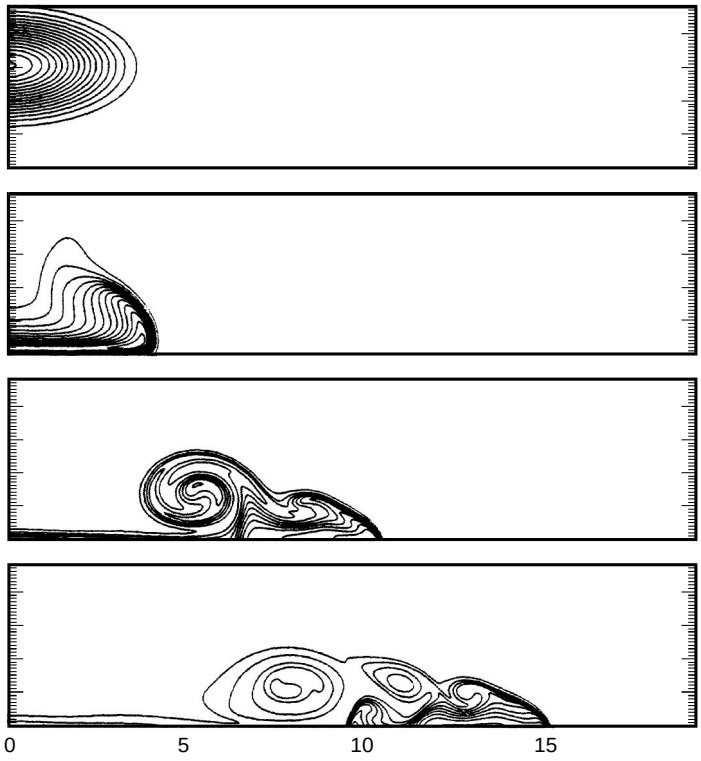


Figure 8.2.1. Initial conditions and Benchmark solutions for perturbation potential temperature θ for the descending blob of cold air. Solution times are 0, 300 *s*, 600 *s*, 900 *s*. (From Figure 1 of Leonard and MacVean, 1995.)

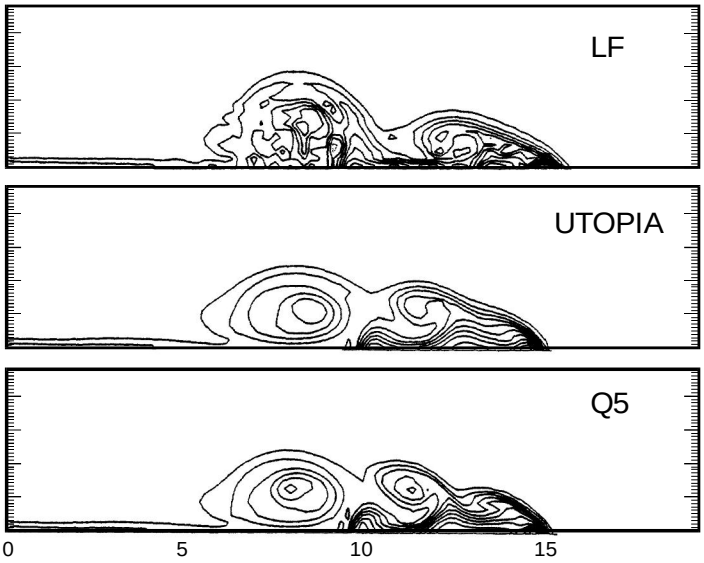


Figure 8.2.2. Results after 900 *s* on the 200 *m* mesh for the leapfrog, UTOPIA, and Quasi-5th-order method. (From Figure 6 of Leonard and MacVean, 1995.)

	LF	UTOPIA	Q5	U5
Δ = 25 m				
RelErr	0.30	–0.04	0.03	0.03
E_1	6.33	–0.05	0.03	0.03
GCI (F_s = 3)	19.00	–0.15	0.10	0.10
Δ = 50 m				
RelErr	20.30	–0.41	0.14	0.14
E_1	12.15	+1.40	0.92	0.95
GCI (F_s = 3)	36.44	+4.21	2.76	2.85
Δ = 100 m				
RelErr	67.71	+6.58	3.10	3.17
E_1	51.79	+2.66	4.90	4.83
GCI (F_s = 3)	155.36	+7.99	14.69	14.50
Δ = 200 m				
RelErr	101.07	+11.44	24.40	24.22
E_1	17.38	–13.20	7.48	3.87
GCI (F_s = 3)	52.15	–39.60	22.43	11.62

Table 8.2.1. Uncertainty diagnostics for $|\theta_{max}|$, in %. (From Table 1 of Leonard and MacVean, 1995.)

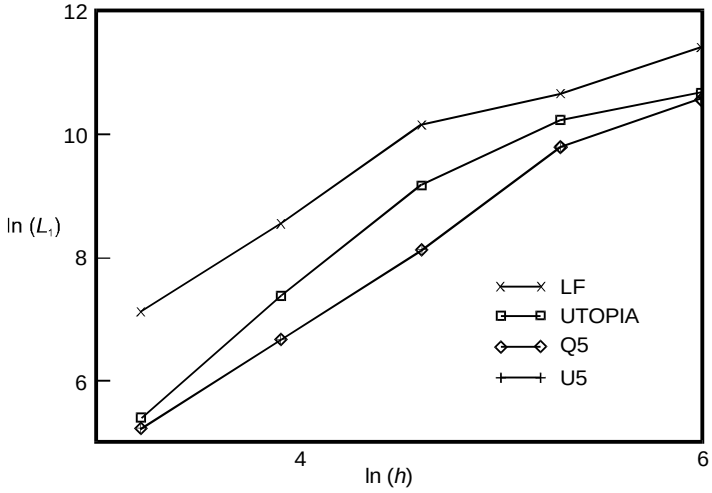


Figure 8.2.3. Error norm (L_1) vs. mesh size for leapfrog (LF), UTOPIA, Quasi-5th-order (Q5) and reference method (U5). (From Figure 4 of Leonard and MacVean, 1995.)

The following discussion of these results is taken verbatim (except for the change in notation noted above) from Leonard and MacVean (1995). "One sees immediately that simple uncertainty diagnostics are difficult to devise. In particular, Roache's GCI appears to be (usually) much too conservative. Our E_1 (= GCI / 3) appears to be somewhat more useful as a relative error estimator but not particularly accurate in absolute terms."

My own evaluation differs somewhat. First, I repeat that the GCI was not intended to provide an error estimate, but a conservative error band, equal to 3× the error estimate. Given that clarification, we still see that the results of Table 8.2.1 are still somewhat favorable to the philosophy of conservatism. There are 16 cases considered (4 methods, 4 fine grids). For 10 of these 16 cases, the GCI is very conservative and E_1 is either a good estimate or is itself very conservative, the incidences being strongly skewed to the finest grids, as expected. In 3 of the remaining 6 cases, the GCI is still conservative (less than the factor of 3) but E_1 is *not*. As anticipated and as experienced for the simple 1-D Burgers equation (Section 5.9.1) the generalized Richardson Estimator is *not* conservative. More surprising, even the GCI itself, with the full (original) "factor of safety" = 3, is not conservative in the remaining 3 out of 16 cases. As expected, these failures occur on the coarsest grid of 200 *m*. (Note that this is the coarsest *fine* grid in the coarse-fine pairs; i.e., the results in Table 8.2.1 for the 200 *m* grid were obtained from the pair of grid solutions on a coarse grid of 400m and a fine grid of 200 *m*.) Considering the noise evident in the top contour plot of Figure 8.2.2, one might argue that inspection would suggest that the LF method was not in the asymptotic range, so a CFD practitioner would know better than to expect the original GCI to be conservative. Based on folklore horror stories of the general competence level of commercial code users, I doubt this. With more certainty, I suggest that many sophisticated users would be fooled by the plausible results in Figure 8.2.2 for the Q5 method, yet even for this method the GCI with $F_s = 1$ is not conservative in Table 8.2.1. Furthermore, note that the conservatism of the calculations has been enhanced (compared to standard minimal practice) because the authors have experimentally determined observed values of p. The nominally 3rd-order UTOPIA method shows a convergence rate of $p = 2.83$. This is good performance for the method, since the nominal value was determined by analysis of the constant velocity problem, whereas the present benchmark advection field is far from uniform. But the point is that, if only the nominal result of the formal analysis $p = 3$ had been used, the UTOPIA result would be less conservative. In summary, in my evaluation, the thorough study of Leonard and MacVean makes a rather strong case for maintaining conservatism in the GCI error band, unless as originally suggested (Roache, 1994) three or more grid solutions are calculated to establish p.

Further evidence of the importance of experimentally determining p comes from the results of Leonard and MacVean (1995) for 3 flux-limited schemes:

i. the MUSCL scheme of Van Leer (1979 applied one-dimensionally, i.e. with the transverse gradient terms;
ii. LQ5, the earlier Q5 scheme with the first multidimensional limiter of Leonard et al (1995); and
iii. 1UP, a first-order upstream flux integral method for advection with second-order physical diffusion.

Nonlinear flux limiters present special difficulties for evaluation of observed orders of convergence. This wide class of algorithms was described in Roache (1998b), Chapter 5 (Section M) with an application in Chapter 17. Originally developed for shock flows, they are equally applicable to any solution that develops steep gradients or "fronts", steep being measured relative to the grid spacing Δ. They all involve non-ordered approximations applied adaptively (i.e. by monitoring the character of the developing solution) and *locally* to avoid the over- and under-shoots typical of shocks and high gradient regions. In such regions, the concept of "truncation error" becomes meaningless. The methods are typically developed in 1-D but applied in multi-dimensions, so the performance depends strongly on the angle between the velocity

vector and the grid lines, deteriorating as the angle $\rightarrow 45^o$. Thus observed order of convergence is (not surprisingly) strongly problem dependent.

The convergence behaviors from Leonard and MacVean (1995) are shown in Figure 8.2.4. One would expect LQ5 to have at best 2nd-order convergence, and less when the flux limiter is active (at the fronts), and the MUSCL scheme applied one-dimensionally to be somewhat less than 2nd-order. In fact, both exhibit a higher-than-theoretical value of $p = 2.4$, "rather intriguing but, as yet, unexplained." If $p = 2$ had been used in a two-grid application of the GCI, the conservatism would have been increased even beyond the safety factor of $F_s = 3$. On the other hand, the 1UP method of nominally 1st-order convergence might be expected to differ from $p = 1$, if at all, with a slight improvement (in the non-rigorously asymptotic range) because of the 2nd-order accuracy in the diffusion terms. In fact, the observed value is a miserable $p = 0.35$. (The well-known smearing of solution detail by 1st-order methods is shown in Figure 8.2.5.) The GCI denominator ($r^p - 1$) would have been calculated (with $p = 2$) as 1, whereas the value with observed $p = 0.35$ would be 6.25. The conservative $F_s = 3$ is overwhelmed by this non-conservative factor 1/6.25, so that *even the GCI is non-conservative* by more than a factor of 2. The simple E_1 estimator would be non-conservative by the factor of 6.25. This behavior would be disastrous for reporting ; imagine publishing an "uncertainty diagnostic" for (say) a lift coefficient of 4%, then being shown that your calculation was actually in error by 25%.

To reiterate my position (Chapter 5), based upon careful multiple grid resolution studies like those of Leonard and MacVean (1995) and others (see papers in Johnson and Hughes, 1995), I recommend that the GCI be calculated with $F_s = 1.25$ *only* for grid convergence studies based on three or more grids resulting in reasonable observed *p*. Recall that two grid solutions are necessary to estimate the error if *p* is already known, three grid solutions are necessary to determine the observed *p*, and *four* grid solutions are necessary to determine that the observed *p* is constant, i.e. that the sequence of solutions is within the asymptotic range. (Of course, if three grids are used and the observed $p \sim$ the theoretical value, e.g. observed $p \sim 2$ for a theoretically 2nd-order method, then one would have high confidence in using $p \sim 2$ without performing the fourth grid solution.) For the minimal two-grid solutions on a new problem (i.e., a problem for which *p* cannot be confidently inferred from a nearby problem), I recommend that the GCI should still be calculated with the usually (but not always) conservative value of $F_s = 3$. See also discussion in Chapter 5.

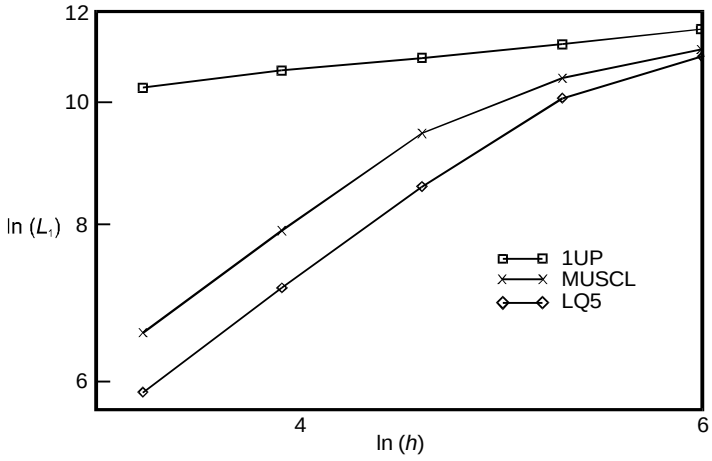


Figure 8.2.4. Error norm (L1) vs. mesh size for first-order (1UP), MUSCL, and limited quasi-5th-order (LQ5) methods. (From Figure 7 of Leonard and MacVean, 1995.)

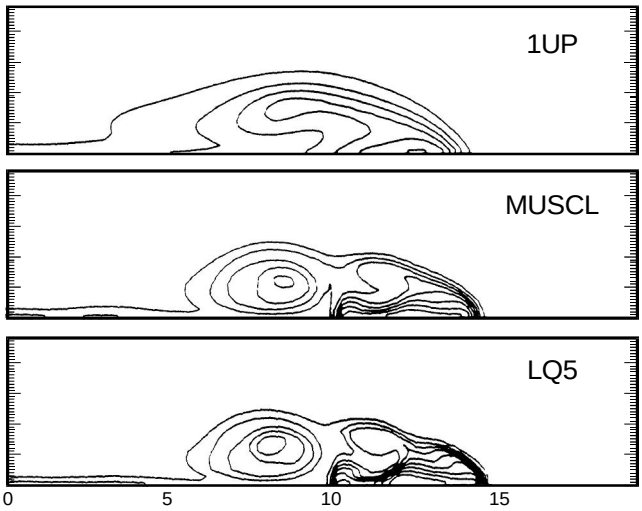


Figure 8.2.5. Results after 900 *s* on the 200 *m* mesh for first-order (1UP), MUSCL, and limited quasi-5th-order (LQ5) methods. (From Figure 9 of Leonard and MacVean, 1995.)

8.3 Δ SOME GOOD NEWS FOR TURBULENCE MODELING

As noted in Chapter 3, the error for mixed order methods should, strictly speaking, be banded using the lowest order of the method, i.e. 3rd-order differencing for advection terms used with 2nd-order differencing for other terms should be estimated using $p = 2$. Although this will be true in all cases *strictly* speaking, a *practical* and most advantageous exception *sometimes* exists for turbulence modeling.

In k-ε and k-ω turbulence models, it is a matter of common computational experience that the numerical accuracy of the transport equations for these variables does not have a major effect on flow average quantities of usual engineering interest, notably skin friction and pressure distribution. If 1st-order or hybrid differencing is used for the advection terms in the turbulence equations, the convergence rate for these equations will of course be $p = 1$, but the experimentally observed rate of convergence for flow average quantities, over realistic grid resolutions, is $p = 2$. (Salari, 1997; see e.g., Bergstrom and Gebart, 1997.) Bergstrom and Gebart (1997) experimentally showed $p \sim 2$ ($p = 2.08$) for flow average quantities over the finest three grids in a four-grid calculation, demonstrating that a practical asymptotic range had been attained, and was converging at 2nd-order accuracy. This is a significant Verification and confidence builder for both the algorithm and the commercial code used, CFDS-Flow3D (AEA Technology, 1994). (The code is now known as CFX.) The use of 1st-order differencing in the turbulence equations greatly stabilizes them (Ekaterinaris and Menter, 1994, Ekaterinaris, 1995) and is recommended, *provided only* that the user Verifies the experimental observed $p = 2$ for each significantly new (rather than "nearby") calculation.

Unfortunately, more recent experience casts doubt on this good news. Rumsey and Thomas (2008), using the NASA code FUN3D, found that the accuracy of the mean flow quantities may be reduced to first order. Whether these disparate experiences are due to different turbulence models, different algorithms, coding details, or the physical problem is not known at this time.

Other aspects of turbulence modeling bring bad news for calculation verification; see Section 8.14.

8.4 MYTH OF THE "CONVERGED SOLUTION"

Computational PDE practitioners, especially industrial engineers using general-purpose commercial codes, would like to be able to obtain, at least in principle, a "converged solution" (converged both iteratively and in time-space discretization) and then be able to confidently post-process that solution for any engineering measure of interest. Unfortunately, this cannot be possible in any general sense. The problem is again the *a priori* unknown correlation between possible error measures of engineering and scientific interest. Of course, expert judgment (based on experience with a nearby class of problems) can be valuable, but the problem is not soluble in any general sense.

For example, in our experience with dynamic stall and oscillating airfoil and wing calculations (Salari and Roache, 1990), lift was well converged at the finest discretization achieved ($141 \times 55 \times 55$ in 3-D, and 461×71 in 2-D), drag was more problematical, and moment definitely was not converged. If one had performed the grid convergence study (or other error estimation exercise) examining only the lift, one could only infer from engineering judgment (experience on related problems) whether or not resolution was adequate for drag and moment. Viegas et al (1988) noted the same variance of convergence with this additional distinction. Their calculations of lift and *pressure* drag were perhaps adequately converged (with a ~1% persistent oscillation) but moment (as above) and *friction* drag were not. If some new error measure became of interest, say position of transition, or location of trailing edge separation, or second harmonic component of the unsteady pressure coefficient (of interest for helicopter dynamic stall analysis; McCroskey, 1981), the convergence study would have to be repeated for this error measure, strictly speaking. Other examples are plentiful.

Rosenfeld (1994) used grid resolution studies of bluff-body wakes with resolution up to 513×513 points, and examined convergence in both the physical and Fourier domains. He showed that very fine meshes were required to obtain convergence in the physical domain, but that Fourier components converge on coarser grids faster than either the phase angle or the solution in the physical domain. A major factor in this behavior is the resolution required for the initial transients, which do effect the phase angle of the solutions, but are of no importance (physically or numerically) to the fully developed periodic solution. "These findings indicate that the mesh resolution has a more pronounced effect on the phase velocity of the vortices than on phenomena related to magnitude," so that coarser grids may be acceptable for prediction of force coefficients.

Blottner (1990) noted that hypersonic sphere-cones calculations require much higher grid resolution for heat-flux calculations than for surface pressure calculations with the same level of uncertainty.

Sengupta et al (1995) showed that calculations of the branching between symmetric and non-symmetric flow patterns require special attention to grid resolution.

Rumsey et al (1996) calculated unsteady laminar flow over a circular cylinder, and unsteady turbulent flow over an 18% thick circular arc airfoil. They explored the sensitivity of grid convergence to many physical and numerical parameters, including turbulence model (algebraic vs. one-equation), implicit vs. explicit time differencing, and various time-step sub-iteration schemes. They noted that for some conditions, "even results on the finest 369×129 grid are not completely code converged," yet [as is well known] the "oscillation frequency is relatively insensitive to grid density for both turbulence models."

Certainly, it is always possible to devise some error measure that is exquisitely sensitive to discretization error, e.g., some high-order statistical correlations in turbulent flow calculations, so that this measure is far from converged even when other, more benign measures are well converged. A practical example is a boundary layer stability calculation. For most engineering applications, skin friction and wall heat transfer are of principal interest, and these are sensitive to the first normal derivative of velocity at the wall. However, it is known from classical stability theory (e.g., Lin, 1967) that laminar boundary layer stability is dependent on the diffusion of vorticity across the critical layer (the y-position at which the mean flow speed equals the disturbance wave speed). The principal component of boundary layer vorticity magnitude is $\partial u / \partial y$, so this means that the appropriate measure of accuracy is of the term $\partial^3 u / \partial y^3$, which may be expected to be more difficult to "converge" than simply the velocity component u. Conversely, accurate convergence of pressure may be unnecessary.

Thus, the concept of a "converged solution," ascertained to be so *independent of the intended error measure*, is a myth.

8.5 ESOTERIC CODING MISTAKES

As noted in Chapter 3, general PDE codes (more general than the simple Poisson equation in nonorthogonal coordinates) would be difficult to include in a theorem on code Verification because of esoteric errors. The difficult aspects of the codes are not *algebraic* complexity; in Steinberg and Roache (1985) we convincingly verified 1800 lines of dense Fortran. The more difficult and vexing problems come from totally distinct types of complexity, arising from option combinations and switches like conditional differencing. Esoteric errors can arise because of nonlinear flux limiters like FCT, TVD, hybrid or type-dependent differencing, etc. This does not invalidate the claim that the Method of Manufactured Solutions (Chapter 3) Verifies the Code. Obviously, the method cannot Verify a feature that is not turned on! This merely says the obvious, that in order to test a code with many features, one may require many test runs to turn on these features.

A type of error unlikely to be included in an accuracy Verification theorem arose in a code for calculating supersonic base pressure (Mueller et al, 1970; Mueller and Roache, 1973) using the Chapman-Korst flow model. Several Fortran variables were defined for groups of terms involving Mach number M, such as MS = M**2, M1 = M – 1.0, M2 = SQRT (MS – 1.0), etc. One of these variables was not declared REAL. The original code was initially exercised and considered to be bug-free for Mach numbers of 1.5, 2, 3, 4, 5, and 6, with excellent agreement with experiment. It was of course correct for integer M only, and quite good for M = 1.5, leading to a false conclusion on accuracy. For M = 1.3, it failed unambiguously, leading to a coding correction.

In the first example in Section 1.3 (from Roache, 1982), a factor of 2 error in cross derivatives in a grid generation code did not show up during extensive testing. The test suite was composed of homogeneous elliptic grid generating equations that produce almost conformal transformations (with discrepancies due only to boundary distributions) for which the cross derivative terms vanish in the limit of $\Delta \rightarrow 0$. Although revealed by the general code accuracy Verification procedure, as described in Chapter 3, one can easily envision being misled by particular problems.

8.6 FALSE VERIFICATION TEST OF A PARTICLE TRACKER

The 2-D and 3-D SECO_TRACKER codes (Roache, 1993a,b) were built to provide temporal high-order particle tracking. They use a modified version of a 5th-order Runge-Kutta-Fehlberg ODE integrator called RKF45 (Shampine et al, 1976) for integration in time combined with linear velocity interpolation in

space and time to produce particle tracking with Verified accuracy of O[$(\Delta x)^2$, $(\Delta t)^5$]. This may appear to be inconsistent with the typical 2nd-order flow field accuracy, but is in fact justifiable. The key concept is that the flow codes are Eulerian, whereas the particle tracking is intrinsically Lagrangian. For a steady Eulerian velocity field with linear variation in space, there is *zero* time truncation error in the velocity field, yet the (Lagrangian) solution for particle position involves exponential functions in time. The higher time variability of particle position, compared to Eulerian velocity fields, justifies the use of higher accuracy tracking algorithms. See also Baptista (1987).

In Roache et al (1990), we described the following "hard story" in the debugging experience with SECO_TRACKER.

8.6.1 Spatial Convergence of Tracker Codes

The first benchmark test case for the Tracker codes was designed to verify spatial convergence. The benchmark solution is a 1-D trajectory. The 1-D velocity field varies in space and time as

$$| V |= 0.5\frac{\cos(time)}{\cos(x)} \tag{8.6.1.1}$$

The 1-D solution is

$$x\_part = a\sin\left(0.5\cdot\sin(time)\right) \tag{8.6.1.2}$$

which solution has sufficient structure to exercise all terms in the Taylor series expansion of the discretization error. The solution is rotated with the direction cosine $dcos1 \le 1$, and $dcos2 = \sqrt{1-dcos1^2}$. Using $dcos1 = 0.5$ gives the trajectory angle $\alpha_1 = 60°$, so the trajectory is not through the grid corners (to assure that interpolation is exercised). The test results are shown in Table 8.6.1.1.

The term *relerr_p* is the relative error in final position, and *relerr_coef* is calculated as *relerr_p*·$(il - 1)^2$. For a method which is 2nd-order accurate in space, this coefficient should become asymptotically constant as *il* is increased. The time discretization error is O(Δt^5) and is negligible compared to the spatial discretization error for this problem.

As seen in Table 8.6.1.1, this coefficient is indeed roughly constant. (The slight decrease in *relerr_coef* as the grid is refined would theoretically indicate faster than 2nd-order convergence, but this distinction is not significant, and is evidence of less than asymptotic behavior, round-off error, and time discretization error.)

The successive ratios of relative error in particle position are displayed in the last column of Table 8.6.1.1. The ratio of successive errors as the spatial step is halved, from $(il - 1) \equiv N = 4, 8, 16$, is better than the theoretical value for the 2nd-order method, $2^2 = 4$. This test of 2nd-order spatial discretization error is not as sensitive to computer round-off error as is the 5th-order time discretization test. The better-than-theoretical performance will not generally hold for arbitrary velocity fields. The failure in Table 8.6.1.1 at $N = 32$ is indicative of accumulated round-off error using the single precision version of the code on an old Vax computer with about 7–8 significant figures of accuracy for floating point calculations. (The single precision version was developed for later conversion to the Cray-XMP, for which any use of double precision variables prevented vectorizing.)

The above test results were obtained for collocated grids (MAC = 0); similar results were obtained for a staggered (Marker-and-Cell or Arakawa-C grid; MAC = 1) in Table 8.6.1.2, and for a stretched grid in Table 8.6.1.3.

il × *jl*	*relerr_p* (%)	*relerr_coef*	error ratio
5 × 5	2.88 E+00	4.61 E–01	–
9 × 9	5.74 E–01	3.68 E–01	5.0
17 × 17	6.97 E–02	1.78 E–01	8.2
33 × 33	5.06 E–02	5.18 E–01	1.4

TABLE 8.6.1.1 Two-D Tracker Results For Test #1, Uniform Grid, MAC = 0.
(From Table 8 of Roache et al, 1990.)

il × *jl*	*relerr_p* (%)	*relerr_coef*
5 × 5	1.6238 E+00	2.5981 E–01
9 × 9	4.0202 E–01	2.5729 E–01
17 × 17	5.3066 E–02	1.3585 E–01
33 × 33	5.1116 E–02	5.2343 E–01

Table 8.6.1.2. Two-D Tracker Results For Test #1, Uniform Grid, MAC = 1.
(From Table 9 of Roache et al, 1990.)

il × *jl*	*relerr_p* (%)	*relerr_coef*
5 × 5	4.4230 E+00	7.0768 E–01
9 × 9	7.6609 E–01	4.9030 E–01
17 × 17	8.4578 E–02	2.1652 E–01
33 × 33	4.7120 E–02	4.8250 E–01

Table 8.6.1.3. Two-D Tracker Results For Test #1, Stretched Grid, MAC = 0.
(From Table 10 of Roache et al, 1990.)

8.6.2 Temporal Convergence of Tracker Codes: A False Negative Test

The second benchmark test case for the particle tracker codes was designed to verify temporal convergence. A solid-body rotation velocity field with rotational speed = 2π radians/unit time should return the particle to its original position at final time = 1.

N	error in x_part	error ratio
10	4.9E–02	–
20	1.7E–03	28.8
40	5.5E–05	30.9
80	4.0E–05	1.4

Table 8.6.2.1. Two-D Tracker Results For Test #2, Uniform Grid, MAC = 0.
(From Table 11 of Roache et al, 1990.)

For an initial position at *x_part* = (0.75, 0.0) the errors in the final *x*-position at *N* time steps were obtained, as shown in Table 8.6.2.1.

The ratio of successive errors as the time step is halved, from ($il - 1$) ≡ N = 10, 20, 40, very closely fits the theoretical value for the 5-th order method, $2^5 = 32$. The failure from N = 40 to 80 is again indicative of accumulated round-off error using the single precision version of RKF45 on an old Vax computer with about 7–8 significant figures of accuracy for floating point calculations.

Although this benchmark test case #2 does verify the 5th-order time discretization error, there is no spatial interpolation error, since the velocity field for solid body rotation is linear in space.

This is an important distinction. In actual application, a *serious* coding mistake was not detected by this seemingly convincing benchmark test case, i.e. we obtained a "false negative" error test. The code logic which located the cell indexes containing the particle position was in error. However, since the solid body rotation gives $u = -cy$ and $v = +cx$, i.e. a linear variation in velocity components, the linear interpolation to find $u(x, y, t)$ and $v(x, y, t)$ is algebraically exact, no matter what cell is used as the basis for the interpolation. (The wrong cell merely results in linear *extrapolation*, which is still algebraically exact.) Only when the previous benchmark test case #1, involving the (non-trivial) rotated 1-D trajectory, was exercised was the coding error discovered.

Similar results were obtained from the 3-D tracker codes. The benchmark test case was the 3-D analog of the first 2-D problem above. The same 1-D solution was rotated with direction cosines $dcos1$, $2 \le 1$, and $dcos3 = (1 - dcos1^2 - dcos2^2)^{1/2}$. Using $dcos1 = 0.5$ and $dcos2 = 0.6$ gives $\alpha_1 = 60°$ and $\alpha_2 = 53.13°$, so the trajectory is not through the grid corners. Results are shown in Tables 8.6.2.2 and 8.6.2.3.

Particle trajectory calculations can be quirky. Some FEM codes provide discontinuous flow velocity solutions that can result in peculiar and totally erroneous qualitative behavior of particle tracking. If the discontinuous velocity solutions are used without smoothing, they can result in trajectory calculations with particle "trapping" at an element interface; with smoothing, they can result in "a peculiar 'kick-back' on the trajectory close to a fracture zone". See Figure 3.23 of OECD (1988).

8.7 INADEQUACY OF SINGLE GRID CALCULATIONS FOR PARAMETER TRENDS

Considerable effort has been invested in developing solution algorithms which always give smooth, reasonable looking solutions at coarse grid resolution, with the idea that even under-resolved solutions could provide some useful engineering information, e.g. trends of some flow functional with parameters. Even the hypercritical Oreskes et al (1994) and pessimistic Konikow (1992) (see discussion in Chapter 2), while claiming that Validation and Verification are impossible, stated that computational simulations can provide reliable information on trends.

il = jl = kl	*relerr_p* (%)	*relerr_coef =*
5	2.8987 E+00	4.6379 E–01
9	5.2918 E–01	3.3868 E–01
17	6.5718 E–02	1.6824 E–01
33	5.0346 E–02	5.1555 E–01

Table 8.6.2.2. Three-D Tracker Results For Test #1, Uniform Grid, MAC = 0.
(From Table 12 of Roache et al, 1990.)

N	*relerr_p*	*N* error ratio
4	2.8786	
8	0.5292	5.4
16	0.0667	7.9
32	0.0503	1.3

Table 8.6.2.3. Three-D Tracker Results For Test #2, Uniform Grid, MAC = 0.
(From Table 13 of Roache et al, 1990.)

In fact, nothing can be taken for granted in this respect. In a classic paper for the Finite Element community entitled "Don't Suppress the Wiggles - They Are Trying to Tell You Something," Gresho and Lee (1981) warned against the design of algorithms which provide smooth solutions at coarse resolutions. The most common such method is simple upstream differencing for advection, which tends to obliterate small scale flow features, but there are more subtle methods with more subtle qualitative errors. Brown and Minion (1995) gave examples of misleading behavior of a Godunov projection method for incompressible Navier-Stokes equations. The method always produced "smooth, apparently physical solutions" even when under-resolved. However, closer examination indicates the presence of "spurious nonphysical vortices that are artifacts of the under-resolution." That is, rather than *obliterate* small-scale features as simple upstream differencing does, this method *introduces* false small-scale features. The authors noted that "these artifacts are not unique to Godunov methods" but "are observed with other difference approximations as well." Their lesson is that "it is not sufficient that the computed solutions appear smooth and well-resolved," and they noted again the observation of Gresho and Lee (1981) and many others, that the *obvious* failure of simpler methods is less dangerous. "Since the centered methods fail rather badly in the under-resolved case, it is somewhat easier to know when one is properly resolving the computed solutions for those methods." On the other hand, one prefers smooth solutions at all levels of grid resolution to guide solution adaptive grid generation, so we should not eliminate these methods entirely.

Even the use of 2nd-order accurate discretization does not guarantee correct parameter trends for under-resolved grids. In a very early pioneering paper, Burggraf (1966) computed the now-common Benchmark problem of steady incompressible Navier-Stokes flow in a driven cavity using 2nd-order centered differences. At the low Reynolds numbers, his solutions were smooth and realistic looking, clearly showing not only the principal vortex but also corner eddies, even for coarse resolutions of 20×20 and 40×40 cells. Burggraf noted the trend of movement of the position of these corner eddies with change in Reynolds number. This is surely a simple trend, and one that might be expected to be reliably predicted with grids so slightly under-resolved that the qualitatively correct flow features were present. In fact,

Burggraf was not misled, noting that the *opposite trend* of eddy position with Reynolds number was obtained with the 20×20 grid compared to the 40×40 grid.

Trends and sensitivity calculations (derivatives of flow solution variables or functionals with respect to parameters that determine the flow) are complicated by discontinuities such as shock waves and even the end points of (continuous) rarefaction waves. See Appel and Gunzburger (1997) for a taxonomy of sensitivity calculation approaches and a clear exposition of difficulties with discontinuous solutions, including numerical algorithmic issues and grid refinement.

These examples do not indicate a hopeless situation. They simply indicate the need for grid resolution studies, no matter what the numerical method used, and no matter if the first grid solution looks plausible. The danger is in the commonly expressed opinion that qualitative accuracy of parameter trends can be taken for granted, which is nothing more than *wishful thinking*.

Furthermore, for high-consequence applications (e.g. nuclear power industry) the code users cannot shirk their responsibility to verify the codes they are using if the code vendors have not done a convincing job (i.e. both thorough and thoroughly documented). The standards of accuracy may well be less than for a final design code, but V&V is still required. It is apparently true that the following statement must be made.

An erroneous code can predict erroneous trends;
an inaccurate computation can predict inaccurate trends.

8.8 HARD-WIRED DATA VS. USER INPUT DATA

Data on physical properties is a source of error in simulations, and can cloud the distinctions between Verification of Codes vs. Verification of Calculations, between Verification and Validation, and between conceptual model errors and code errors.

For example, consider thermodynamic properties for individual chemical species, as discussed by Oberkampf et al (1995). Experimental errors in input values should be considered in Validation, clearly. However, the variation of properties with temperature will be accounted by table look-up and interpolation, which introduces additional error and could arguably be considered a numerical error of the simulation, or a numerical error of the experimental data reduction, depending on who supplies the interpolation algorithm. (Oberkampf et al noted that for "table look-up/interpolation or curve fits...surprisingly, for many cases in the past, the approaches resulted in large errors.") Likewise, if the data and interpolation algorithm are hard-wired in the code, the issue is Verification and Validation of the code, but if it supplied by the user, it is a conceptual modeling issue of Validation. Code QA (Quality Assurance) issues (see Chapter 12) are also clouded if the code has options for *both* hard-wired data (which may have gone through Certification/QA) *and* for user over-ride with other input data. This code flexibility is obviously a desirable feature for the engineering analyst, yet it causes genuine problems for engineering management. The code Certification/QA process cannot cover undefined user input, and the very meaning of Certification/QA becomes questionable. For an analyst to claim use of a Certified or Quality Assured code may give misleading confidence if the user has over-ridden the hard-wired property data that was an inherent part of the Certification/QA process.

8.9 DEGRADED RATE OF CONVERGENCE DUE TO USER MODELING ERRORS

Because of its importance, we again describe the example problem from Section 6.12, a groundwater contaminant transport calculation. The point in Chapter 3 was the remarkable sensitivity of systematic grid convergence testing; the point herein is that the code itself was proven to be 2nd-order accurate for this type

of problem, yet a subtle modeling error by the *user* (i.e. not a code error) during the grid convergence testing degraded the rate of convergence to 1st order.

In the groundwater contaminant transport calculations (advection-diffusion + decay, retardation, and matrix diffusion), use of a plausible single-grid-block representation for a point source as the grid is refined introduces error in a finite volume formulation. In this cell configuration, the cell faces align with the boundaries of the computational domain, and doubling the number of cells requires the *location* of the single cell representing the source to *shift by* $\Delta/2$. It is to be expected that the solution accuracy in the neighborhood of the source would be affected, but surprisingly, the accuracy of time-integrated *discharge* across boundaries far from the source was also degraded to 1st-order accuracy. See Section 6.12 for details.

8.10 Δ LESSONS FROM NONLINEAR DYNAMICS

H. Yee and colleagues (Yee et al, 1991; Lafon and Yee, 1992; Yee and Sweby, 1996) gave many examples of aerodynamics solutions involving "nonlinear dynamics" (see additional literature citations in Yee and Sweby, 1996). Spurious (grossly erroneous) numerical solutions for steady state problems with strongly nonlinear source terms can be obtained for stable implicit algorithms applied beyond the linear stability limit. The solutions, and the strange and chaotic convergence behavior, are interesting and valuable to explain iteration convergence behavior relative to initial conditions through representation of *basins of attraction*. These are maps of initial condition parameters which lead to iteration convergence; an example is shown in Figure 8.10.1. The system being solved is for a nonequilibrium flow field relaxation of the 1-D Euler equations for a (N_2, N) mixture, representative of shock tube experiments or hypersonic wind tunnels. The model is expressed as the following single ODE (plus two algebraic equations for ρ and T).

$$\frac{dz}{dx} = S(\rho, T, z) \tag{8.10.1}$$

where z is the mass fraction of the N_2 species, ρ is the density of the mixture, and T is the temperature.

Figure 8.10.1 shows the experimentally determined basins of attraction for this system using various numerical methods. In each of these plots, the shaded region denotes the basin of attraction, in which combinations of initial z values and step size Δx converge to the asymptotes, which are depicted by the solid black line. The unshaded region indicates combinations of initial z and Δx values that do not converge to a physical solution of the problem. "As can be seen, the basins of attraction narrow considerably for the larger values of Δx. (Note that the axis scale is 10^{-5}!)"

These calculations are extremely computationally intensive, and cannot be expected to be performed as a matter of course. In Yee et al (1991), the authors warned against obtaining spurious solutions, especially with implicit methods applied well beyond the stability limit of explicit methods. The points of their work are the danger of being misled by such calculations, the elucidation of the convergence behavior through description of the basins of attractions, and the need for further research to guarantee *a priori* accuracy. Their studies on basins of attraction shed some light on the "phenomenon of near (but lack of) convergence in large stiff systems," or what other authors have described as "stalling" of iteration convergence (e.g., see Roache, 1998b), by associating the behavior with the bifurcation of spirals to limit cycles. The authors claimed that "isolation of the sources of numerical uncertainties is of fundamental importance," but we do not agree that it is a necessary prelude to Verification of Calculations. Results from further such research are of course inherently interesting, and perhaps "can be a viable complement to the standard guidelines for numerical analysis and CFD practices," but we already have the tools necessary to Verify Calculations.

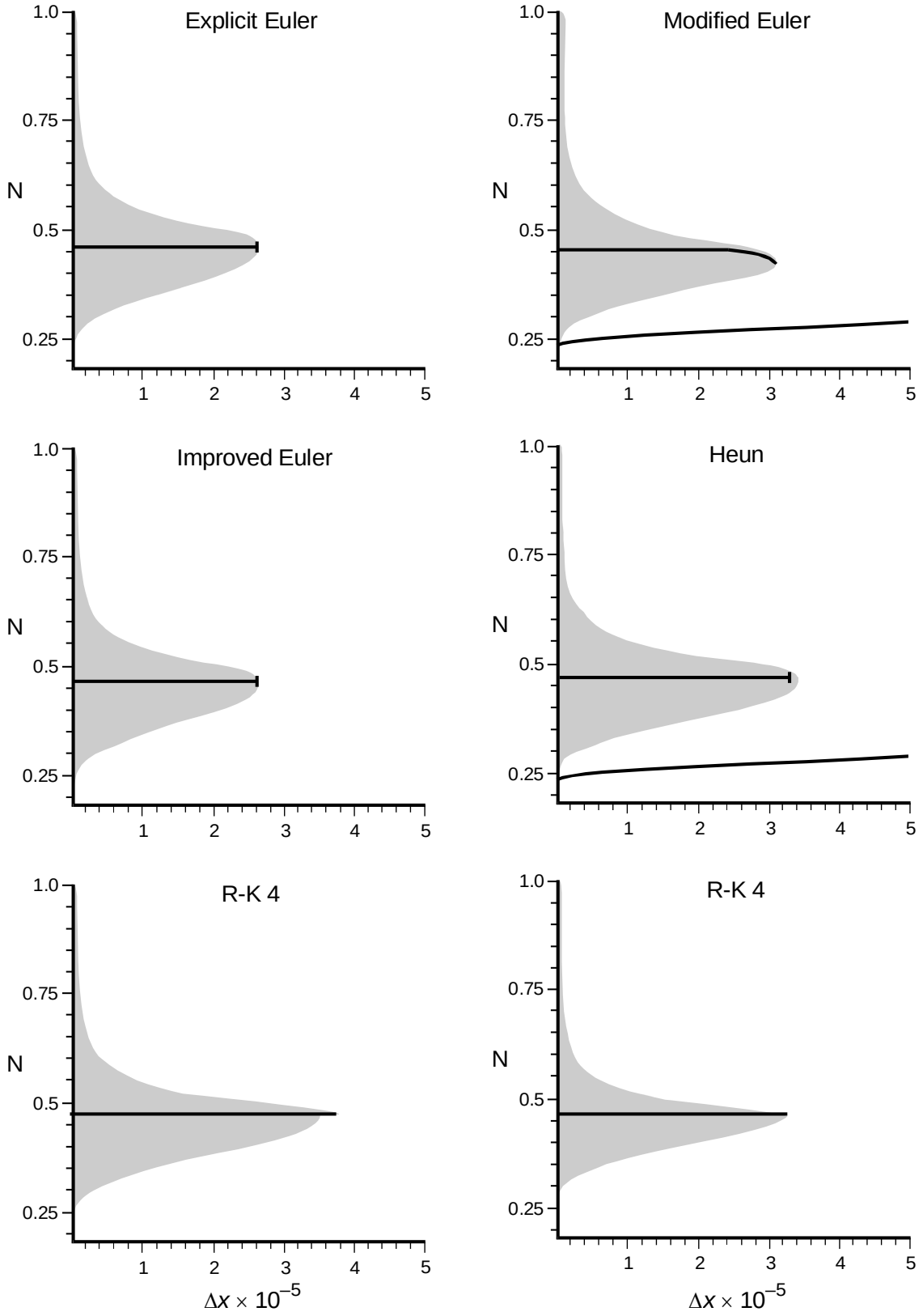


Figure 8.10.1. Basins of attraction for Equation 8.10.1 for various numerical methods.
(From Figure 4.1 of Yee and Sweby, 1996.)

The major goal motivating their work (Yee, 1992) was the development of numerical methods that guarantee *a priori* accuracy, without the need for grid convergence testing or other error estimators. We consider this goal to be unrealistic for any non-trivial problem (perhaps for anything but the Laplace equation), and that the real lesson of their examples is simply the need for *a posteriori* error estimation. We reject the claim (Yee and Sweby, 1996) that standard practices such as grid convergence tests may be "not possible (e.g., too CPU intensive)," as previously discussed (Chapter 4).

In all but one of the examples in Yee et al (1991), the most cursory time-step convergence study would have quickly revealed the inadequacy of the temporal resolution, with no reasonable chance of being misled into a non-physical solution. The one remaining example (involving a fourth-order Runge-Kutta integration) could have been misleading with a cursory study, but would have been revealed in a thorough systematic study. Many of the other extensive list of CFD topics discussed, especially in Yee and Sweby (1996), are overly broad and constitute non-issues to numerical uncertainties and Verification, e.g., solution strategies, or improving iteration convergence rates. (My remarks apply to computational simulations of physical processes, not to studies of numerical chaos for its own mathematical interest, which I do not share.)

In my opinion, the most important and surprising point to be gained from these examples is the following. With nonlinear iterative methods (not necessarily nonlinear *problems*, but simply nonlinear *methods*, such as classical 4th-order Runge-Kutta time integration, or nonlinear flux limiters such as FCT, TVD, ENO, etc.) it is necessary to check the effect of changing Δt (or equivalently, the relaxation parameter for non-time-accurate iterative methods) on the solution *even for a steady-state problem*.

As noted, the nonlinearity need not be part of the problem definition, but can arise simply from the numerical method itself. Nonlinear flux limiters introduce nonlinearity to linear problems whenever wave fronts in the solution are not well resolved. As noted by Yee et al (1991) such methods are especially likely to cause spurious steady-state solutions when they are implicit and when the time step used greatly exceeds the allowable stable time step for an explicit method. Note that some more straight-forward methods are also nonlinear, notably the common family of Lax-Wendroff methods. Applied to the constant coefficient scalar advection equation,

$$\frac{\partial f}{\partial t} = -V \frac{\partial f}{\partial x} \tag{8.10.2}$$

all the Lax-Wendroff methods reduce to (Roache, 1998b)

$$f_i^{n+1} = -c(f_{i+1}^n - f_{i-1}^n) + \frac{c^2}{2}(f_{i+1}^n + 2f_i^n + f_{i-1}^n) \tag{8.10.3}$$

where the Courant number is

$$c = V \frac{\Delta t}{\Delta x} \tag{8.10.4}$$

Thus, although the modeled equation is linear, the discrete equation is nonlinear not in the dependent variable f, but in the discretization parameters Δt and Δx (through the c^2). That the steady state discrete solution is actually a function of Δt for a Lax-Wendroff Euler code was demonstrated by Roache (1972a). (See also Appendix B of Roache, 1998b.) The interpretation given was in terms of the artificial viscosity for *steady state* solutions, which is different from the *transient* artificial viscosity for some methods (notably Lax-Wendroff). Once forewarned, the practitioner can easily test for this effect by changing the Δt (or analogous steady state iteration parameter) after the solution has been obtained. Such Verification is

economical because the bulk of the computational cost will have been expended in obtaining the base solution. If there is no sensitivity to Δt, this fact is verified with the work equivalent of an additional *single time step*. Also, unlike automated spatial grid convergence testing, this feature would be trivial to include in a code as an automatic feature, along with a warning message to the user if the Verification fails.

Since most of the examples of spurious solutions are trivial to check with time-step convergence testing, the sometimes high costs of spatial grid convergence testing are not an issue here. However, it is *very significant* that single-gird convergence estimators (Chapter 7) by themselves (i.e., without time-step convergence tests) presumably would *not* reveal these spurious solutions. Of course, the time-step convergence test is cheap, and involves no problem of spatial grid generation.

Another example of "chaotic-like" solutions is found in Yee and Sjogreen (2006, 2007). Multiple grid solutions (as fine as 6401x801) using highly accurate algorithms (up to sixth-order filter methods) converge very convincingly for the eddies of Richtmyer-Meshkov instability when viscosity below $\mathrm{Re} \sim 10^4$ is included. For higher Re and for inviscid equations, the solutions are chaotic and continue to change with grid resolution. But the inviscid equations are being used to solve an essentially viscous phenomenon (and for 2-D structures like eddies, there is no equivalent of the Rankine-Hugoniot relations for the high-Re limit for shocks). The equations are ill-posed, as revealed faithfully by the grid convergence testing. The intended point was not "the failure of grid refinement" but rather the inapplicability of grid refinement to chaotic solutions (Yee, 2007).

8.11 ADAPTIVE AND LOCAL TIME STEPPING, AND STEADY STATE

This phenomenon of spurious nonlinear dynamics solutions also suggests that the simplest adaptive time stepping approach will not be adequate for such nonlinear algorithms. That is, a simple adaptive time stepping algorithm of the *forward type* (not repeating the previous time step calculation, but only adjusting the next time step) could be tricked into accepting a spurious nonlinear solution. However, this can be adequately checked by storing the Δt sequence produced by the adaptive time stepping algorithm, and then repeating the calculation with the entire sequence reduced or increased by some factor, analogous to spatial grid refinement or coarsening.

In a related issue, Benek et al (1996) express concern about "unsteady acceleration schemes based upon local time stepping." These schemes basically consist of choosing a *local* Δt close to the optimum value indicated by analysis applied locally, and usually amount to setting the local Courant number ~ 1. Obviously, physical transient behavior will not be faithfully represented, since the method is inconsistent with the original PDEs. Benek et al (1996) stated the following. "It is assumed that such a procedure will produce the same converged solution as an unaccelerated method for steady state flows. Is there an effect on the approximation of unsteady flow with this type of acceleration?"

The answer follows. In the limit of all $\Delta t \rightarrow 0$ but at different rates, i.e. maintaining some Δt's larger than others, the PDE consistently simulated is one with a distorted time variable, so unsteady solutions will not be correct. Clearly, a continuum steady state solution for distorted time PDE will also be a continuum steady state for the undistorted time PDE, so the method is not dangerous. However, it is also conceivable that severely distorted local time stepping could confuse the non-uniqueness issue (see Chapter 2), possibly favoring a different steady state solution than a time-consistent method. Since uniqueness must be dealt with *ad hoc* in any case, we do not consider that local time stepping schemes are a significant problem.

8.12 OTHER QUESTIONS RELATED TO THE STEADY STATE

There are other questions regarding the attainment of a "steady state," raised by Benek et al (1996) and others.

- "How long should the simulation be run before the starting transients no longer affect the solution?"
- "What are the optimal initial conditions from which to start the simulation [to quickly attain the steady state]?"
- "If the steady state approximation is used [in the choice of the governing equations] will the same time averaged quantities be predicted [as would be obtained by time-averaging the solution of the time-dependent equations]?"

The response to all these questions is that they do not involve computational issues *per se*. That is, the answers are not computational but physical.

For example, it is well known that pitot pressure probes, which give a single physically-averaged pressure reading, do not produce the average pressure in a fluctuating field, but a reading higher than the average; e.g., see Dolling (1996.) Computational simulations can provide answers, but not *a priori*, anymore than experiments can. Like the strictly computational question of what will constitute adequate grid resolution, it is simply asking too much of Computational Science to answer such questions *a priori*, but the questions are answerable *a posteriori* from the simulations. Certainly, it is not true *generally* that steady state equations will produce the same time-averaged quantities as would be obtained by time-averaging the solution of the time-dependent equations. If they did, turbulence modelers would be out of business, since time-averaged turbulence would simply reduce to the laminar flow equations.

8.13 § LAGRANGIAN CALCULATIONS

Lagrangian calculations are not nearly as common as Eulerian, so it is expected that there would be fewer Verification exercises (code or calculation). But besides being less common, multi-dimensional Lagrangian calculations and error estimations are inherently more difficult conceptually. Time error estimation (isolated) is similar to the Eulerian approach, but the spatial aspects are complicated by the moving spatial grid. The "moving" of the grid is not merely an episodic re-gridding as occurs in solution-adaptive Eulerian calculations (although that is also necessary in Lagrangian calculations when a mesh begins to tangle, and involves re-mapping and re-zoning). Rather, the mesh movement is tied to flow "particle" tracking (mass is identically conserved) and therefore requires temporal accuracy. Essentially, a Lagrangian calculation is the original (and perhaps ultimate) solution adaptive grid. A Los Alamos report by Hemez and Brock (2008) described Verification of a Lagrangian code. Most exercises were performed by refining or coarsening initial grids by $r = 2$, and examining convergence of field data, time histories and integrated simulation features over as many as eight levels of refinement.

Results show convergence and dependable error estimation, but with many subtleties. (Hemez and Brock, 2008; Hemez, 2008) Pure Lagrangian calculations can converge at observed $p = 1$ on the test problems, as expected. For ALE (Arbitrary Lagrangian-Eulerian) solutions, mesh convergence overall is slow, with observed p as low as 0.23. This performance is greatly improved by a "modal decomposition" in which the first three modes capture 98% of the overall information, and allow poor performance (perhaps even instability) of higher modes to be ignored. The first and second modes exhibit improved convergence but still less than first order, with $p_1 = 0.73$ and $p_2 = 0.62$. The most troublesome aspect is the negative effect of ALE re-mapping and re-zoning, which theoretically should not affect the convergence rate but in fact dominate the error if enacted every time step, to the point that "asymptotic order of convergence" becomes meaningless. In some tests with non-uniform initial meshes, results with ALE re-mapping every

100 time steps were four times more accurate than results with re-mapping every time step. The authors noted that more analysis is required.

For a bibliography of Verification in the ASC Project through 2005, including exemplary high quality studies, see Brock (2005).

8.14 § LEAST SQUARES GCI FOR NOISY CONVERGENCE (RANS)

The Least Squares GCI procedure has been described in Section 5.11. As previously discussed, noisy convergence has various causes, including just mixed order differencing (Section 5.9.4). But the notorious offender is RANS turbulence models which have switches for different flow regions, whose boundaries in physical space change as grid resolution changes. These are essential non-smooth behaviors. A test for convergence behavior over one grid triplet cannot be considered conclusive, especially for a single variable (point value or solution functional). Strictly speaking, even results from several grid triplets could be misleading. However, as the number of grid triplets and/or the number of variables examined increases, confidence increases, and the Least Squares GCI approach improves reliability. Experience shows that difficulty occurs mostly when the modeler is trying to get away with a minimum amount of work and is straining for claims of minimum uncertainty; when sufficient computational resources are dedicated and realistic expectations of numerical uncertainty are held, convincing grid convergence can be obtained, even for RANS models. See especially Eça and Hoekstra (2008b,c).

8.14.1 § Overcoming False Indication of Convergence due to Sampling

Although convergence is essentially impossible to strictly guarantee, confidence is increased with more grids and more observed variables. Once again, the additional sampling could be done on nearby problems, rather than for every case in a large parametric study. The additional grids do not necessarily need to be finer, but just fill in gaps in the grid sequence. The variety of grid refinement factors r from these grid fill-in solutions would help, limited by noise from iteration convergence and those terrible RANS switches.

8.14.2 § Don't Shoot the Messenger!

Systematic grid convergence testing for benign problems, e.g. simple heat conduction,, reveals that the performance of the solution algorithm is well behaved. Observed orders of convergence p can be obtained that are remarkably close to theoretical values, even for unstructured grid generation and unstructured refinement (e.g. see Section 6.4.2). For other problems, notably RANS turbulence models with step function switches and high local gradients and sloppy iteration convergence, observed convergence is often noisy and non-monotone. The point here is that such noise is a property of the physical problem, the computational model and the solution algorithms, not the grid convergence test *per se*. It is not a shortcoming of the V&V procedure. Rather it is a reflection of the difficulty of the physical problem, and/or the shortcoming of the computational model and the solution algorithms, which are *revealed* by the V&V procedure, not *caused* by it. Please, do not shoot the messenger.

On the other hand, note that noise in p does not necessarily imply that the numerics are inaccurate or non-convergent (Roache, 2003a). In fact, the *size* of the error might be acceptably small for Calculation Verification. However, the observed *order* can be degraded and noisy, meaning that error estimates, and therefore the use of the corrected solution and/or the Uncertainty calculation, are more problematical. Rather than attempt to extricate Uncertainty estimates from noisy data, the Least Squares GCI offers a more robust alternative: as Raven et al. (2002) stated, "The use of common sense may be unavoidable."

PART III

VALIDATION

In formal logic, a contradiction is the signal of defeat.
In the evolution of real knowledge, a contradiction
marks the first step in progress towards a victory.

Alfred North Whitehead.

Part II of this book covered "Verification", the process of determining whether or not a code is "solving the equations right", a purely mathematical question. Part III now covers "Validation", the process of determining whether or not a code is "solving the right equations" for a particular physical problem. Verification is essentially and strictly an activity in *mathematics*, the mathematics of numerical analysis, whereas Validation is essentially and strictly an activity in *science*: physics, chemistry, fluid dynamics, even the "soft" sciences of economics, sociology, etc. Common sense and the testimony of experienced practitioners strongly indicates that rigorous Verification, first of the Code, then of the particular Calculation, should *precede* any Validation exercise.

In the first edition of this book (V&V1), I had somewhat less to say on Validation than Verification, because there was less methodology involved. Fortunately, this situation has vastly improved. Chapters 9 and 10 have new material added to the first edition, but follow the same Validation methodology, which can be described as an "error bar" approach. This is in contrast to the Validation Uncertainty approach, pioneered by H. Coleman and developed fully in V&V20. The new Chapter 11 of this book gives a brief description of this Validation Uncertainty approach, which we consider to be a major development in V&V.

It is worthwhile to repeat here the limitations of our attempts at semantic distinction, already discussed in the conclusion to Chapter 2. Although valuable, it is clear that we (the scientific - mathematical - engineering community) are *not* going to achieve uniform, non-overlapping terminology, even in the most nearly universally accepted terms, Verification and Validation. (See especially Appendix B.) Nevertheless, following the advice of Chuang Tzu (see Introduction to Chapter 2), it is worthwhile to keep in mind the ideas *behind* Verification and Validation. In reading journal papers

and reports, it is not a good idea to try to enforce terminology or taxonomies, but rather to try to detect the authors' terminology or taxonomies (often implicit, rather than explicitly defined) and to learn from the authors' experiences and perspectives.

CHAPTER 9

DIFFICULTIES WITH EXPERIMENTS AND VALIDATION

This Chapter 9 covers general concepts of experimentation and Validation exercises. The following Chapter 10 will present specific examples. The theme chosen for this Chapter is the *difficulties* with experiments, for good reason. Experimental data is not as sacrosanct as many computational practitioners believe.

9.1 CREDULOUSNESS

There is a saying in the aerodynamics community. "No one believes the CFD results except the one who performed the calculation, and everyone believes the experimental results except the one who performed the experiment." (On a personal note, my very early career experience was as an experimentalist, so I have never had a problem with excessive gullibility.)

Very significant progress has been made in "non-intrusive" experimental techniques (e.g., Marvin, 1995) including 3-D laser velocimetry of mean and fluctuating velocities, laser fluorescence for mean and fluctuating densities and temperatures, laser interferometry for local skin friction, and pressure-sensitive and thermographic paints for surface pressure and heating. These are vast improvements over previous clearly intrusive techniques such as pitot-static tubes, hot wire anemometers, thermocouples, etc. (Even flow visualization can be intrusive for liquid free convection flows, as the illumination provides a heat source; Jankowski, 1985.) At the other extreme, removal of rock from its field location for laboratory testing can clearly have order of magnitude effects on permeability measurements. Alternately, the preferential selection of field samples which are intact can bias the sampling towards lower permeabilities (Brinster, 1995). But even for virtually non-intrusive measurement techniques, there are still experimental

errors involved. For example, wind tunnels suffer from flow angularity and blockage effects, which are further complicated because they vary with angle of attack of the models. Likewise, hypersonic facilities often are plagued with axial static pressure variation that has no counterpart in flight.

Briefly, it is naive to accept a single experiment as the final word. As Marvin (1995) noted, "Reliance on single experiments or measurement procedures during the code Validation process should be viewed with caution because of facility and instrumentation limitations. Therefore, redundant measurements and similar experiments performed in more than one facility are desirable." Also, Aeschliman and Oberkampf (1997, p. 13) make an interesting observation based on their extensive experience. "The general point is that as one progresses down the list to more difficult quantities for CFD to predict, the experimental uncertainty generally increases also."

Uncritical acceptance of experimental values can and has held up progress for years. Both false invalidation (i.e., a negative conclusion on Validation accuracy) and false Validation have occurred. Wilcox (1994) described one case where disagreement with ablating heat transfer data was originally attributed to inadequacy of a turbulence model. After several years, direct communication with the experimentalists clarified previously incomplete boundary conditions and produced excellent agreement. Conversely, a turbulent pipe flow computation (with a different turbulence model) that agreed well with experiment proved to be false Validation; later reviews showed *both* experiment and computation were seriously in error (Aeschliman and Oberkampf, 1997).

Barber (1996) provided a review of several code Validation/Certification studies in the aerodynamics literature, and cited examples wherein missing experimental details have lead to poor Benchmark calculation comparisons. The details he discussed are

1. the geometrical definition,
2. the data reduction procedures, and
3. the dominant physics.

Similarly, Marvin (1995) noted that "Accuracy assessments for experimental data are essential; otherwise there is no quantitative means for determining the Validated range of a code." Required are "error estimates for test geometry dimensions, test operating and freestream conditions, model and flowfield measured variables, and instrumentation." In the very difficult experiments associated with hypersonic airbreathing engines, "geometrical definition" errors include the axial and vertical cowl *positions*. Other sources of error are model wall temperature and tunnel operating conditions (Van Wie and Rice, 1996). In these experiments on hypersonic inlet performance, the data reduction errors reach a new level. "Given that CFD techniques play a critical role in the measurement process (through estimating certain quantities and sensitivities), it becomes difficult to separate the Validation of the measurement procedure from the Validation of the [computational] analysis procedure. In this situation, the experimental and [computational] analytical techniques are intertwined in a single process." (Van Wie and Rice, 1996).

Also, Marvin (1995) stated that "Uncertainty analysis is a well established method for determining experimental data accuracy, and it should be a prerequisite for all Validation databases," for which he cited the well-known paper by Moffat (1981). For more recent systematic discussion of experimental uncertainty, see Coleman and Steele (1995), Coleman et al (1995), Coleman (1996), Steele et al (1996), and Aeschliman and Oberkampf (1997).

In the aerodynamics community, there is a consensus among those who have seriously considered Validation issues that results from older experiments not designed specifically for Validation are usually inadequate for the task. For example, Aeschliman and Oberkampf (1997, p. 10) stated the following. "No rational computational fluid dynamicist would suggest there is no need for Verification or Validation of CFD codes. A common view among CFDers, however, is that while code V&V is indeed necessary, the process - particularly the Validation step - can be accomplished through comparison to existing data,

documented in reports or archival journals, obtained for some purpose other than CFD code Validation. We most strongly disagree."

Their reasons, based on wide experience, were as follows. "Almost invariably, critical details are missing from published data, particularly for archival journal publications where discussion is limited in the interest of reducing paper length. It is critically important that the boundary and/or initial conditions assumed by the code be accurately known from the experiment." Some specific examples will be given in Chapter 10. We know that this situation in the aerodynamics community is representative of many other areas of computational science and engineering, perhaps most. My own database for this conclusion includes groundwater flow and transport, free surface flows, heat transfer, combustion, computational solid mechanics (CSM) or structures. The simplest conceivable CSM experiments, the bending of loaded cantilever beams, typically lack data on compliance at the supporting wall.

Porter (1996), reporting on several workshops in the aerodynamics community on Validation/Certification, included these consensus statements (as of 1996).

- There is no focused effort to provide the understanding of fundamental physical phenomena so they can be modeled in CFD codes.
- There is a dearth of archival quality, benchmark experimental data specifically designed for CFD code Validation and Calibration.
- Experimental data that do exist have deficiencies, inadequate documentation, and are not readily accessible to the community at large.

9.2 Δ VALIDATION IN SCIENCE THEORY AND COMPUTATION

9.2.1 Historical Methods of Validating Scientific Theories

The task of Validating a conceptual model and code has been likened to the general problem of Validating scientific theories[96] (Mehta, 1996). "In the philosophy of science, the historical methods of validating theories are rationalism, empiricism, and positive economics. Rationalism involves the logical development of the model based on indisputable axioms...Empiricism requires that every axiom, deduction, assumption, or outcome be empirically confirmed...Positive economics is concerned with the model's predictive capability and not its structure, assumptions, or derivation. An example is the calibration of a model to predict certain features and quantities." (Mehta, 1996; Naylor and Finger, 1967).

Pure rationalism is perhaps appropriate for ethereal regimes of pure science, but most scientists and all engineers would not trust major projects to involved theories developed by difficult-to-follow derivations of "indisputable axioms." On the other end of the spectrum, positive economics could be regarded as nothing more than thinly disguised blind extrapolation, e.g. stock market models. As a practical matter for code Validation in science and engineering projects, empiricism seems to be the operative philosophy. It is not so daunting a task when approached modularly, i.e. by building on past work.

Ultimately, code/model Validation will come down to comparison (directly or indirectly) of code predictions with physical experiments. (We do not agree with Mehta, 1996, Figure 6 and elsewhere, that Sensitivity-Uncertainty Analysis can be included with rationalism, empiricism, and positive economics as a method of Validation.) The comparison can be direct or indirect. Indirect comparison occurs when a previously Validated code/model is taken as a Benchmark. In this philosophy, the Benchmark code may be regarded as a repository of experimental information as well as a means of interpolating and smoothing

[96] An ostensibly reasonable position that will be challenged in the following paragraphs.

previous experimental results. The Validation exercise still compares numerical answers to experimental data, but it is second-hand experimental data, one level removed from the original experiment. As a practical matter, the error bars of the original experiments will be widened by additional computational and conceptual uncertainties of the Benchmark code. Thus, indirect Validation may be convenient and even appropriate for a project, but it does not solve any of the difficulties with experiments and Validation; rather, it compounds them.

9.2.2 § Objections to Validation based on the Philosophy of Karl Popper

The two most influential philosophers of science in the 20th century were Karl Popper and Thomas Kuhn (e.g., see Kasser, 2006). They used very different approaches. Kuhn (1962) was innovative for studying how scientists actually worked, his own studies being as much history of science as philosophy of science. Popper (1980) was very traditional, engaging in the kind of hair-splitting subtleties that can give academics a bad reputation. Popper was highly influential in philosophical circles for some time, although his work is of less current interest to philosophers of science (Kasser, 2006) [97]. However, he has had a lasting influence on scientists and engineers, and on the question of Validation. He has been quoted as an authoritative witness to the fundamental impossibility of Validation of computer models[98] by a Blue Ribbon Panel on Simulation-Based Engineering Science (NSF, 2006) and in widely cited papers by Oreskes et al (1994) and Konikow and Bredehoeft (1992), the latter entitled "Groundwater Models Cannot be Validated." [99]

This importance in the computational modeling community is remarkable, considering that the applicable philosophical arguments appear in the first edition of Popper's most cited book, *The Logic of Scientific Discovery* (Popper, 1980), the first edition of which was published in 1934, well before the advent of modern computers and modern computational modeling. Whatever Popper's contributions or claims were, they were not directed specifically towards Validation of computer models, but to scientific theories in general.

The most often cited contribution of Popper is his supposed invention of the concept that a scientific theory must be falsifiable in principle, otherwise it is not science. He used this as the basis for solution of Kant's "Problem of Demarcation," i.e. separation of science from metaphysics, or in terms of modern concern, separation of science from pseudo-science. Although this criterion is very popular among scientists, it does not hold up very well to close scrutiny, as described insightfully by Kasser (2006). According to later interpreters, Popper is said to have replaced the criterion of *verifiability* used by logical positivists with his concept of *falsifiability*. Popper himself seemed to claim this as a contribution (pg. 280) but in fact it was already used by his chosen adversaries, the logical positivists, as Popper himself had acknowledged earlier (page 17). In fact, the logical positivists allowed the possibility of proof (verifiability) as well as disproof (falsifiability) and required *both*, not merely *either*, to deserve their term of *completely decidable*. Popper's contribution was thus not to invent a criterion, but to reject one. While insisting (rightly, we agree) that any statement aspiring to be categorized as science be *testable* and *falsifiable* in principle, Popper also insisted that no such candidate scientific statement can ever claim to have been validated (verified).

---

[97] Also corroborated by lack of mention in recent publications of the Philosophy of Science Association.

[98] Another claim of impossibility of Validation has been given by Hazelrigg (2003) based on Arrow's Impossibility Theorem, but this has no perceptible relevance to the concept of Validation of computational models.

[99] See also discussion in Section 2.2.

Actually, Popper did not use the term *validate* but rather used *verify* in the sense that we now used *validate*, which causes some confusion in attempting to quote him! In relation to his approach to scientific theory, *verify* may be preferable on etymological grounds, rooted in the Latin for *truth*. The *truth* of a scientific theory was Popper's concern. But in the modern use of *Validation* in conjunction with computational PDE models, we are not concerned with some convoluted, vaguely defined concept of *truth* but rather with the simple, well-defined concept of *accuracy*. Many (not all) scientists would agree with Oreskes (1999, p. 314, 368) that Newton's laws of motion and Einstein's theory of relativity cannot both be true.[100] Yet both can be demonstrated to be *accurate* within well-defined parameter ranges, i.e. a "domain of validation" in our computational PDE terms. In fact, the Newtonian laws are as accurate as Einstein's for any human-scale problem and tolerance, and are much preferable because of their relative simplicity. Popper would have argued that neither could be claimed to be *true* or *verified* [*validated*] but he allowed the word *corroborated*.[101] Even if one were inclined to grant Popper's position for scientific theories, it is not applicable to Validation of computational PDE models. As Rykiel (1996) stated, "These terms are defined in a limited technical sense applicable to the evaluation of simulation models, and not as general philosophical concepts." He also noted that "Single minded focus on falsification is a superficial treatment of a complex subject..." and noted, with examples, that "The impulse to falsify can result in 'naive falsification'."

However one may judge the applicability of Popper's philosophy to computational models, it is certainly not less applicable to scientific theory in general. His position is as clear as it is useless[102] to practical engineering and science; "every scientific statement must remain *tentative forever*." (Pg. 280.) To avoid disputation and agonizing over what he or we may mean by *truth*, we can grant his statement, in some rarefied and harmless sense, while noting that Validation of computational PDE models is thereby positioned in the same category as Newton's laws of motion and gravity, Einstein's theories, entropy, Darwinian evolution, conservation of mass, Fourier heat conduction, etc. We are in good, respectable company.

---

[100] Precisely, she wrote (p. 314) that "Classical mechanics can be partly rescued from the scrap heap of discarded knowledge by arguing that it remains approximately true at speeds far less than [that of] light..." and (p. 338) "Either space and time are absolute or they are not; these are not epistemologically reconcilable positions." (We would use "accurate" rather than "approximately true," a philosophically troublesome term which would seem to dilute any claim to epistemology.) Whatever the resolution of this philosophical point, this fascinating book should be required reading for any student of science.

[101] Oreskes et al (1994) prefer that Popper's term *corroboration* should be applied to what everyone else calls Validation of computational models, on the basis that it is less misleading to the general public. But as noted in Appendix C of V&V1 and by Rykiel (1996) all three terms - verification, validation, corroboration - are synonyms in general non-technical use, so one is as misleading as another. See also the next Section 9.2.3.

[102] If my evaluation of Popper sounds too harsh, consider his claim, p. 444; "... the statement 'this container contains water' is a testable but non-verifiable hypothesis, transcending experience." And, I would add, transcending common sense. I can tolerate Popper's position, given his perspective and intellectual tradition, but I cannot understand how practical engineers could take this stuff seriously. They also ignore Popper's distinction between *strict universality* and *numerical universality*; the latter *is* falsifiable according to Popper himself, and it is the appropriate association for modern Validation. (It is appropriate not merely because of Popper's use of "numerical," which today could be better termed "enumerable." See discussion by Popper (1980), pg. 40.)

9.2.3 § Validation in Ecological Modeling

The paper by Rykiel (1996) is entitled "Testing ecological models: the meaning of validation." Although ecological models require special considerations, Rykiel's penetrating discussion of Validation and related issues are relevant to all areas of computational PDEs. The paper is highly recommended reading, but the reader should recognize that Rykiel's concept of Validation includes adequacy criteria or what we have referred to as "pass-fail" criteria.[103] As noted earlier, this is a justifiable and not uncommon conception (as in V&V1), but our recommended approach is to consider Validation to be primarily an exercise in accuracy assessment rather than adequacy assessment, which we reserve for some term like Certification or Qualification. The reader may find some minor inconsistencies within the paper resulting from this conceptualization. Also, the author's "Verification" refers only to Code Verification.

Rykiel gave a chronological review of Validation concepts in the ecological literature which shows a considerable range of opinion. In regard to Popper's preoccupation with *truth* (see previous Section 9.2.2), Rykiel stated "Validation is not a procedure for testing scientific theory or for certifying the 'truth' of current scientific understanding ..." He also quoted Levins (1966) that "A mathematical model is neither an hypothesis nor a theory."

Ecology is certainly a science, and a socially important one, but the data sets are and will remain fuzzy compared to more controlled disciplines.[104] It is doubtful that many of the precepts espoused in this book are applicable to ecological modeling. More flexible standards seem appropriate. Whereas this book considers Validation to be applicable only to experimental data sets with some estimate of experimental error and uncertainty, Rykiel considered Validation tests to include both qualitative and quantitative measures of system performance. He listed 13 types of Validation procedures, including low-end qualitative ones like "face validity" [expert opinion] and visualization techniques, then through comparison to other models, internal validity [self-consistency], to quantitative data comparisons like sensitivity analysis [i.e. using sensitivity as the observable f] and predictive validation [in the strict temporal sense]. The softer of the procedures would be rejected for claims of Validation for disciplines like engineering[105], being relegated to the category of preliminary evaluations, but this is probably not appropriate for ecological modeling.

Rykiel also argued that Validation is neither required for all ecological projects nor necessary for scientific value. [One could imagine a model built just to test sensitivity and "what if" scenarios, e.g. "what if" the coefficients of a predator-prey relation, which are not directly measurable, are radically changed. This could lead to insight about the underlying conceptual model, certainly part of scientific progress, and might suggest new and perhaps testable hypotheses without any claims of predictive ability for the model.] "Exploration of model behavior without Validation testing is a legitimate, reportable activity." The standards espoused in the rest of this book are probably not applicable to ecological modeling,

9.2.4 § Validating Temporal "Predictions" vs Outcomes

When speaking of the "prediction" of a computational model for Validation with an experiment, the word should not be taken literally to indicate a time sequence. Usually, the terms "outcome" or "results" would better convey the process. Validation involves comparison of the outcome of a model with the outcome of an experiment. (The outcomes or results include not only the best value of variables but the associated estimates of errors and/or uncertainties.)

---

[103] See Section 2.3.1 or Appendix B.

[104] "The relative inaccuracy and imprecision of ecological data also places limits on model testability." (Rykiel, 1996)

[105] See V&V10, V&V20, ASCE/EWRI (2009), and the rest of this book.

This is true for any scientific theory, not just computational models. Philosophers of science agree, virtually without dissent, that the time sequence does not determine the worth of the theory. In some turbulence modeling workshops[106] much has been made of the distinction between temporal "prediction" vs "postdiction", i.e. running the code to obtain model outcome after scrutiny of the experimental data. Certainly such a postdiction is not as impressive as a temporal prediction or "blind" comparison as urged by Oberkampf and Trucano (2007, 2008) especially in a modeling area that allows for adjustment of questionable free parameters (which was more common practice in the 1980s than in the 21st century) and guidance by experimental data in local grid generation. As noted earlier, the most common form of "tweaking" a model arises not because of inadequacy of the general computational model but because of incompletely measured experiments that do not define the problem.[107] Yet V&V10 prefers strongly (in earlier drafts, insisted) that simulations must predate experiments to qualify for Validation. This reflects a trial jury mentality rather than a scientific one. In 1915, in one of the greatest achievements in science, Einstein "predicted" the perihelion advance of Mercury with his General Theory of Relativity. But the experimental data and conclusion had been published by Newcomb eighteen years earlier. Was Einstein a fraud? Not compared to Newton, who "predicted" the motion of planets a century after Kepler's data were known. Of course, Validation of theories or computational models is an ongoing process, but this is true whether or not the initial Validation resulted from a prediction or a postdiction. There is no logical case for insisting on temporal predictions for Validation, rather than agreement of outcomes.[108]

9.3 THEORY-LADEN EXPERIMENT

As we noted in Chapter 2, "Validation" in a real sense applies to the Conceptual Model embodied in a code, rather than the code itself. (The code correctness is the subject of Verification, etc.) In regard to the present subject of experiments and Validation, we note the following.

A Conceptual Model is needed prior to experimental data gathering.

For a broad definition of "Conceptual Model," I adopt that of Johnson (1996, p. 172) described in a rather different context (historical-critical models or methods) but that still fits our needs. "I use the term *model* to mean an imaginative construal of the subject being studied, as well as a structured picture of both process and product: a model is a paradigm within which the data pertinent to a discipline makes sense."

[106] E.g. Kline et al (1985a,b), Levy et al (2002)

[107] As Oberkampf and Trucano (2007, 2008) stated, "Without a doubt, the most common [calibration] parameters that are optimized are those that were not provided by the experimentalist in the documentation of the experiment."

[108] If the fraud, deception and/or "tweaking" paradigms seem more worldly-wise and mine more naive, consider how naive it would be to *trust* claims of a temporal sequence in strict "prediction." Publication dates certainly do not accurately reflect access to experimental results (especially preliminary results). Are experimentalists now to be further burdened with security measures? (See Section 10.19.) Has it not been long recognized that such secrecy is detrimental to democratic values and creativity and progress in science? Furthermore, the process of comparison and model updating (the polite term for tweaking) ultimately is iterative and evaluations are conditional, no matter who does what first. However, insistence on temporal prediction is more justifiable if the goal of a Workshop is not so much Validation of individual computational models but rather an assessment of "state of the art" as in Hemsch (2002a) and Levy (2002) and the ongoing AIAA Drag Prediction Workshops (http://aaac.larc.nasa.gov/tsab/cfdlarc/aiaa-dpw/) where poor agreement would selectively discourage participation and thereby bias the assessment.

For example, an experimental measurement of permeability for flow in porous media *requires* the conceptual model of Darcy Law, which *defines* "permeability". Likewise, hot-wire anemometry requires the Newtonian law of heat transfer. Pressure measurement requires non-directionality of pressure (diagonalization of *P* tensor, which is not so obvious - see any fluid dynamics text). There are also assumed scales of unsteadiness and spatial variation involved in any experimental data taking. As a practical matter (vs. rarefied philosophizing) the "model" is so minimal for Aristotelian pure observation that it has no consequences, but for practical data gathering in science, engineering, and other disciplines, it is important to keep this fact in mind (Johnson, 1996).

"Every observation is laden with theory."

Likewise, most experiments involve computation, and many now involve *simulations*. Examples will be given in Chapter 10.

9.4 RANDOM AND SYSTEMATIC ERRORS IN EXPERIMENTS

Certainly every experiment has some uncertainty associated with the results, and certainly it makes no sense to expect or even look for computational comparisons finer than this experimental uncertainty. As Coleman (1996) stated, "the [experimental] uncertainties should set the scale at which comparisons (or Validations) should be made or attempted. This holds for data-data, model-data, and model-model comparisons."

"Total error can be considered to be composed of two components: a *precision* (random) component and a *bias* (systematic) component..."..."An error is classified as precision if it contributes to the scatter of the data [in repeat experiments in the same facility]; otherwise, it is a bias error. It is assumed that corrections have been made for all systematic errors whose values are known." Unfortunately, all experimentalists know that random component errors are straightforward to evaluate (though tedious and often expensive), whereas systematic (bias) errors are difficult. It is only bias errors that can be removed by Calibration; e.g., see Van Wie and Rice (1996).

Coleman (1996) gave a partial classification of possible sources of bias errors in experiments:

- calibration errors,
- data acquisition errors,
- data reduction errors,
- test technique errors, etc.

He also considered the propagation of random and bias errors through the data reduction process, and the important but difficult consideration of *correlated* precision uncertainties.

Aeschliman et al (1995) provided an excellent overview of ground-based experimental aerodynamics uncertainty analysis, including a very specific and complete case study of the Sandia National Laboratory's long-term project called JCEAP, for Joint Computational/Experimental Aerodynamics Program, on hypersonic wind tunnels. Aeschliman and Oberkampf (1997) also highly recommend the AGARD (1994) compendium as "by far the most detailed prescription for dealing with systematic and random errors in wind tunnel data when the systematic errors have been previously identified and estimated," and stated that this reference "identifies virtually every conceivable source of experimental error in wind tunnel testing and greatly improves the art of wind tunnel test data uncertainty analysis."

9.5 EXPERIMENTAL ERRORS IN PHYSICAL PROPERTIES

Physical property values (e.g., viscosities, specific heats) can be a source of error in both computations and experiments. If a code is based on constant properties, and this causes a deviation of the predictions from the experimental values, the blame is clearly on the (conceptual) modeling. However, it may also be true that a better average value would move the predictions into acceptable agreement with experiment. Also, if the problem is not variability *per se*, but just uncertainty in the correct value, it is hard to blame the computation, especially since the uncertainty is an *experimental* uncertainty (although not the *same* experiment), and since the same uncertainty often leads to experimental error in the data reduction. Coleman (1996) gave an example of a (significant) 2% uncertainty in computed specific impulse of a rocket engine due to the range of available reaction rate data. The distinction between "uncertainty" in properties and "variability" over a parameter range can become murky. Coleman (1996) used the word "uncertainty" in the following sense. "The uncertainty U associated with a measured quantity or a predicted quantity defines the $\pm U$ interval within which we expect the true (but unknown) value of that quantity to lie 95 times out of 100." That is, this "uncertainty' is *defined* to be approximately 20:1 odds or expectation. (For a Gaussian distribution, this corresponds to 2σ, but the definition and concept do not depend on any distribution.)

9.6 BOUNDARY CONDITIONS, CONTINUUM AND NUMERICAL

When modelers speak of errors in boundary conditions, they usually are thinking of errors in the numerical modeling, i.e. errors that are *ordered*, either in a discretization measure Δ or in distance to some far-field boundary L_b. These are the subject of Verification, i.e. a purely mathematical problem. As Oberkampf et al (1995) pointed out, there are also plenty of physical modeling problems associated with even the apparently simplest of boundary conditions in CFD, that of a wall. These include velocity slip and temperature jump in rarefied flow, boundary conditions for chemical species, vibrational nonequilibrium effects, porous walls (continuous porosity or discrete jets?), fidelity of surface shape, surface roughness for some turbulence studies, etc. These are problems in the description of the *continuum* boundary conditions, and are therefore the proper subject of Validation rather than Verification.

9.7 TRENDS, COMPUTATIONAL AND EXPERIMENTAL

In Chapter 8, we have already noted the false optimism often associated with predicting trends, rather than absolute values, from computations. Specifically, we may not be justified in assuming that correct trends (of solutions with parameter variations) will always be predicted with even *qualitative* accuracy in under-resolved grid simulations. (That is, counter-examples exist wherein under-resolved grid calculations have predicted the wrong *directions* for trends, let alone the wrong quantitative slopes.) If one attempts a Validation of a code for the prediction of trends rather than absolute values, one encounters the parallel situation for *experimental* errors, as discussed perceptively by Coleman (1996).

Consider an experiment (and corresponding computation) to predict the increment Δ in some flow functional C_D (which might be a drag coefficient, a discharge coefficient, or some other functional) with a change in geometry (or some other parameter). The ΔC_D from two experimental tests would be compared to the ΔC_D determined by two simulations in order to Validate the Code. In treating the uncertainty of the physical experiment, Coleman stated that "consideration of correlated bias uncertainties is crucial," and that "many misconceptions have appeared in the engineering literature regarding the effects of bias uncertainties" in such comparative tests, including two ANSI/ASME (1984, 1986) standards! These

publications claimed that the uncertainty in the (difference) increment measurement "will be composed only of precision (random) errors, because the bias (systematic) errors will cancel out." Coleman shows that these statements are incorrect in general, both for difference tests (Coleman, 1996) and for ratio tests (Coleman et al, 1995). Fortunately, the systematic uncertainty in the increment *can* be significantly less than the systematic uncertainty in the absolute value uncertainty from either test, especially if the same test apparatus and instrumentation are used in the two tests. The minimum increment that can be distinguished in a comparative experimental test must certainly be larger than its (total) uncertainty. This also applies to computational simulations, and as a practical matter demands *more* stringent iteration convergence criteria for comparative computations of increments Δ, more so when Δ is small; otherwise, the true increment Δ will be lost in the noise of incomplete iteration convergence.

In a related article, Steele et al (1996) discussed subtleties of determining confidence intervals for experiments in which some measured variables have asymmetric systematic uncertainties. Again, the ANSI/ASME (1984, 1986) standard methods are found to be lacking. The method of Steele et al would also be applicable to numerical uncertainties.

Van Wie and Rice (1996) gave a good example (their Figure 14) of simulations predicting trends well, and absolute values not so well, for mass flows in hypersonic airbreathing engines. They also stated that "Even if perfect agreement between experiment and calculation is achieved at one point, the code is nearly useless if the experimental and computational results trend in opposite directions." This is not an adequate condemnation, in my opinion. Rather than "nearly useless," I would say that such agreement is either totally useless or pernicious, since it has such potential to mislead if the trend discrepancy goes unnoticed.

Benek et al (1996) described the experimental methodology associated with measuring the increment of total aircraft drag caused by engine inlets. They noted that experimental accuracy of this [physical] increment is frequently limited by the insufficient number of pressure measurements (i.e., *experimental* resolution) and that the advantage of a simulation is the higher resolution. Then, "the CFD inlet drag [physical] increment can be used as either a correction [numerical increment] to the measured [physical] increment or as a replacement for it." This increment-of-an-increment process, i.e. a numerical increment of a physical increment, would seem to be a powerful capability for simulation, with uniquely tolerant requirements for experimental Validation.

9.8 FALSE NEGATIVES AND FALSE POSITIVES

Aharoni (1995) discussed the pitfalls of using physics experiments to prove or disprove physical theories, but the experience appears to be applicable to simulations as well. He claimed that "sometimes theory and experiment are both correct but do not agree with each other; sometimes a wrong theory agrees with experiment. One must therefore be careful not to jump to conclusions."

His examples are from magnetism, and the article is worth reading, although flawed, in my opinion. The *practice* recommended is good, but the semantics are inadequate and misleading. In a not uncommon pattern, the word "theory" is used ambiguously in the above summary. Essentially, the reader should separate his word "theory" into "theory A" and "theory B." The author then says that theory B is responsible for the disagreement with experiment and we (the theoreticians) erroneously attribute the disagreement to theory A. This is somewhat of a shell game with words. Another example involves a (possibly) correct theory that disagrees with experiments, but the details show that it is simply a matter of (perhaps insurmountable) experimental control. Thus it is not fair to say that the (possibly correct) theory does not agree with the experiment, but only that all experiments to date are flawed, and are incapable of validating or invalidating the theory. This is an "un-measurable theory" at present (an increasingly familiar theme in modern cosmology). The fault is with the experiment, and so should not be described as a "disagreement with theory" but just as an experimental failure.

However, Aharoni's examples are significant to practical science, especially in view of the comments in the previous Section 9.3, e.g., "*Every observation is laden with theory.*" As Aharoni stated it, "a theoretical value is rarely compared with a *directly* measured value. More often than not, there are hidden assumptions in the analysis of the experimental data." In Aharoni's example, the assumption was one of one-dimensionality near a wall, which turned out to be incorrect.

The principal message is again the difficulty of comparisons with experiments, which should not lightly be taken as correct.

9.9 "NEARBY" PROBLEMS

As noted in Chapter 2, in a meaningful though overly scrupulous sense, a "Code" cannot be Validated, but only a Calculation (or range or calculations with a code/model) can be Validated. However, it is clear that physical problems and their solutions present more-or-less continuum responses in their parameter spaces. Although parameter "transition" boundaries do occur, at which solution properties can change discontinuously or rapidly, these parameter transition boundaries are at least countable, and are usually few. The determination of the parameter transition boundaries is the task of the entire professional community (experimental, theoretical, computational) working in the subject area. A few examples of such parameter transition boundaries include the following.

Examples Of Parameter Transition Boundaries

- In aerodynamics:
  - the appearance of local supersonic flows
  - boundary layer transition from laminar to turbulent flow
  - appearance of separated flows
  - vortex shedding in the dynamic stall cycle, etc.

- In ocean calculations:
  - the encounter of the continental shelf break
  - thermocline modeling
  - occurrence of salt fingers, etc.

- In groundwater flow and transport calculations:
  - loss of Darcy flow conditions due to significant fracturing or high Reynolds number effects ("high" in this context meaning $Re \sim 1$)
  - appearance of vadose zone or two-phase flow effects, etc.

- In any modeling:
  - the appearance of geometric singularities like sharp corners.

Away from such parameter transition boundaries, and for most of the allowable parameter range variation, solution response is more or less continuous, and Validation for one particular parameter combination can be used to impute confidence to the computation of a nearby problem. Thus, most Validation exercises are not viewed as being restricted scrupulously to a single parameter combination but as having some value for nearby problems.

Nevertheless, the nearby condition is often harder to achieve than one might think. For example, a NACA 0012 airfoil is very close geometrically to a an NACA 0015 airfoil, yet their dynamic stall behavior can be significantly different. (For other testimony in aerodynamics, see Cosner, 1994.) The methodical determination of the class of nearby problems leads to flow taxonomies for Validation particular to each branch of science and engineering, examples of which will be given in Chapter 10. In this sense, we can meaningfully speak of Validation of a Code/Model, with the understanding that the Validation is restricted to the class of nearby problems, the parameter range for which ideally is demarcated as part of the Validation procedure. (An approach to extending the domain of validation by interpolation will be given in Section 11.12.)

9.10 DIFFICULTY OF THE OPTION TREE

The difficulty of the "option tree" growth, discussed previously for Verification of Codes, applies also to Validations, which are often inconclusive. For example, the option trees can include choice of turbulence models, wind tunnel wall effects, wind tunnel flow angularity effects, etc. (Melnik et al, 1996). Tuning of parameters in such models may be good for one set of experiments but bad for another. This regrettable situation is easy to criticize as poor science, but it is a fact of life for many engineering endeavors.

9.11 DATA SPARSITY AND LACK OF SYNCHRONICITY: GROUNDWATER, OCEAN/LAKE, AND METEOROLOGY MODELING

Westerink and Roache (1995) discussed some Validation issues in ocean modeling and other geophysical flows. A geophysical flow modeler typically encounters three major types of error to be accounted in comparing a numerical solution to the physical system.

1. Formulation errors are the result of missing physics and/or more often due to inadequate constitutive relationships.
2. Measured data errors due to inaccuracies in data acquisition and/or analysis techniques as well as data interpretation.
3. Numerical solution errors are incurred due to time and space discretization steps, boundary placement, boundary condition implementation, etc.

Of course, any physical measurement involves some error, and all fields of science and engineering are affected, but areas of geophysical modeling routinely encounter special difficulties with regard to

- data sparsity (groundwater, ocean/lake, and meteorological modeling), and
- lack of synchronicity (ocean/lake and meteorological modeling)

The data sparsity difficulty for groundwater modeling includes data on properties of the medium, which is usually heterogeneous, with properties often varying over orders of magnitude in a region of interest. The more universal difficulty (common to groundwater, ocean/lake, and meteorological modeling) is sparseness in *initial conditions*. If time-dependent simulations were started with only measured data, the PDEs would be grossly under-determined. This sparseness means that initial condition data must be "made up" by interpolation of the experimental data, which is typically sparse and non-uniformly distributed in space. This interpolation problem is usually treated by the family of interpolation methods called *kriging*. Kriging is a geostatistical procedure that uses a weighted moving average interpolation to estimate values at a large number of nodes (the computational grid) from sparse and irregularly spaced data. In a least-squares sense, kriging is the best linear *unbiased* estimator and reproduces the measurements exactly. (E.g., see Bras and

Rodriguez-Iturbe, 1985; Englund and Sparcs, 1991.) On the other hand, if the data itself has significant uncertainty (both in regard to the *value* of the variable and to its *location*), it may be preferable to use techniques that do not identically reproduce all data point values, but effectively smooth out a spatial distribution to reduce the effect of outliers.

Besides the data sparsity difficulty, ocean/lake and meteorological modeling involve another thorny problem of *lack of data synchronicity*. Not only are initial condition values sparsely distributed in space, they also are not all available at the same time. To further complicate matters, there often is no hard and fast *time* = 0 for the simulation. As new meteorological data is continually received, it must be incorporated into the model as a kind of continually updated initial condition data set. Note that one cannot simply inject new (later time) data into a computational grid, since it will undoubtedly be incompatible with the solution of the discretized equations at that point in space and time. This continual incorporation of new data into the simulation is a most challenging problem. For a readable overview of geophysical (meteorological) modeling problems, see Somerville (1996).

These three types of errors (formulation errors, measured data errors, and numerical solution errors) all affect the Validation exercise. It is common practice in geophysical problems to lump together all three errors and to only consider how well the numerical solution compares to the measured data. Physically based parameters are subsequently adjusted in the model to obtain "improved results". However, this approach can adversely affect the representation of the physical system.

Lumped error minimization can in fact give the modeler a false sense of model accuracy since it improves the comparison to the often limited measured data for a few variables, but can actually degrade the accuracy of all variables away from the data points or for variables which are not included in the error minimization. For example, in a tidal computation, tuning the model to fit free surface elevations can lead to a significant misrepresentation of the computed velocities. Furthermore, while tuning exercises can lead to satisfactory results for a select portion of the response spectrum, the lumped error/tuning process does not in general lead to good results over the entire spectrum. Again, this indicates that the basic physics of the process are not correctly represented. For example, in a depth averaged tidal model based on the shallow water equations, the bottom friction factor is often tuned to obtain a "best fit" for the dominant tidal constituent, yet secondary astronomical and nonlinear overtide and compound tidal constituents can fare relatively poorly (Westerink et al, 1989; Grenier et al, 1995).

Thus it is vital for the modeler to obtain separate estimates for numerical and data errors, allowing an estimate of the formulation error, assuming the total error is known. Only then can a reasonable assessment be made as to the appropriateness of the physics incorporated into the model. This will provide a basis for making adjustments in the physically based model parameters and/or the form of the constitutive relationships incorporated into the model. Making modifications in the formulation in this way will certainly lead to a much improved solution for all variables throughout the domain instead of only a few variables at select points within the domain.

Separately defining the numerical error also allows the modeler to define an improved spatial and/or temporal discretization, thus reducing the total model error. This can be especially useful in designing unstructured graded FEM grids. Finally, separate error assessments permit the modeler to compare the level of numerical and data errors and to judge what level of numerical accuracy and grid refinement is appropriate.

9.12 EFFECT OF PARAMETER RESOLUTION ON GRID CONVERGENCE

As the level of grid resolution is increased, the level of parameter resolution can also change, which complicates the Validation. See previous discussion in Chapter 8, Section 8.1.3.

9.13 SCALE OF UNSTEADINESS

In the previous chapter, we addressed the question of Benek et al (1996). "Let us assume that CFD can accurately compute the unsteady flow. If the steady state approximation is then used, will the same time averaged quantities be predicted?" The answer is "no" in general, but it is a question of scale, and the error bar for the approximation can be determined by computational modeling.

We can hardly expect more of the simulation than of the physics. For example, it is well known that pitot pressure probes, which give a single physically-averaged pressure reading, do not produce the average pressure in a fluctuating field, but a reading higher than the average; e.g., see Dolling (1996).

Even if one never used the steady-state form of the governing equations, but always used the time-dependent form, the question of scale of unsteadiness remains important, because of filtering effects of Δt and Δx. In principle, these ambiguities would be removed by grid refinement studies, but as a practical matter, small temporal or physical solution scales can be missed. It is probably not an overstatement to claim that virtually all practical flow problems are unsteady at some scale, even in the mean flow. (This even presumes a clear distinction between turbulence unsteadiness and mean flow unsteadiness.)

In regard to the present subject of experimental difficulties, the point is that these small scales are also missed in many *experimental* techniques. As Dolling (1996) stated: "A large fraction of the experimental database used for Validation of CFD predictions consists of time-averaged measurements. If meaningful conclusions are to be drawn from comparisons of these time-averaged data with computations, it is important to understand how the data are generated physically." Dolling's examples and his discussion are worth detailed consideration, as they bear significantly on very general questions of Validation.

As already noted, pitot pressure sensors effectively smooth a pressure in time, and the signal is not the same as the simple arithmetic average of the true instantaneous pressures. The pressure is also averaged in space (over the pressure orifice), introducing a similar error that can be significant in regions of flow with small-scale features, e.g. boundary layers. Flow visualization techniques involve a smoothing in space and time scales; for example, a schlieren photograph of a shock wave-boundary layer interaction will appear steady in a photograph with a common light source duration, but will disclose high frequency oscillations of the shock "foot" penetration (and associated high frequency velocity perturbations) when taken with short duration spark photography. Dolling (1996) described experiments in compression ramp hypersonic flows which ostensibly appeared to be steady, but in which mean flow unsteadiness (i.e., below the fluctuation scale of turbulence models) was a dominant phenomenon. The oscillations of the separation shock caused major intermittency in the pressure signal (~ 50% of the mean). Furthermore, the boundary layer separation line indicated by surface patterns (the kerosene -lampblack technique) is *not* near the center of the spread in the intermittent separation positions, but is at or close to the downstream boundary of the region of intermittent separation. The discrepancy is significant, as measured in the appropriate scale of boundary layer thickness. Three-dimensional tests on swept surfaces show that the separation line indicated from surface patterns moves upstream (in the intermittent region) as sweep is increased.

A major qualitative discrepancy between such experiments and the many computations performed over years is that the (steady) simulations always show a much steeper pressure rise (in the area of the shock-boundary layer interaction) than the experiments. Dolling pointed out that this is a natural and *unavoidable* outcome of the assumption of steady state in the simulations: there is *no* steady shock position that will produce the smoothed-out pressure rise of the experiments, which are due to temporal smoothing of a sharp shock position.

Although these examples of Dolling's have the common element of explaining computational - experimental discrepancy on the basis of the assumed steadiness in the computations, there is a fundamental distinction to be made in regard to the responsibility, between experimenter and computational modeler. First, it would have been preferable if all the experimental documentation had warned explicitly

and emphatically that the data were time-averaged and that instantaneous temporal variations were in fact large. Still, the CFD predictions should be expected to be able to model the unsteady flow, and then time-average the CFD results to compare with experiments. It would seem to be the responsibility (though a very difficult one) of the computational modeler to reproduce (within some useful tolerance) the averaged positions of the separation shock, and thus reproduce the time-averaged rise to the pressure plateau. The same is true of the time-averaged separation lines. However, there is one significant proviso: *that the unsteadiness is quantitatively inherent to the problem definition*.

The point is this: if the unsteadiness is actually caused by, or significantly modified by, un-reported experimental artifacts (such as fluctuations in tunnel operating conditions, free-stream turbulence, or model vibration), then the CFD prediction cannot be expected to anticipate these uncontrolled and unreported experimental variables.[109] The responsibility for the failed Validation in this case would rest with the experimenter. With some knowledge of fluid dynamics, most would agree that it is probable that the unsteadiness of shock position and separation position are in fact inherent to this problem, so that we would expect the steady state assumption used in the CFD simulation to be at fault, but it is also possible that experimental artifacts contribute something to the discrepancy. More pointedly, the problem Dolling described in regard to pitot pressure measurements is clearly an experimental artifact. Perhaps the CFD simulation could account for the spatial smoothing of pressure data in the boundary layer, but it would seem that the experimenter could more properly account for this in the data reduction technique. For the temporal smoothing (and "pumping" of the pitot pressure above the temporal mean) the CFD practitioner cannot be expected to account for all the relevant experimental artifacts, including (in one cited experiment) the "several feet of plastic tubing" connecting the wall pressure tappings to the scani-valve. This is experimental error, and the failure of the Validation exercise is not an indictment of the code or its assumptions.

For physical airfoils, there is always some scale of unsteady flow (evidenced by small-scale vortex shedding) yet steady-state equation solutions can be acceptably Validated. However, at higher angles of attack, the scale of unsteadiness increases, and perhaps because of this, the accuracy of Validation deteriorates (Jameson and Martinelli, 1996). Whether this discrepancy is viewed as a shortcoming of the conceptual modeling or as a criticism of the code, the Validation of the conceptual code/model combination has failed in this parameter range.

These and other examples from many areas of science demonstrate that the assumption of steadiness is not always justified and is scale dependent. Significantly, the assumption is probably made (implicitly) as often in experiments as in computations, and even more so in theoretical analyses. Recall that in most cases, wind tunnel flow is not the ultimate interest; rather, the wind tunnel flow is itself a "model" for an engineering flow of interest. Likewise, Darcy flow in a laboratory is a "model" for groundwater flows in nature, etc. In either case, the assumption may be invalid at the scale of interest, making Validation impossible or ambiguous.

Even if the intended application of an experiment (e.g., free flight) produces a steady state flow (at an acceptable scale), the experiment may not, since many experimental ground-based facilities are transient or border-line steady, especially for high enthalpy flows. Van Wie and Rice (1996) noted that "In general, more accurate measurements can be made in long-duration facilities as compared to pulse facilities." When purportedly "steady" experimental data really contain artifacts from tunnel or shock tube transients, a literally impossible burden is placed on the computational modeler.

[109] See also Section 9.20.

9.14 SPATIAL SCALES, SCALING UP, AND DIMENSIONALITY

Similar considerations apply to assumptions of physical scales and of dimensionality. It is well recognized that turbulence modeling involves a separation of scales, but this is true of virtually all macroscopic concepts. Laminar viscosity works as a concept only when the scale of molecular mean free motion is far removed from the physical size of the aerodynamic body, and fails in high altitude, high speed flows. At the other end of the flow spectrum, the measurement of permeability in a porous medium involves spatial averaging; if the scale of the variability is extreme, it may not be possible to adequately define an "average" permeability. Even qualitative features may not be accounted properly; as computational experiments show, a "scaled up" averaged permeability in strong velocity fields will require a tensor form, even though the underlying fine-scale permeability is a scalar (Durlofsky, 1991; Russell, 1996). Likewise, an elementary thought experiment will quickly disclose that alternating layers of isotropic material with differing heat conductivity will result in a non-isotropic (i.e., tensor property) material when "scaled up," i.e. when *either computations or lumped physical measurements* are made at a scale larger than the layer size.

The assumption of dimensionality can be equally misleading. The following examples of simple *laminar* flows demonstrate the problem. Williams and Baker (1997) showed that the often-reported under-estimation of reattachment length for a backward-facing step in incompressible flow above $Re \sim 400$ is due to the assumption of two dimensionality; their fully 3-D calculations agree with experiments. Likewise, the following examples were provided by Oberkampf et al (1995). Pironneau (1990) noted that nominally 2-D channel flow with a infinite periodic array of cylindrical obstructions exhibits a critical Reynolds number *Re* (based on channel half width) of $Re \cong 150$, above which the flow remains laminar and 2-D but becomes unsteady; above $Re \cong 600$ the laminar flow becomes 3-D and unsteady. Rudy et al (1991) noted that laminar hypersonic flow over a 2-D compression corner with a large deflection becomes 3-D. [It is not necessary to consider hypersonic flows - the classical problem of Goertler vortices (e.g., Saric, 1994) is a strong example of 3-D flow in an 2-D geometry.] Mittal and Balachnadar (1995) have performed "impressive computational work" to shed light on another classical and deceptively simple flow configuration, that of low *Re* flow perpendicular to a long circular cylinder. For $Re \lesssim 49$, the flow is steady and 2-D. For $49 \lesssim Re \lesssim 180$, only 2-D but unsteady flow exists. For $Re \gtrsim 180$, experiments had suggested that only 3-D unsteady flow exists. Mittal and Balachnadar (1995) shed light on this issue with simulations, which of course have the great advantage over physical experiments of controlled time-dependence and dimensionality. They computed the flow at $Re = 525$ using both a 2-D and a 3-D simulation. "They found that both solutions converged to a periodic solution, but the mean drag coefficient for the 2-D simulation was 1.44 and the 3-D simulation produced a value of 1.24. Experimental measurements yield a value very near their 3-D simulation value." Oberkampf et al (1995) noted the lesson involved, that the 2-D assumption might seem to be appropriate, and reasonable computational results are produced, but the results for drag are inaccurate *even for this very simple flow*. "With the change of one parameter (Reynolds number), over a relatively small range, three fundamentally different flow fields, i.e., solutions to the Navier-Stokes equations, emerge."

Even when a flow *can* be 2-D, it is a well-recognized fact of experimental life that it is difficult to achieve accurately 2-D experimental flows in planar geometries. (Axisymmetric 2-D flows are easier.)

Some problems do not scale up well, e.g. engines (Benek et al, 1996). Thus, the aerospace "vehicle performance modeling process becomes a decomposition of effects: the propulsion system characterized by full scale simulations and the aerodynamics characterized by scaled simulations."

In Aeschliman et al (1995), there are many points of agreement with the present book, notably their recommendation for physical experiments specifically designed for Validation. Experiments designed specifically for Validation of codes are quite different from those designed for other practical uses of the

data. They are easier in some ways, but more difficult in others. The reason for additional difficulty is the measurements necessary for simulations. The typical wind tunnel approach is to quantify small effects like flow angularity, and perhaps apply correction to data for lift, drag, etc. usually involving a shift in the measured angle of attack. However, this is not sufficient for point values of velocity components or pressures. What is needed is full measurement of inflow properties.

On the other hand, experiments for code Validation are easier in some respects. A very important point is that, for purposes of Code Validation, high quality wind tunnel flow characteristics are often *not* required, as long as they are measured and reported in detail. For example, consider non-uniform flow properties at the beginning of the test section of a wind tunnel. Variations in flow angularity of 5–10^{o} would be unacceptable for almost any engineering measurement, if the purpose were to predict free-flight quantities like lift, drag, etc. But if the purpose were to Validate a panel code (using inviscid equations), such flow angularity might well be acceptable, but *if and only if* the experimental map of flow angularity was given. Once so Validated, the code could then be used with some confidence for the prediction of the free-flight case, the assumption being that the wind tunnel flow, although too distorted to provide quantitative predictions of the free flight case, was still "nearby" and therefore covered the class of Validation problems for the code. Concerns with model blockage effects in wind tunnel tests are thus greatly reduced, if not virtually eliminated, for experiments designed only for Validation. For similar statements on this approach, see Oberkampf et al (1995) and Marvin (1995, p. 1784).

Aeschliman et al (1995) presented a "Case Study for CFD Code Validation Methodology" which is recommended reading for anyone serious about design of experiments for Validation. Also, good case studies were provided by AGARD (1988).

9.15 ASSUMPTIONS OF PERIODICITY

As noted earlier, the fundamental considerations regarding the solution qualities of spatial scales, scaling-up, dimensionality, and steadiness are not unique to simulations, but are equally significant to laboratory and field measurements and to theoretical analyses. Precise quantitative experimental techniques that give only local values often carry with them assumptions about these qualities. Global experimental techniques such as flow visualization are often less precise or simply qualitative, but are less susceptible to gross errors in assumptions. An example from experimental boundary layer transition studies (Knapp and Roache, 1968) illustrated a common assumption that is one level more subtle than an assumption of steadiness: the assumption of (regular) periodicity.

The most studied path to boundary layer transition involves the development Tollmien-Schlichting waves, which are a clearly unsteady phenomenon. However, the assumption is commonly made that they develop in a regular fixed pattern, i.e., the flow exhibits (regular) periodicity. In fact, some erratic behavior had been noted from the earliest studies, and experiments typically were designed to regularize this periodicity, through the introduction of small-amplitude periodic disturbances via single-frequency sound or vibrating ribbons. Even in quasi-natural (unforced) transition, flow visualization with still photographs appeared to show regular periodic traveling waves. Occasional photographic images showing no waves in the transition region could easily be attributed to experimental error due to drift of the smoke flow. But high-speed cinematography showed clearly that the unforced phenomenon was *not* regularly periodic, but in fact displayed a bursting of quasi-periodic waves. Predictions of skin friction and surface heating which would assume regular periodicity could be in significant disagreement.

9.16 OTHER DIFFICULTIES OF VALIDATION IN AEROSPACE

As noted in previous chapters, Verification for unsteady flows is not qualitatively different from steady flows; it necessarily involves more data, but is just more of the same process. However, Validation is much more difficult for unsteady flows just because of the paucity of data, e.g. Benek et al (1996). They also noted a clearly non-scientific impediment to Validation. "The major stumbling block to Validation of wind tunnel correction methodology is that the majority of data is either classified or proprietary, and therefore, not generally available." The same issues will occur in other than the aerospace industry.

9.17 UNIVERSAL TURBULENCE MODELS vs. ZONAL MODELING

A fundamental difficulty with Validation of turbulence modeling is the *essential* lack of a universal turbulence model. By *essential* I mean that the problem is not going to be solved. Not everyone agrees, and some continue the quest, but many turbulence researchers believe it is not going to succeed. (For example, see discussions in Kline et al, 1981.) Experience has shown that turbulence models that are moderately successful for a *very* limited class of flows (e.g., 2-D incompressible steady boundary layers in zero pressure gradient with no-slip walls) fail miserably and cannot be salvaged by tuning when the class is slightly expanded (e.g. the addition of an adverse pressure gradient to the k-ε turbulence model). The reason for the failure is fundamental, though probably not provable nor universally agreed upon. The "universal turbulence model" is already known, namely, the unsteady and small-scale Navier-Stokes equations. It is asking much of mathematics to devise a different and radically simpler set of equations that provide approximately the same solutions for all classes of flows, i.e. for all flow parameters and boundary conditions. The nearest thing to a viable universal turbulence model is the large eddy simulation (LES) approach, which requires so many fine-scale details that it still requires zonal turbulence modeling to be nearly practical.

In my opinion (as I expressed in Kline et al, 1981) the root of the problem is semantic; *turbulence* is just too broad a term to be meaningful. By creating the words "laminar" and "turbulent" and using them in juxtaposition, we can deceive ourselves into thinking that they have equal weight and meaning. In fact, "laminar" means something, but "turbulent" does not define the condition. If instead we used the term "non-laminar," the lack of specificity would be more obvious. Consider the folly of describing certain flows as "lacking laminarity" and expecting that condition to define the appropriate governing equations! There is just not enough information there to define the problem and to create alternate approximate equations. Using the word "turbulent" vs. "laminar" is like using the words "color" or "non-gray" vs. "gray." We do not expect the words "color" or "non-gray" to adequately define an artist's palette; we need words like green, mauve, fuchsia, ...

Conferences, symposia, and workshops on turbulence modeling that involve Benchmark comparisons with experimental data, especially if the data is not published beforehand (i.e., true *temporal* "prediction" vs. "postdiction") invariably have come to the following conclusions, perhaps stated more delicately.

1. Sloppy numerical work confuses the entire meeting.
2. "Every dog has his day," i.e. virtually every turbulence model can predict something correctly for some problem.
3. No single turbulence model predicts all features of a moderately complex flow.

This last evaluation is not the result of overly stringent accuracy standards. For examples, see Kline et al (1981), de Vahl Davis (1994), Pollard (1995), and European efforts (ERCOFTAC, GAMM, BRITE/EURAM, EUROVAL) mentioned by Pollard (1994).

The alternative to the hopeless quest for a universal model of non-laminarity is the "zonal modeling" approach espoused by Kline and others; see pro and con discussions in Kline et al (1981), Avva et al (1988). In this approach, turbulence models appropriate to each zone of a complex flow are used, e.g. a different model for an attached and approximately zero-pressure gradient boundary layer, another for a separated shear layer, another for swirling flow, etc. This approach leads to significant numerical and coding difficulties (as noted by Rizzi and Vos, 1996), e.g. specification of boundary conditions between zones, and the transfer of information through zone boundaries when different dependent variables are used in each zone, requiring transition/blending functions, etc. Ideally, a zonal modeling code would have a built-in "expert system" for adapting the zonal models to the developing flow solution, rather than requiring the user to define zone boundaries and select zonal models *a priori*.

However it is implemented, a zonal modeling approach greatly complicates each of the following:

1. the act of Verification of code,
2. the act of Verification of calculations, and
3. the act of Validation, even if the individual turbulence models have previously been Validated on single-zone flows.

9.18 Δ SPECIFIC AND GENERAL SENSES OF *MODEL* AND MODEL VALIDATION[110]

Model in a general sense (often termed a weak model) is the model form, or the general mathematical formulation, e.g. the incompressible Navier-Stokes equations, or the Fourier law of heat conduction. *Model* in a specific sense (often termed a strong model) includes all the parameter values, boundary values, and initial conditions needed to define a particular problem, e.g. Reynolds number, airfoil shape and angle of attack, or the conductivity and specific heat.[111]

In Chapter 2 and elsewhere, we noted that, in a meaningful sense, one cannot Validate a Code but only a particular calculation or at least a range of nearby calculations defined over some limited parameter range. However, we can also speak of Validating a Code and its underlying conceptual model in a more general sense and still be meaningful, provided that we make these distinctions.

We need the specific parameters and boundary values to run a simulation, so in a sense, we can only validate specifics. The same is true for experimental confirmation of physics theories, i.e. we only have specific samples of physical cases. However, when we have validated many specific cases, we generalize. It is understood what turbulence modelers mean when they say that the *k*-ε *model* has been validated (or is acceptably accurate) for attached boundary layers in favorable pressure gradients, but validation accuracy fails in adverse pressure gradients. The details will vary with particular cases (airfoils, *Re*, *M*, etc.) but there is a sense that the general *k*-ε model is validated in a range of parameter space, i.e. the domain of validation. Thus, one performs specific model validation which ultimately results in an ensemble general model validation or community-level acceptance of the general model.

---

[110] Material added to this Section of V&V1 is adapted from Appendix C of V&V20.

[111] The paper by Leijnse and Hassanizadeh (1994), written in response to Konikow and Bredehoeft (1992), was important for unraveling the semantics associated with the word *model*, in particular, whether *model* includes parameter values, and introduced the qualifiers *strong* and *weak*. They also discussed the related terms of "prediction models" and "analysis models." They asserted that not only is there no clear and unambiguous meaning of *validation* (neither in the general public nor in the scientific community), but also that there is no unanimity within the scientific community (in their case, the groundwater modeling community) on what constitutes a *model*. Also recall the ambiguity regarding inclusion of the *grid* in *model* discussed in Section 2.20.

Yet, validation of a model in the general (weak) sense may be a less daunting task than validation in the specific (strong) sense. For example, one may demonstrate that Darcy flow assumptions are valid for a geologic site, and even obtain quantitative prediction to an acceptable level of accuracy. However, if the strong definition of model is used, and the input includes uncontrollable factors (e.g., climate effects such as rainfall into the groundwater model) then one could argue [as did Konikow and Bredehoeft (1992), without benefit of this distinction of weak and strong definitions of *model*] that one could never hope to Validate the (predictive) model in this strong sense. Even then, we would argue that Validation in a meaningful sense is possible, depending on error tolerances, and especially if a single predictive calculation is replaced by more meaningful input parametric uncertainty study, e.g. Monte Carlo simulations that sample rainfall over historical ranges, etc.

In any case, whether comparisons of simulations with physical data are judged to be acceptable or not, the analyst would do well to acknowledge this distinction between strong (specific) and weak (general) definitions of *model* and to define or describe terms with sufficient precision. As Leijnse and Hassanizadeh (1994) asserted, these considerations "do not support abandoning the use of the term 'model Validation'. It only makes it apparent that the modelers have to carefully specify what their models constitute and what is being Validated." They quoted McCombie and McKinley (1993) who stated that if confusion arises about Validation, "use of the term Validation is much less to blame than poor science and sloppy documentation." It would be impossible to revise, and wrong to ignore, these contradictory existing practices so the context will have to guide the reader.

9.19 MYTH OF THE "TOTALLY VALIDATED CODE"

In discussing Verification in Chapter 8, we noted the "myth of the converged solution," in the sense that different variables can converge at different rates, and that it would always be possible to devise some error measure that is exquisitely sensitive to discretization error, so that this measure is far from converged even when other, more benign measures are well converged. Thus, in the pursuit of Verification of a Calculation, the concept of a "converged solution," ascertained to be so *independent of the intended error measure*, is a myth.

There is an obvious analog in Validation. Although we want to maintain that it is meaningful to speak of a "Validated Code/Model" (either a commercial code or a specialized scientific code) this has to be understood in the context

1. of a class of nearby problems,
2. for specified variables, and
3. for a specified level of accuracy.

Given a set of variables known to be of interest for an engineering project, one can follow the directive of Tjonneland (1988). "If you want a Validated Code you should look for the most critical part of the flow and for the toughest variable to get right and look at that." In the terminology defined in Chapter 2, this would certainly be appropriate for a Certification exercise, i.e. a project-oriented evaluation of a code, with understood variables and accuracy. But more generally speaking, the above 3 restrictions on the context must be recognized and honored.

For a real-world example, consider Validation of Code based on a k-ε turbulence model. Many Validation exercises have demonstrated that k-ε is good for attached boundary layer flows in zero and favorable pressure gradients, and is inadequate for adverse pressure gradients, swirl, etc. However, "good" is defined in the context of engineering needs, primarily friction drag (i.e., shear stress at the wall τ). While the k-ε model *predicts* turbulent kinetic energy k and dissipation rate ε, Validation is usually based on τ.

But suppose someone now wants to predict not just τ at the wall but the ε distribution throughout the boundary layer, say for applications to optical propagation through boundary layers (e.g., Truman and Lee, 1990). Validation for τ is encouraging, but it cannot be assumed that there is such a strong correlation that this can be taken for Validation of $\varepsilon(y)$ throughout the boundary layer. The calculation would have to be Verified for $\varepsilon(y)$ (first for iteration convergence, if that variable had not previously been tested in the iteration convergence criteria, and second for grid convergence) and then Validated for $\varepsilon(y)$. In actual fact, the k-ε model prediction of $\varepsilon(y)$ is inadequate.

Thus, the concept of a "Validated Code," ascertained to be so *independent of the intended variable measured*, is a myth.

9.20 § FRAUDULENCE IN FINANCIAL RISK MODELING

The world-wide financial disaster of autumn 2009 was due in significant part to failure of financial risk modeling. While ostensibly a failure of Validation of economic models, the blame probably should go elsewhere, depending on where one draws the line between an economic model *per se* and user input parameters as specified by modelers.

Quigley (2008) described the visit of a hedge fund manager to Moody's Investor Services, the credit rating agency that "had routinely declared what we now call toxic assets" to be AA-rated securities on the basis of economic models. The hedge fund manager inquired what would happen to these securities, all backed by home mortgages, if housing prices were to fall. The astounding answer was that the computer model would not accept such input.

If the risk model was built to not accept the possibility of declining house prices, then it is indeed a model error, and a gross one. An educated guess is that the restriction was added in a "wrapper" around the core financial model, and the wrapper was a lie. Investment experts certainly knew that real estate prices decline, sometimes precipitously, as in the world-wide Great Depression. And their "models" (the core models, not the user-friendly wrapper) would have predicted the outcome correctly; the system is unstable and would collapse because of leverage. But the *users* did not want to allow this possibility.

Financial models, like any other models, can be lied to. Financial models have worked very well, and made a lot of people rich. But lying to a model, or building a wrapper that accomplishes the lie automatically, is fraudulent.

9.21 § NEED FOR CONTROLLED AND MEASURED EXPERIMENTS[112]

Quality validation experiments are rare, at least in fluid dynamics and similar fields. The turbulent flow backstep problem of Driver and Seegmiller (1985) is often cited as an example of high quality experimental work, yet the organizers of the Lisbon III V&V Workshop (Eça and Hoekstra, 2008) realized upon careful reading that there was no measurement of inflow boundary layer properties at the upper wall, and no experimental uncertainty given for the time-averaged velocity components or Reynolds stresses. When the V&V20 project was begun, it was intended to use real experimental data on a heat exchanger for the example end-to-end problem of the document. No adequate experiment, with thorough measurement of inflow and outflow values and uncertainties, could be found. Instead, synthetic experimental values were used to illustrate the V&V procedures. When the follow-on documents to V&V10 (Computational Solid Mechanics) were begun in 2007, the original intention was to use real experimental data for a simple structural problem, but no adequate experiment could be found; instead, synthetic experimental values were

[112] From Roache (2008b)

used. The ASCE V&V document for free surface flows (Wang et al, 2009) has plenty of real and valuable experimental data, ranging from laboratory scale to site scale (e.g. sections of the Mississippi River, San Francisco Bay), but uncertainty statements are rare. In a 5-day workshop (Anon., 2008) on Nuclear Reactor modeling held in Idaho in July 2008, only one paper had any experimental uncertainty estimates; this work by Dr. H. McIlroy and colleagues (McIlroy et al, 2006) is an exceptionally high quality PIV (particle image velocimetry) experiment designed expressly for CFD validation, with complete flowfield measurements (including inflow) and complete uncertainty measurements. It is so rare as to be perhaps singular for these flows. The benchmark study for validation experiments in aerodynamics remains Aeschliman and Oberkampf (1997, 1998). Oberkampf and Trucano (2007, 2008) emphasized the importance and common neglect (in their considerable experience) of experimentalists performing measurement and documentation of all input quantities. In a 5-day Workshop on Combustion Modeling (Anon., 2006) held in Pittsburgh in February 2006, only one experimental paper (on stationary gas turbines at NSF) had complete description of inflow properties - not *uncertainties*, just *measurements* of the inflow properties.

The last example leads to a distinction between *uncontrolled* experiments and *unmeasured* experiments. Some experiments will fundamentally include uncontrolled parameters, e.g. atmospheric tests cannot control the conditions of the atmosphere. But it may be possible to measure the uncontrolled variables, at least at some coarse resolution, e.g. average wind speed, temperature, etc. Laboratory experiments are more controlled but still fail at some scale, producing variable results with repeat experiments. The worst situation is the unmeasured experiment. Of course, the quantities of interest [(termed "System Response Quantities" or SRQ by Oberkampf and Trucano (2007, 2008)] are measured but often the parameters defining the physics are not measured, e.g. inflow properties in a wind tunnel or compliance in a beam bending experiment. Note that an experimental procedure might be controlled yet unmeasured, i.e. the experimental protocol produces repeatable input parameters and therefore repeatable measurements, yet we do not know what are the values of these repeatable input parameters. This is a surprisingly common situation, and is much more serious than the lack of experimental uncertainty estimates. This is not merely a computational issue; such experiments cannot be repeated across facilities. The discrepancy between facilities, in those rare instances where they have been compared, is typically attributed to "experimental *bias* errors" but are often more fundamentally attributable to elementary lack of measurement of the controlling parameters of the experiment.

The majority of experimental works, at least in aerodynamics and fluid dynamics, not only fail to report experimental uncertainties but fail to measure all pertinent inflow quantities, so that results could not be compared with other *experiments* - forget CFD. Uncontrolled, unmeasured experiments are the norm, and experimental work is the weakest link in V&V.

CHAPTER 10

VALIDATIONS BY ERROR BARS

This Chapter title description "Validations by Error Bars" is chosen to differentiate the straightforward and customary practice of assessing Validation/Certification level by comparing, often graphically, experimental values *including experimental uncertainties* to computational values *including computational uncertainties*, from both less and more refined approaches. The inclusion of uncertainties distinguishes this approach from mere comparison of best experimental values (usually the mean of repeat calculations, all too infrequently corrected by an estimate of systematic error) to the best computational values (usually the finest grid solution, often without an estimate of error or uncertainty). The approach that ignores errors and uncertainties is naive, and deserves the somewhat pejorative term "viewgraph norm" (Oberkampf et al, 2002, 2004) although the approach still has considerable value. As argued in Section 2.3 and Appendix B, such comparison without uncertainty estimates no longer deserve the designation of full "Validation." Many of the Validations in this Chapter do deserve this designation, although many lack the inclusion of experimental uncertainties (no fault of the modelers); though incomplete, these semi-Validations may be the only source of information for engineering decisions. Publication standards would be improved if all were held just to this level. However, compared to Validations by Error Bars, the total Validation Uncertainty approach promulgated in ASME ANSI Standard V&V20 offers more clear insight into Validation and provides guidance regarding what can be done next. This total Validation Uncertainty approach will be described in the following Chapter 11. We do not suggest that the total Validation Uncertainty approach supersedes the Error Bar approach; both are valuable and informative.

The examples of Validation/Certifications (and semi-Validations), Calibrations, and Certifications in this Chapter are each presented with a philosophy or "moral of the story" in mind. Sometimes the moral is just the high quality of the work and/or the early date (i.e., the high quality obtained with earlier and therefore comparatively primitive computer power). More often, the example is chosen to elucidate some aspect of Validation/Certification exercises.

10.1 SOURCES OF PHYSICAL MODELING ERRORS IN AERODYNAMICS CFD

Oberkampf et al (1995) gave the following taxonomy of sources and list of non-exclusive sub-categories of physical modeling errors in CFD. Their emphasis is in aerodynamics, and the original paper includes worthwhile discussion based on their extensive experience.

- Physical Modeling Errors for Aerodynamics CFD
- Partial Differential Equations for Fluid Dynamics
  - Inviscid Flow
  - Viscous Flow
  - Incompressible Flow
  - Gas with Vibrationally Excited Molecules
  - Inert Gas Mixtures
  - Chemically Reacting Gas
  - Turbulent/Transitional Flow
  - Additional Physical Phenomena (e.g. thermal nonequilibrium, ionized flows, radiative transfer in gases, multi-phase mixtures)
  - Temporal Nature Assumptions
  - Spatial Dimensionality Assumptions
- Auxiliary (or Closure) Physical Models
  - Equations of State
  - Thermodynamic Properties
  - Transport Properties
  - Chemical Model, Reactions, and Rates
  - Turbulence Model
- Boundary Conditions for the Partial Differential Equations
  - Wall Boundary Conditions
  - Open Boundary Conditions
  - Free Surface Boundary Conditions

In addition, simple specification of geometry (even smooth surfaces) can introduce significant errors. Cosner (1995) stated "Serious issues are unresolved in transferring geometry efficiently from CAD systems to the CFD grid generators."

10.2 ACCURACY LEVEL FOR VALIDATION /CERTIFICATION

As previously discussed in Chapter 2 and more in Appendix B, some prevalent concepts of code/model Validation (though not our recommended use) involve an error tolerance or pass/fail accuracy level, determined "from the perspective of its intended uses" (Mehta, 1995). In all prevalent concepts of Certification (specific to a project) the required accuracy level is an essential aspect. In many of the sources cited, the term "Validation" was used with this sense of included error tolerance or accuracy level. Rather than change the terminology of the original papers from "Validation" to "Certification" or (just as misleading) to use the original term when we now want it to apply to "Certification," I have chosen the clumsy but transparent device of rendering the term as "Validation/Certification" when appropriate. (When the original is clearly consistent with Validation as defined herein without a pass/fail accuracy level, the word "Validation" is retained.)

A code/model could be convincingly Validated/Certified for one use but shown to be unacceptably inaccurate for another use, even if the physical problems are identical. (For example, see Van Wie and Rice, 1996.) Also, note that Validation/Certification will tend to be more precise, or have higher standards of accuracy level, for a purely scientific project that for an engineering project. As Rizzi and Vos (1996) noted, "in industry this [level of acceptable accuracy] usually is in the direction of 'just enough' certainty for the problem at hand." In considering the accuracy level appropriate for a Validation/Certification, it is also worthwhile to keep in mind that a simulation can have various expectations, depending on the application. In increasing level of accuracy requirements, the simulation may be expected (Benek et al, 1996) to

- only provide diagnostic information,
- supply only incremental data, or
- generate "base line data" for the performance model data base.

10.3 GENERIC MODELS VS. REALISTIC MODELS FOR VALIDATION/CERTIFICATION AND CALIBRATION: PHASES OF VALIDATION/CERTIFICATION

Marvin (1995) noted somewhat different Validation/Certification experimental needs for different stages of aerospace CFD code development: research codes, pilot codes, and production codes, referring to "building block and benchmark experiments." He noted the historical development of Validating computations (and theoretical approaches) "first, by performing small-scale experiments in ground-based facilities to Validate the approaches and any related extrapolation techniques and, later, by data from flight experiments with design hardware or prototypes." He noted that research organizations (such as NASA) tend to view Validation/Certification from a more fundamental level, stemming from a long history in CFD research and pilot-code development, whereas industry tends to view Validation/Certification in a broader sense, having to additionally consider the total design process in which the code is used. Wang et al (2009), in the ASCE V&V document for 3-D free surface flows, similarly distinguished between Validation for "unit processes" by controlled laboratory-scale experiments and Validation for the necessarily uncontrolled situation of coarse resolution field experiments (of which that document contains a wealth of information).

Although design engineers would prefer to see codes Validated and Calibrated for complex geometries, these increase experimental uncertainties. It is often preferable to perform Validation/Calibration exercises for simpler generic models, especially for new areas of investigation. Marvin (1995) cited an example of approaching the formidable challenges of Validation/Certification in hypersonic flow. A generic hypersonic vehicle model was tested to provide Benchmark data for Validating forebody codes developed for use in integrating air-breathing propulsion systems with vehicle airframes (Lockman et al, 1992). Marvin (1995) also noted the attraction of flight experiments designed with Code Validation/Certification as an integral part of their motivation, though he noted these are often difficult and expensive.

Marvin (1995) also provided an excellent example of a success story for experiments specifically designed for Validation/Calibration, rather than "practical" designs, in the development of the Johnson-King (1985) turbulence model. The experiment (Bachalo and Johnson, 1979) that provided much of the data for guidance of the development and Validation/Certification for transonic wing applications did not remotely resemble a wing. The test configuration was an axisymmetric cylinder with a circular arc section, aligned parallel to the flow; the motivation was to avoid extraneous 3-D effects. "The turbulence model derived made use of experimental evidence on the development of the turbulent shear stresses through shock waves of varying strength that developed on the circular arc section as the freestream Mach number was varied." The resulting Johnson-King non-equilibrium turbulence model has an intermediate level of complexity, more so than simple algebraic models (equilibrium models which have no "history"

dependence) but less so than two-equation models like k- ε and k-ω because it uses a single additional ordinary differential equation (i.e., 1-D) rather than multiple partial differential equations. For a small increase in computing burden and complexity over algebraic models, the Johnson-King model provided better predictions of pressure recovery, separation location and extent, and velocity profile development. Improvements to the original model (Johnson and Coakley, 1990) have been made, and the Validation/Certification has been extended to additional experiments (Marvin, 1995).

Similarly, Marvin (1995) cited the example of Menter's (1992) in developing a variation of the Wilkox k-ω model (e.g., see Wilcox, 1993) based on the reliable "building block experiments" on non-design geometries of Driver (1991).

Sindir et al (1996), like Marvin (1995), Melnik et al (1995), and Rizzi and Vos (1996), see Validation/Certification as a multi-phase process. Their code "Validation/Certification" process is perhaps closer to what others call Certification, but the meaning is clear, and Validation/Certification as used herein is certainly involved at all phases. Their four phases (see Figure 10.3.1) progress from

- Unit Problems (single flow features), to
- Benchmark Cases (more than one flow feature, simple to moderate flow physics), to
- Simplified/Partial Flow Path (multiple flow features, moderate to complex flow physics), to
- Actual Hardware (complete flow physics).

As the phases progress in geometric and flow complexity, the authors noted a tendency for decreasing availability and accuracy of experimental data.

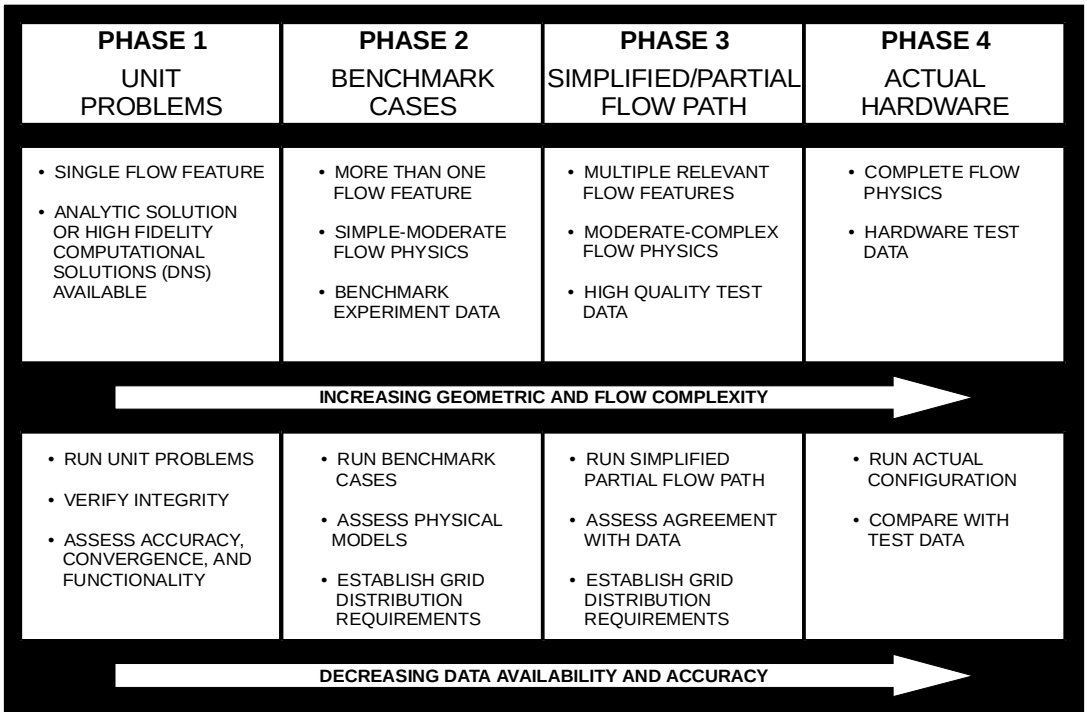


Figure 10.3.1. A Four-Phase View of Code Validation/Certification.
(From Figure 1 of Sindir et al, 1996.)

They also noted three levels of "Validation/Certification" depending on the use:

- Conceptual Design – Validated Code,
- Preliminary Design – Validated Code, and
- Detail Design – Validated Code,

with increasingly high standards of accuracy and specificity. Here, it is clear that their use of "Validation" corresponds more to other's use of "Certification." They gave a case study of this Validation/Certification process in the application of their UniFlo code to a turbomachinery dynamic loading problem.

Marvin (1995) cited the example of the experiments by Olsen and Seegmiller (1993) specifically designed for code Validation/Certification. The problem was transonic flow over a low aspect ratio wing. Rather than try to simulate free-flight conditions as well as possible, they tested the model in a solid wall wind tunnel, so that issues of corrections for transonic tunnel wall ventilation and interference were eliminated. The key feature is that conditions at the tunnel walls and free-stream inflow values were measured and included in the code Validation computation. (The modeling included viscous calculations at the tunnel walls by way of boundary layer calculations and a displacement thickness included in the CFD code being Validated.) In such exercises, once the basic code is Validated, practical calculations can be obtained with better far-field boundary conditions (which themselves must have been Verified and/or Validated).

10.4 CFD AND EXPERIMENTAL FACILITY CORRECTIONS

Benek et al (1996) gave an overview of CFD successes in providing corrections (Calibrations) for experimental facilities. (See discussion of "Trends, Computational and Experimental" in Section 9.7.) Sickles and Erickson (1990) demonstrated excellent agreement between measured and computed corrections to an aircraft configuration. The calculations of Martin et al (1993) of increments of normal force and pitching moment on the space shuttle launch configuration "agreed well with measurements at several model scales." Willhite et al (1995) computed sting interference corrections that "compared well over a wide range of flow conditions." Hinkleman (1995) made "unpublished comparisons of drag increments accounting for a mismatch of wind tunnel and flight Reynolds number" that were "in excellent agreement with the measured values" from a "conventional" vs. a "high Reynolds number wind tunnel." Benek et al (1996) noted the potential for CFD to "make practicable the evaluation of increments that can not be readily obtained experimentally," such as scale model geometric compromises (like oversized boundary layer gutters in wind tunnels), and Reynolds number corrections for inlet swirl and total pressure maps. However, they also noted "The major stumbling block to Validation of wind tunnel correction methodology is that the majority of data is either classified or proprietary, and therefore, not generally available." The use of CFD in the operation of *adaptive* wind tunnel walls is discussed briefly in Section 10.6.4 below.

10.5 VERIFICATION MUST BE INDEPENDENT OF VALIDATION: AIRFOIL CALCULATIONS

In Chapter 2 and elsewhere, the importance of performing Verification of Calculations *before* Validation was stressed. Logically speaking, the actual time sequence would be immaterial, of course; the point is that Verification is required regardless of agreement with experiments. But people can get led astray with fortuitous agreement with experiments, and are tempted to not bother with Verification of

Calculations. Zingg (1992) has pointed out the importance of this precept in his airfoil calculations involving thorough and systematic grid refinement studies (previously cited in Chapter 6 as examples of good Verification practice). In 4 out of the 7 flow cases calculated, agreement with experiment was better with coarse grid calculations than with fine grid calculations. This would have given a "false positive" Validation/Certification, caused by inadequate grid resolution, had not the systematic Verification been performed independent of agreement with experiments.

Rizzi and Vos (1996) noted the same phenomenon, and identified the reason. The physical model used includes the Baldwin-Lomax turbulence model. Being algebraic (a "zero-equation" model), there is no lag in adjustment to new boundary layer edge conditions. This may be an adequate approximation for slowly varying edge conditions, but is not for transonic flows with shocks. Physically, these flows require a small distance aft of the shock for turbulence equilibrium to be attained. Quite by accident, spatial discretization provides such a lag in adjustment. The longitudinal discretization increment Δx thus provides a *non-physical* parameter that behaves something like a physical relaxation length. The agreement with experiment occurs only from this numerical artifact, the computational error happening to partially cancel the physical modeling error in these cases. As Rizzi and Vos say, "Issues related to the interaction of these two sources of errors have been of constant concern to the CFD and turbulence modeling communities because fortuitous cancellation of errors can lead to a very erroneous conclusion about Validation."

Clearly, to jump to a conclusion about Validation without thorough Verification is to build a house on sand.

10.6 SYNERGISM BETWEEN COMPUTATION AND VALIDATION EXPERIMENTS

Aeschliman and Oberkampf (1997, p. 14) noted that beneficial synergism arises when experiments are designed specifically for Validation. "By a 'synergism', we mean an activity whose primary intent is to meet a requirement for one approach, whether CFD or experiment, but which generates improvements in capability and/or accuracy of the other, such that both computational and experimental methods benefit." Presented below are instances of such synergism.

10.6.1 Artificial Heart Valves

In pioneering work on simulation of artificial heart valves, Mueller and colleagues (Mueller, 1974; Underwood and Mueller, 1977,1979) showed the advantages of an integrated numerical and experimental program. Over the multi-year program, four different experimental flow visualization techniques (electrochemical, dye, hydrogen bubble, and nylon micro-spheres) were used to Validate the numerical solutions. Even using 1972 vintage computers such as the UNIVAC 1107 and the IBM 370/155, this group managed to perform reasonable grid convergence testing, i.e. Verification as well as Validation. "The overall flow patterns as well as the occurrence, location and extent of separated flow regions obtained numerically agreed very well with the experimental results for the laminar flow cases." (Underwood and Mueller, 1977.) The advantage of the integrated experimental/numerical program worked both ways; "the numerical results were found to be very helpful in positioning the hot-wire and hot-film anemometer probes in later physical experiments." (Underwood and Mueller, 1979.)

10.6.2 Transonic Flow

Marvin (1995) cited another interesting example of synergism between computations and Validation experiments when cooperative efforts are made. The related papers by McDevitt et al (1976), Levy (1978) and Marvin et al (1980) involved experiments and simulations of transonic flow over a circular arc airfoil.

Airfoil pressures, tunnel wall pressures, and flowfield velocity profiles were measured experimentally with a (non-intrusive) laser anemometer system. Depending on the experimental Mach number *M*, steady flow or unforced buffet flow developed. The buffet onset condition displayed hysteresis; the buffet *M* domain was different for experiments during which *M* was increased or decreased. Computations using a time-accurate Navier-Stokes code and a simple turbulence model were performed at three *M*. At the intermediate *M* = 0.755 the computations were not converging. The experimentalists pointed out that the flow was unsteady at this *M* for experiments wherein *M* was decreased once buffet occurred, and that perhaps the computation was in fact reproducing the correct physics. With this motivation, the computational modelers continued the simulation for a time equivalent to several airfoil chords of flow travel, and indeed periodic buffet developed in the simulation. The magnitude and frequency of the computed buffet "compared well with the experiment, although the frequency was found to be about 20% less than the experimental value." According to Marvin (1995), "These studies confirmed for the first time that time-accurate solution methods used to solve the Reynolds-averaged Navier-Stokes equations had the potential for predicting buffet."

10.6.3 Simple-to-Complex Geometry Flows

Aeschliman and Oberkampf (1997, p. 14) also presented an example of beneficial synergism between CFD and experimentation in the Validation activity in their long-duration hypersonic facility.

"If in a wind tunnel experiment the wind tunnel model is designed for easy modification from geometrically simple to complex, it becomes possible to produce a wide range of flow conditions. The geometrically simple flows could possibly be calculated with high confidence, while the complex geometry flows may exceed the current computational state of the art. As an example, for attached, perfect gas, laminar flow over a slender sphere/cone at low angle of attack, confidence in the computed solutions for flow over the simple model with simple flow physics can be at such a high level that the results are usable for an *in-situ* calibration of the freestream wind tunnel flow. This type of calibration can provide new, and sometimes surprising, information about the facility. For flow over more complex geometries, the measurements can be used to Validate the code."

10.6.4 Operation of Adaptive Wind Tunnel Walls

The use of CFD in the operation of *adaptive* wind tunnel walls and in correcting for wall and model support interference is described by Aeschliman and Oberkampf (1997, p. 14) as "a synergism that has a large potential payoff. It is desirable to test aircraft configurations at the largest possible scales to maximize Reynolds number, a goal which is in immediate conflict with minimizing interferences." That is, the larger the wind tunnel model, the better is the Reynolds number, but the worse is the tunnel interference. The interference effects can be calculated (and removed from the experimental results) by CFD. These authors and others noted that the "CFD capability required to compute interference corrections must advance in concert with the testing requirements."

Likewise, Aeschliman and Oberkampf (p. 14) stated "In a similar vein, advances in the use of CFD to compute flows in perforated-wall wind tunnels are retarded by a lack of well characterized wall boundary conditions. Detailed measurement of the actual wall boundary conditions as a function of test section location and given tunnel operational parameters would directly improve wind tunnel data accuracy, in addition to providing the needed [boundary conditions] for a CFD calculation."

10.6.5 Boundary Layer Transition

The paper by Haynes, Reed and Saric (1996) presented an excellent example of Validating a reduced set of equations, the nonlinear Parabolized Stability Equations, for 3-D crossflow-dominated transition by direct comparison with extremely carefully performed wind tunnel experiments. This paper contains many insights and suggestions on the general topic of Validation, and especially provides an excellent instance of synergism between computation, theory, and Validation experiments.

As noted in Chapter 2, a major opportunity for computational simulation technologies like CFD is to contribute to experimental work being used to predict the importance of difficult-to-measure quantities like wind tunnel flow angularity, non-uniform stagnation enthalpy, surface waviness, etc. (See Aeschliman et al, 1995.). A premier example of this given by Haynes et al is the sensitivity of boundary layer transition to free stream vorticity, which is "an unusually difficult experiment" but which can be modeled by CFD. Surprisingly, when streamwise vorticity is not included in the simulations, "the subharmonic mode is present as predicted by theory but not seen experimentally. When streamwise vorticity (as is present in the flow from the turbulence screens upstream of the nozzle) is also included, the subharmonic mode is overshadowed by the fundamental mode (as in the experiments!)"..."Here is a case in which the computations have explained [long standing] discrepancies between theory and experiments."

Haynes et al (1996) also demonstrated conclusively that, although linear theory predicts many features, a nonlinear calculation is required to obtain complete Validation for the growth rates for stationary cross-flow waves.

Other examples cited in Haynes et al (1996) are the computations by Fasel (1990) in which the CFD simulations isolated the significant effect of a small streamwise pressure gradient, and the CFD simulations by Joslin and Street (1992) in which subharmonic modal discrepancies with experiments were eliminated by including a small adverse pressure gradient and a small effective frequency variation in the input disturbance.

Haynes et al (1996) also suggested an innovative and synergistic use of CFD simulations in conjunctions with experiments. Their simulations have shown that free-stream vorticity has a major effect on transition, yet free-stream vorticity cannot be measured! "At first glance, the situation seems hopeless. How do you quantify an effect you can't even measure? How do you ever hope to establish [an experimental] database for CFD Validation when you can't provide precise initial and boundary conditions?" Their solution is to systematically apply CFD (spatial DNS) simulations to study the effect of different conditions along edges of the computational domain, "identifying those quantities in the boundary layer that are indicative of the forcing." The key is that "amplified quantities in the boundary layer are often larger than their predecessors in the freestream and therefore measurable, thus providing a database." They call for "CFD leadership in the identification, cataloging, and modeling of the effects of freestream disturbances."

The subject area of direct spatial simulation of boundary layer transition is one of the few that proves difficult from the aspect of computer round-off error. From Haynes et al (1996): "Because of the long fetch from the onset of instability to breakdown and the large amplitude ratios associated with this process [$O(e^{10})$ and larger], resolution and bit accuracy limit how far into breakdown a spatial simulation can go. Because of the large differences in amplitudes throughout the domain and the large growth rates known to exist near breakdown where smaller scales appear, truncation and round-off errors can easily contaminate the solution. Consequently, spatial simulations are currently unable to proceed completely through transition and into turbulence."

This area of fluid dynamics (boundary layer transition) can also be sensitive to outflow boundary conditions, which (as noted in Chapter 2) might appear to be a "gray area" in the distinction between Verification and Validation, but, in the final analysis, is best considered part of (the purely mathematical

area of) Verification. Haynes et al (1996) suggested five exercises for testing [Verifying] a CFD boundary layer transition code:

a. grid refinement studies,
b. solving test problems for which the solution is known,
c. changing the 'far-field' boundary conditions systematically and re-solving,
d. comparing linear growth rates, neutral points, and eigenfunctions with linear stability theory,
e. running the unsteady code with time-independent boundary conditions to ensure that the calculations remain steady.

They also require that the "manufactured solutions should be chosen with topological qualities similar to those anticipated for the solution to the 'real' problem (e.g. gradients close to the wall)." As noted in Chapter 3, We do not believe this is a necessary feature mathematically, although it is desirable from the viewpoint of intuition and confidence building.

These studies from Haynes et al (1996) are powerful examples, not only of Validations of computational PDE codes, but also of the power of simulations when performed very carefully, with rigorously Verified codes, in conjunction with high-level theory and highly controlled experiments. These results could not have been attained by haphazard application of commercial codes by unqualified engineering personnel; such exercises could only serve to pollute the literature, and degrade the reputation of computational PDEs. Likewise, sloppy experiments do much more harm than good. As Haynes et al (1996) noted for boundary layer transition (but the observation holds in many other areas of fluid dynamics, heat transfer, and other areas of applied science), "Validation requires comparison with careful archival experiments, but few such experiments have been performed." Computations can contribute to the "determination of relevant Validation experiments."

10.7 DIFFICULTY OF DEFINING A "NEARBY" PROBLEM

As noted in Chapter 2, we would not want to be so scrupulous as to insist that only a single Calculation can be Validated. However, the concept of Validation of a Code, rather than just a single calculation, is meaningful *provided* that the evaluation of "Validated" is applied only to a range of "nearby" problems, close to the discrete set of experimental Validation parameter space. It was also noted that the concept of a nearby problem will fail near parameter transition boundaries, which should be identifiable from knowledge of the general field (see list in Chapter 9, Section 9.9.). However, the following example exemplifies how difficult this can be.

Consider the effect of chemistry modeling on aerodynamic coefficients. Without trying to decide *a priori* by theoretical considerations whether or not chemistry is important, one might reasonably assume that, if chemistry could be shown to be insignificant in a given parameter range for one aerodynamic coefficient, it would also be insignificant for another coefficient at the same parametric conditions, i.e., surely this would constitute a nearby problem. Unfortunately, this is not always the case. Marvin (1995) gave an example of experimental data on sharp and blunt cone models fired in a hypervelocity gun range, compared to two sets of calculations, one assuming a perfect gas, and one using non-equilibrium chemistry. High enthalpy, reacting but laminar flow was established over the model, so chemistry was an issue, but turbulence modeling was not. For a 5^o blunt cone, perfect gas calculations for drag were within the experimental uncertainty and similarly close to the nonequilibrium chemistry calculations. However, the data for drag are insensitive to chemistry modeling only because drag is composed mainly of forebody pressure drag, whereas the flowfield generally, and the pitching moment C_m especially, prove to be sensitive to chemistry. The error in C_m for the non-equilibrium air calculations (full Navier-Stokes and seven species air) was within the estimated experimental error of ±3% over the entire range of angle of attack α from 0°

to 8°, but the error for the perfect gas calculation at $\alpha = 5°$ is an order of magnitude larger, ~ 30–35%. This is indeed a sobering example of the difficulty of defining a nearby problem for Validation of a Code.

10.8 MISSING EXPERIMENTAL INFORMATION

Barber (1996) provided a review of several code Validation/Certification studies in the aerodynamics literature, and cited examples wherein missing experimental information have lead to poor Benchmark calculation comparisons. The missing information includes

- the geometrical definition,
- the data reduction procedures, and
- the dominant physics.

Advances in experimental techniques can now supply details of flow fields that were impossible in the past. For example, Eklund et al (1995) performed Validation of the SPARK Navier-Stokes code predictions of Mach 2 flow over a rearward-facing step. This Validation was unique in that the experimental techniques used (Laser-Induced Iodine Fluorescence and Laser Doppler Anemometry) produced not only velocity data but also in-stream static temperature measurements. McIlroy et al (2006; see also Condie et al, 2005) used PIV (Particle Image Velocimetry) in an MIR facility (Matched Index of Refraction) to provide detailed measurements of a highly 3-D unsteady turbulent flow in a physical model of a lower plenum in a gas-cooled nuclear reactor, including inflow with strong nonuniformities and non-normal velocity components, which are essential to defining the problem to be modeled in a Validation exercise. Significantly, *all* the measurements include uncertainty estimates, a rarity.

Hutton (2006) gave an example of a code comparison exercise on turbulence modeling that failed for lack of such experimental measurements. As is often the case, a wide range of results were obtained, even between groups using the same "code." As expected, the difficulty was a familiar one: *code ≠ model*, and the experimental problem specification was not complete because of unspecified radial velocity at inflow. Unlike the Third Lisbon V&V Workshop (Eça and Hoekstra, 2008), "very few" of the contributors performed grid convergence studies. (Hutton referred to one university group who "actually" refined the grid!) This presented an excellent example of a K-ε calculation that agreed better with experiment for a coarse grid than a fine grid, not just with a single-valued solution functional but over a distribution. The systematic cancellation of discretization error and modeling error occurs because K-ε overproduces turbulent kinetic energy while under-resolution reduces it. This is yet another warning against rushing into Validation without first ascertaining grid convergence in Calculation Verification. With his experience, Hutton also affirmed the fact, widely acknowledged in the V&V community but always requiring reiteration, that old experimental data is generally inadequate for good Validation exercises. He presented some examples of a Best Practices Guideline and a QA system for such code exercises, accessible at www.QNET-CFD-KB.com.

Roy and Blottner (2000, 2001) performed exceptionally thorough Code and Calculation Verifications for 2-D hypersonic flows with several 1- and 2 - equation turbulence models. The Validation component was limited by the virtually non-existent experimental uncertainty estimates from a 1968 free flight test at Mach 20 at 24.4 km (80,000 feet) altitude. The atmospheric conditions had to be taken from a standard atmosphere, and not even the geometry was known precisely because the ablated nose tip shape came from other simulations. The experimental condition included a small (0.14 degree) angle of attack while the computations were axisymmetric. [Nevertheless, this experiment with major uncertainties but true conditions was more valuable than laboratory experiments with better uncertainty estimates but wrong conditions.] The Validation levels for surface heat transfer were convincing and, for some of the models, successful.

See also Strazisar and Denton (1995) and discussion in Section 12.12.

10.9 ONSET OF 3-DIMENSIONALITY IN BACKSTEP FLOW

The experiments of Armaly et al (1983) showed the abrupt onset of 3 dimensionality in the nominally 2-D laminar incompressible flow over a backstep. These trustworthy experiments have been used as the basis of comparison for many CFD Validation exercises, mostly for the 2-D flow regime ($Re < 400$). The 2-D CFD predictions for reattachment length are in good agreement with the experiments up to $Re \sim 400$, but predict a longer than observed reattachment length for $Re > 400$, evidently due to the onset of 3-D effects in the experiments. Conjectures on the causes of the 3-D flow (besides possible experimental flow or other errors) included Taylor-Goertler instability associated with the concave streamlines near reattachment, and tunnel side-wall boundary layer growth, the latter being rejected due to discrepancy between the proposed mechanism and experiment as *Re* is increased (Ghia et al, 1989).

The actual mechanism was determined by 3-D calculations by Williams and Baker (1997). This carefully performed study (involving 7 levels of grid refinement, and evaluation of sensitivity to entrance length) first of all showed good agreement with experiments for both 2-D and 3-D flows, thus Validating the numerical solutions (using the finite element Continuity Constraint Method). More interestingly, the detailed flow calculations disclosed a previously unrecognized flow feature, that of a wall jet at the backstep plane that grows in strength with increasing *Re*. This wall jet is the source of 3-D vortices, and the Lagrangian particle tracks of the simulation "reveal a fascinating picture of very complex three-dimensional flow structures."

This study by Williams and Baker (1997) shows the power of carefully performed simulations applied to trustworthy experimental data, and guided by sound theoretical reasoning, to convincingly explain complex phenomena.

10.10 GRAY AREA: "VALIDATION" FROM A CALCULATED BENCHMARK

A gray area exists in using a DNS (Direct Navier-Stokes) computational data base for Benchmarking "Validation" of simpler turbulence models. Should this be called Verification or Validation? Apparently, confidence in the governing equations and the solution (obtained with highly accurate spectral or mixed spectral and high-order FDM) is so high that agreement is taken as a Validation. We are reluctant to confuse the mathematics vs. science distinction of Verification vs. Validation, but we have no reluctance with the word "Benchmark."

10.11 GRAY AREA: "VALIDATION" OF AN EXPERIMENTAL TECHNIQUE BY A COMPUTATION

In Newling et al (1997), a comparison was made between experiment and computation with the object not of Validating the computation, but "Validating" the experimental technique!

The experimental technique was velocity determination by Magnetic Resonance Imaging (MRI). In fact, it is difficult to determine what is the cause for discrepancies - the CFD code, the grid, the experimental flow (especially symmetry problems), the "seeding" of the flow with particles that respond to MRI, the MRI data acquisition, or the MRI data reduction, which involves significant calculations. The authors took the CFD computation as the standard, but in such a case there is no clear-cut "Benchmark" and no certainty of what is being Validated or invalidated. Nevertheless, the overall agreement (at least

qualitative) builds some confidence in the experiment, the MRI technique, and the simulation, at a discrepancy level appropriate for a new technology.

Another approach to Validating a new experimental technique by way of simulations, with more easily defensible results, is that of Jumper and Hugo (1995); see also Hugo et al (1995) and Jumper et al (1994). The paper described the small-aperture beam technique, which is a sparse-instrument method for quantifying the instantaneous optical wave front distortions caused by propagation of a laser beam through an optically active (aberrating) turbulent flowfield. The method takes advantage of the fact that the structures on the wave front *convect* through the viewing aperture. A numerical simulation of a 2-D heated jet, using a discrete vortex model, was used to "explore the validity" of using beam-jitter signals from multiple probe beams to obtain a measure of the distortion (optical path difference, or OPD) in a flow region where eddy production constitutes the major character of the turbulent flowfield. The concept involved is that the numerical simulation provides a Benchmark of high-resolution space and time data for the numerical quadrature of the optical propagation. In this situation, high fidelity between the simulation and a physical experiment is not strictly required. Since "the essential character of the experimental flow appears to be captured by the numerical flow," the numerical data base can be used to "Validate" the experimental technique.

Although these types of studies cloud the distinction between Validation and Verification, their value is indisputable, semantics aside.

10.12 THE MADE-2 EXPERIENCE: CAN GROUNDWATER FLOW MODELS BE VALIDATED?

In Chapter 2, Section 2.2, the paper by Konikow and Bredehoeft (1992), "Groundwater Models Cannot be Validated," was criticized for its excessive pessimism. However, it is true that error tolerance in groundwater flow and transport modeling is huge compared to other fields. Rigorous Validation sometimes seems to be an insurmountable problem for field data on groundwater flow, and some sites are worse than others. A well known difficult case is the MADE-2 experiment.[113]

MADE-2 (for MAcroDispersion Experiment) was a natural-gradient experiment in which water and dissolved contaminants (tritium and four organics) were injected into an unconfined aquifer in a two-day pulse and monitored for 15 months. The tritium provided a conservative tracer (its decay with a 12.26 year half life was easily corrected) while the non-conservative organics (realistic components of fuel spills) exhibited sorption and proved to experience significant natural (probably biological) attenuation, which is highly beneficial to prospects for natural remediation techniques. MADE-2 was unique in two respects: the density of the experimental resolution in both space and time, and the heterogeneity of the site. A 3-D sampling network tracked contaminant plumes, and other extensive field measurements make it one of the most completely characterized sites in the open literature (Boggs et al, 1990). Compared to other experimental field sites, this one is highly heterogeneous. Significantly, it is not known at the present time if this is non-representative of nature in general, or just of the instrumented field sites. Which is more representative of a typical contamination site is hard to say. What is clear is that, given the limits of present field test measuring capability, no combination of Tuning of model parameters, reasonably constrained by in-situ field measurements of properties, and within the conceptual model limits of isotropic Darcy law flow, will reasonably match the field measurements, without introduction of *ad hoc* assumptions.

[113] Boggs et al (1992), Adams and Gelhar (1992), Rehfeldt et al (1992), Boggs and Adams (1992), MacIntyre et al (1993), Stauffer et al (1994).

As dense as the field measurements were (e.g., there were 2500 measurements of hydraulic conductivity in 77 different boreholes), there is room for improvement with conceivable developments in instrumentation. Although permeability measurements were dense, porosity measurements were scarce. (Only one measurement was made of specific yield.) These affect transport calculations for time-dependent flow. Also, the field measurement technique used to determine permeabilities is considered to be state of the art, but nevertheless has a significant limitation.

The borehole flowmeter technique involves injection of water through a probe, and measurement of resistance to infer permeability. Vertical resolution is achieved, so the data is 3-D. This is important for geological formations, since geological layering causes significant heterogeneity (as well as isotropy) in the vertical direction. Of course, there is some inevitable smoothing associated with the data, being inherently averaged over some volume of space of the scale of the probe itself. The more fundamental limitation is the inherent averaging in the circumferential direction. This means that only an effective scalar (non-directional) permeability can be measured. If the site actually had significant tensor properties (i.e., directional preference in hydraulic conductivity) this could not be detected because of instrumentation limitations. Recall the statements of Section 9.3: "A conceptual model is needed prior to experimental data gathering" and "Every observation is laden with theory." The step of converting the raw data into permeabilities *assumes* a Darcy Law formulation with scalar (non-directional) permeability. Nor could the data be re-interpreted with a different assumed form of the law, since the experimental technique has no directional sensitivity.

Likewise, the inference of dispersivities is obviously dependent upon the assumed conceptual and mathematical model, and upon the scale of definition, which clouds the border between continuum definition and discretization (Roache et al, 1997).

The well known vertical directional properties (due to geological layering) were probably not a major problem, because the horizontal resolution was adequate. In the absence of vertical flow data, groundwater modelers typically use a factor of 10 vertical heterogeneity, i.e. a tensor property with vertical permeability assumed to be 1/10 that of the measured or otherwise modeled horizontal permeability. For the MADE-2 site, an average value of vertical anisotropy was calculated as 0.18. The results are usually not very sensitive to the ratio, since vertical flow is a secondary effect. In fact, most groundwater flow simulations are performed with a 2-D code in which the vertical layer is represented with a single control volume, and no vertical flow is calculated. ("Vertical" is also a simplification; the small angle approximation is used, so that no flow is calculated through the top and bottom of the layer, but the layer may be inclined slightly to the horizontal.) For the MADE-2 experiments, we found (Roache and Rucker, 1996; Roache et al, 1997) that in order to obtain good fit with the experimental plume data, it is necessary to use highly anisotropic horizontal conductivities (by factor of 15). These were not substantiated by field measurements, nor can they be because of the circumferential averaging implicit in the field instrumentation. However, the possibility of anisotropy was suggested by aerial photographs of the site that showed some indication of previous streambeds in the area, and by some well draw-down tests. Clearly something is going on, since at the scale of measurement, the observed flow paths (inferred from contaminant tracers) did not follow the head (pressure) gradients; no amount of Tuning with scalar properties can match this behavior.

Whereas improvement is conceivable with new instrumentation that would allow measurement of tensor properties, it is also clear that there are limits of what can be predicted with mathematics. If standards for precision are excessive, it is true that "groundwater models cannot be Validated," but neither can anything else. As stated previously, the concept of Validation must include an acceptable error tolerance or level of precision. It is also true that groundwater flows will always have an error tolerance large compared to most areas of science and engineering.

This does not mean that groundwater modeling is useless, only that it must be used with commensurate tolerance. The "predictions" may be of very coarse precision, but still useful compared to no predictions at

all. Even very long term simulations can be useful, e.g. the 10,000 year simulations *required by law* for low-level radioactive waste disposal[114]. Such exercises properly are not conceived as "predictions" at all, but as rational bounding exercises. No one knows if or when some luckless future wildcat operator will drill into the repository; nevertheless, a rational probabilistic study (including geostatistics) can be made to bound the consequences and rationally and quantitatively (vs. emotionally) evaluate the risks and costs, and allow comparisons to alternative disposal concepts.

As quoted in Chapter 2, Konikow and Bredehoeft (1992) asserted that "In the end, action concerning waste disposal will be a judgment; a professional judgment by the scientific community and a judgment by society." As discussed in Chapter 2, Section 2.2, such replacement of modeling by "judgment by the scientific community" or scientific expertise is nonsense. Decisions must be made quantitatively; we need numbers, whether or not the numbers come from codes. If the decision is based on a classic "back of the envelope calculation" based on a simple theory, then that is the *model*! The only difference between such a calculation and a computer code simulation is that the "back of the envelope" model is incredibly limited, and makes much *less* defensible approximations and assumptions than the computer model. Once the knowledge of the geophysics scientist is put into usable quantifiable terms, it *is* a *model*. And decisions cannot be made on non-flow-model geological data. The type of information dear to the hearts of geologists - the age of the rocks, or the color of the rocks, or the chemical composition, or the description of earth processes that formed the rocks - even if these data were quantitatively accurate, would not answer the question of the suitability of the site or of a proposed remediation technique.

Groundwater flow modeling is not a choice, it is the *only* approach. The only remaining question is the scale of the "modeling," from a suite of thousands of 3-D time-dependent computer simulations with $O(10^6)$ cells over 10,000 years, to a "back of the envelope" single-cell steady-state "model." In any case, the models must be Validated in some sense, at some level of accuracy, no matter how crude, in order to be quantitatively useful.

10.13 DYNAMIC STALL WIND TUNNEL DATA: WHO DOES THE TWEAKING?

In Salari et al (1994), we presented simulations of 2-D dynamic stall using the Wilcox k-ω turbulence model and comparisons with experiments. The experimental data were unpublished. Agreement with the experimental data initially was poor. It became clear from inspection of the data itself (without regard to simulation results) that a problem existed, namely, there was significant flow angularity in the tunnel. (This was evident from the behavior of the static data on the symmetric airfoil.) We (i.e., the modelers) estimated a flow angularity correction from the static tunnel data, not using our simulations, and applied the corrections to the input parameters (angle of attack) of our simulation. The result was greatly improved agreement with experiment.

This kind of data reduction activity creates a gray area - it "muddies the water" of Validation. If such corrections are done by the code user (or worse, by the code developer in a Validation exercise) they could be subject to snickers and knowing accusations of "tweaking the data to fit the computations." But if the experimenter had done the same calculation and called it "data reduction" instead of tweaking, no one would criticize it. The folklore observation is close to literally true: *no one* believes a CFD calculation except the person who performed it, and *everyone* believes the experimental data and reduction except the person who performed it. Cooperation and team work between computer modelers and experimentalists is often and properly called for, yet it does open the door to criticism.

[114] WIPP PA Dept. (1992), Helton et al (1995, 1996).

10.14 CONSORTIUM EFFORT AT CFD CODE CERTIFICATION

Melnik et al (1995) described a collaborative effort in CFD code assessment that covered Verification, Validation, and Certification. The work assessed the usefulness of CFD codes in a production design environment. The project was conducted by a U. S. industry/university team working within a consortium called MADIC (the Multidisciplinary Analysis and Design Industrial Consortium) with funding provided by NASA.

The Code Certification plan was developed by a MADIC/NASA review panel and exercised on five NASA codes in six Certification projects defined by the configuration tested. (Only one configuration was calculated using all five codes, and another using two codes; the remaining three configurations were each calculated with a single code.) The panel developed the overall organizational structure and plan for the Certification process, and facilitated formation of the Certification teams. Although the term QA (Quality Assurance) was not used in the paper, this Certification involved some standardization of metrics, evaluation forms, and log forms, separately published in the open literature (Melnik et al, 1994) and was essentially a QA exercise.

The focus was not on fundamental flow quantities (like velocities, pressures, etc.) but on design performance data; this distinguishes Certification from Validation as the terminology was used by the authors. (See comments in Chapter 2.) Herein, I use the more general term Validation in some cases where Melnik et al used "Certification." Also, when they use the term "Verification" or even "Verification of Code," they are referring to what has been described (see Chapters 2 and 4) as "Verification of Calculations." (They do not address "Verification of Codes" in the sense used herein in Chapters 2 and 3, i.e. demonstration of the order of discretization convergence to a known exact solution.)

Aircraft design performance data include lift, drag, moment, maximum lift and loads, flap effectiveness, stall prediction, buffet onset, internal flow losses, etc. The authors stated that, despite many *ad hoc* studies in the literature, it was not possible to draw broad conclusions, e.g. the fundamental question of how well maximum lift of a simple transport model can be predicted "remains largely unanswered."

The six classes of flow configurations simulated follow. (Only four will be synopsized herein.)

- Isolated Wing
- Nozzle/Boattail Flows
- Axisymmetric Nozzle
- Wing-Body Combination
- Turbine/Compressor
- Flexible Wing (steady and unsteady aeroelastic problems)

The experimental data used was restricted to wind tunnel data because "there is little flight data available that was suitable for Code Certification." This is an impressive statement on the paucity of reliable and complete flight data, coming as it does from a consortium including NASA and major aerospace firms. The teams assessed experimental data with respect to four "factors/issues": "measurements, test conditions, data accuracy, and wind tunnel installation effects." The panel noted that experimental error estimates should be provided for:

- model and support geometry dimensions
- wind tunnel wall and support system interference
- instrumentation
- free stream and operating conditions
- model distortions due to aerodynamic loads.

Corrections for wind tunnel interference should include Mach number, incidence, forces and moments. They noted that "it is also highly desirable that pressure measurements on all wind tunnel walls and model support systems be provided. These can be used either to develop wind tunnel interference corrections or to provide boundary conditions in CFD computations that include the wind tunnel configuration in the computational model." They also pointed to "the continuing need for high quality experiments that are designed specifically for CFD Code Certification" [Validation].

The panel developed a set of criteria and metrics for the assessments "to add uniformity to the process and because Certification of a Code as a design tool requires specific criteria and metrics." The criteria are in four broad categories: "numerical, physical, operability, and range of applicability." The criteria are listed in Figure 10.14.1; see Melnik et al (1995) for further discussion.

A sampling of interesting results from Melnik et al (1995) is now given for various configurations.

10.14.1 Isolated Wing C

This was the only configuration for which all five NASA codes were applied. The wing is highly three dimensional, with aspect ratio = 2.6, leading edge sweep angle = 45°, tip chord/root chord = 0.3, and twist = 8.17°.

The Wing C configuration was simulated for three flow conditions of Mach Number M and angle of attack α, all at Reynolds Number = 10 million: $M = 0.82$, $\alpha = 5°$; $M = 0.84$, $\alpha = 4.62°$; $M = 0.85$, $\alpha = 5°$. Only an algebraic turbulence model was used because only this was common to all five codes.

Grid sensitivity [Verification of the Calculation] was investigated using two single-block C-O topology grids, one with $49 \times 241 \times 49$ (578,641) nodes and the other with $65 \times 321 \times 65$ (1,356,225) nodes. "Solutions on both grids agreed well with each other" so the coarse grid was selected for code-to-code comparisons. Note this is an example of the methodology described in Chapter 5, common especially in such cost-conscious design environments, of Verifying a *coarse* grid calculation by comparison with a fine grid calculation for a single nearby problem. In this case, "nearby" signifies not only small changes in flow conditions M and α but also a "nearby" *code*. The coarse grid is then used for the sequence of simulations including parametric variations and, in this study, five different codes.

Although the stated "Certification" goal was design parameters, comparisons were made for surface pressure coefficients (Figure 11 of Melnik et al, 1995). The assessment was that "all [five] solutions agree quite well at the inboard station. The cause of discrepancy between ADPAC [code] solutions and others at the outboard stations needs to be further investigated."

Comparisons of the five code predictions and the experimental values for aerodynamics coefficients for the case $M = 0.85$, $\alpha = 5°$ are given in Table 10.14.1.1. [The normal force coefficient C_N is close to the lift coefficient C_L at this small α, so the experimental agreement for C_N is indicative of the agreement for C_L and is fairly good, 3.2% discrepancy in the worst case. The discrepancy in drag coefficient C_D might be expected to be worse, but no experimental values were given. However, see the M100 wing-body combination results below. The discrepancy in moment coefficient C_M is large, 75% in the best case, 88% in the worst.] See Melnik et al (1995) for comparison of code economics; total computer times ranged from 3440 to 22000 seconds, and memory required ranged from 17.3 to 39 mega-words.

10.14.2 Nozzle/Boattail Flows

This geometry consisted of a 3-D aft end with a 2-D converging/diverging nozzle. A seven zone grid with a total of 989,595 nodes was used. Verification involved calculations with a grid with 75% grid density and another with 125%. Two codes were used, with three turbulence models: an algebraic or "zero-equation" model (Baldwin-Lomax), and the Baldwin-Barth and Spalart-Allmaras one-equation models.

Both codes confirm the now well-documented inadequacy of algebraic turbulent models in separation, wakes, shear layers, etc., flows for which they were not designed and where they require *ad hoc* adaptations. A representative comparison of surface pressures is given in Figure 10.14.2.1.

10.14.3 Wing-Body Combination

Melnik et al (1995) provided an excellent lesson on the pitfalls of inadequate initial Verification of Calculations, misleading Tuning of experimental corrections, and mixing of Verification and Validation, in their case study of the Wing-Body Combination.

The configuration is a transport type wing (designated M100) mounted on a simple circular cross section fuselage. The mean-chord Reynolds Number was 3.2 million, and boundary layer transition was forced at 7% chord in the experiments. The thin-layer Navier-Stokes code TLNS3D was used with two turbulence models: the one equation Spalart-Allmaras (S-A) model and the blended k-ω/k-ε model of Menter (M–T). Original simulations were done assuming full turbulent flow everywhere; later calculations enforced transition at the experimental location. The code had options for strictly numerical explicit artificial dissipation using either a scalar form of dissipation (SD) or a full matrix form (MD). Although "extensive grid studies were conducted in this project" details were given on two grids, $321 \times 57 \times 49$ (896,553) nodes and one doubled in the spanwise k-direction to $321 \times 57 \times 97$ (1,774,809).

Important to the tale is the fact that an angle of attack correction (for tunnel flow angularity and interference) of $\Delta\alpha° = -0.2636\ C_L$ was supplied with the experimental data. (Note the point made in Chapters 2 and 9, that experiments as well as simulations require computations and therefore include numerical errors.) "The one problem noted in the initial phase of the work was attributed to an aeroelastic effect; the first computations made predicted a 15% higher lift than the experiment. In particular, the section lift at the wing tip section was significantly higher than the experiment," suggesting that the culprit was aeroelasticity, i.e. the wing model was twisting significantly under the aerodynamic load. The experimental report stated "measurements on a similar model indicated maximum aeroelastic wing tip deflection and twist of about +25 mm and 0.25° nose down." So the modelers included the correction for twist in their computations, and this "significantly improved the lift comparisons."

After this twist correction, the drag polar (the plot of C_L vs. C_D, parameterized by α, Figure 10.14.3.1) remained shifted about 69 counts, i.e., the drag coefficient C_D was 0.0069 higher than the experimental data. Originally, this was attributed to inaccuracy in the computations, but the issue was revisited in Melnik et al (1995).

The next effort to reconcile the simulations with the experiments involved Tuning of both numerical parameters and physical models [i.e. an ill-advised mix of Verification and Validation, in the present terminology]. The base line simulation on the $321 \times 57 \times 49$ grid had used scalar dissipation (SD), fully turbulent flow, and the Spalart-Allmaras (S–A) turbulence model, and had produced drag higher than experimental data by 69 counts. A 10 count reduction in this discrepancy was achieved by including another physical feature in the simulation, eliminating fully turbulent flow by including boundary-layer transition at the experimentally fixed location. Another 30 count reduction was achieved by changing the turbulence model from S-A to M-T. After these *conceptual model* changes, pure *numerical* changes were used, doubling the spanwise grid to $321 \times 57 \times 97$ and switching from scalar artificial dissipation (SD) to a matrix form (MD). These numerical changes achieved another 20 count reduction in the drag at zero lift C_{D0} (along the abscissa of Figure 10.14.3.1), high by only 12 counts instead of the earlier value of 69.

This set of simulation parameters was taken as the new base-line, and attention shifted from the drag polar (Figure 10.14.3.1) to generating a new lift curve of CL vs α (Figure 10.14.3.2). Here, the pitfalls of

Numerical	Physical	Operability	Range of Applicability
• Accuracy – Global, local parameters – Spatial accuracy established by grid refinement • Computer requirements – RAM, CPU, disk – Portability • Grid dependent sensitivity • Convergence characteristics • Grid types – Single block – Multiple block – Overset – Unstructured – Adaptive • Portability – Mainframe, mpp, workstations – Distributed Workstations, Clusters	• Accuracy – Performance data – Requirements for design data • Model adequacy – Transition model – Turbulence model	• Reliability – Sensitivity to numerical methods – Tuning parameters & grid robustness • Credibility – Established limitations – Known error estimates • Responsiveness – Wall times – Design environment • Geometry/grid/ solver interfaces • Ease-of-use – Grid automation – Set-up time – User friendly features – Pre- & post-processor integration – User skill requirements • Maintainability – Modularity – Support – Documentation	• Flow regime – Incompressible – Compressible – steady/unsteady – transonic, supersonic, hypersonic – Perfect gas/ real gas • Geometric flexibility • Grid flexibility • B.C. flexibility

Figure 10.14.1. Criteria /Metrics for CFD Code Assessment for Production Design Environments. (From Figure 7, Melnik et al, 1995.)

	C_L	C_D	C_N	C_M
Experiment	–	–	0.5400	–0.0393
Codes:				
CFL3D	0.5256	0.04097	0.5272	–0.0726
TLNS3D	0.5254	0.04290	0.5271	–0.0736
ENSAERO	0.5212	0.0423	0.5230	–0.0739
ADPAC	0.5217	0.03453	0.5227	–0.0686
OVERFLOW	0.5267	0.04422	0.5285	–0.0768

Table 10.14.1.1. Comparisons of five code predictions and the experimental values for aerodynamics coefficients of a highly three dimensional isolated wing for the case $M = 0.85$, $\alpha = 5°$. (From Table 1 of Melnik et al, 1995.)

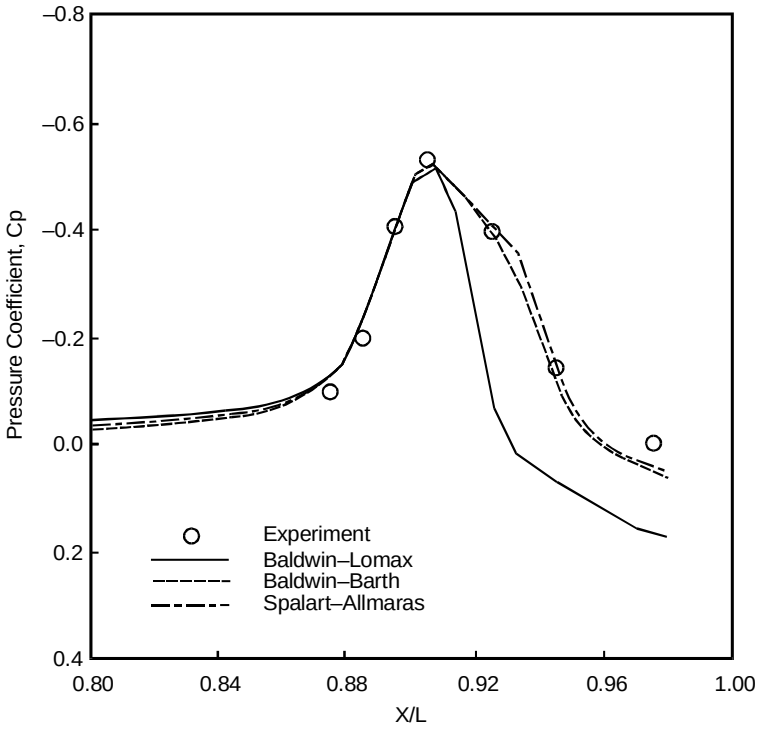


Figure 10.14.2.1. Comparison of Surface Pressures at Mid Sidewall for an Aft-End/2-D Converging-Diverging Nozzle Flow at $M = 0.938$. (From Figure 13b, Melnik et al, 1995.)

Tuning became evident. The old baseline set of simulation parameters had predicted C_L vs. α fairly well, but had produced the 69 count discrepancy in the drag polar. The new baseline set improved the zero-lift drag discrepancy to 12 counts, but now (Figure 10.14.3.2) produced C_L vs. α curves nearly 20% high!

The resolution of the paradox begins with the recognition that the agreement in C_L vs. α of the original baseline was somewhat illusory. The aeroelastic effect (0.25° twist at the wing tip) had been "substantiated by matching the experimental lift, so that it is not surprising that the original base-line computation (solid circles) matched the experimental lift curve." The sleuthing then continued (see Melnik et al, 1995 for details) to conclude that the source of the problem is the under-estimation of aeroelastic correction to angle of attack provided in the experimental data. This correction had been based on under-resolved simulations, i.e. on *inadequate Verification of the calculation inherent in the experiment.*

The team forcefully concluded that Verification of the Calculation should precede Validation and Certification. Especially, this study provided an example of the error and confusion propagated by faulty experimental data reduction calculations, including the pernicious practice of judging grid refinement adequacy simply by agreement with experimental data. "This experience clearly indicates the importance of the three steps (i.e. Verification [of the Calculations], Validation and Certification) in the assessment of a code as outlined in this paper. Before Certifying a code for design, its numerical accuracy must be established (i.e. Verify that the grid used is adequate) and the turbulence/transition model must be Validated. Only then can the Certification [Validation] process begin."

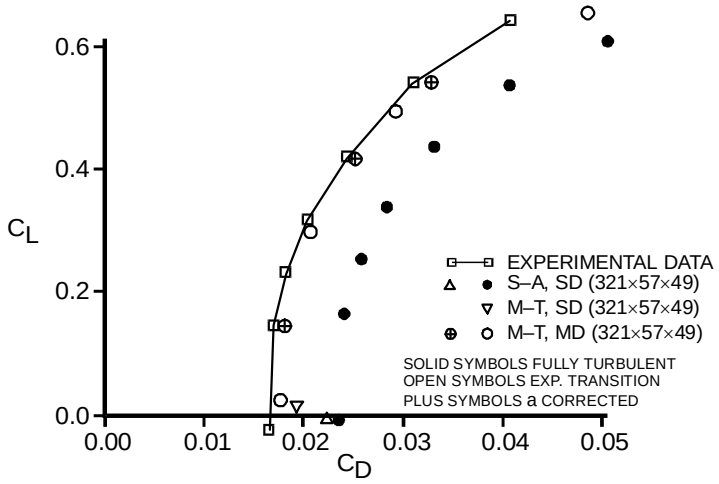


Figure 10.14.3.1. Drag polar for M100 Wing-body Combination; Experimental Data and Simulations Using Code TLNS3D. S-A = Spalart-Allmaras model, M-T = Menter blended *k*-ω/*k*-ε model. SD = scalar artificial numerical dissipation, MD = matrix dissipation. (From Figure 19 of Melnik et al, 1995).

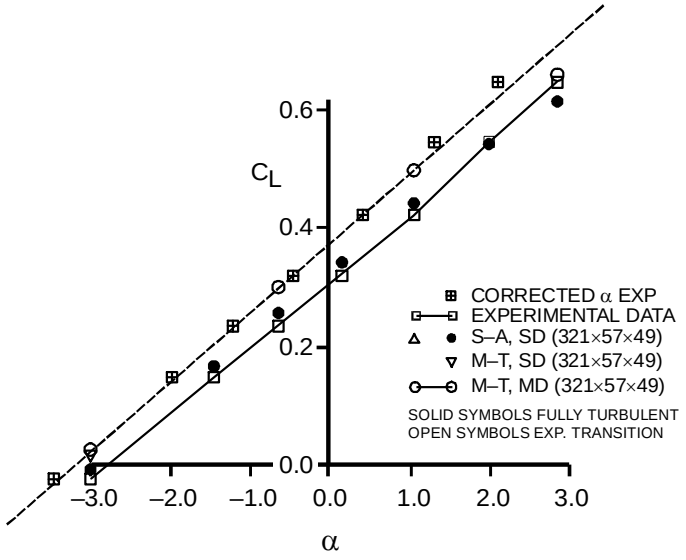


Figure 10.14.3.2. Lift Curve for M100 Wing-body Combination; Experimental Data and Simulations Using Code TLNS3D. S-A = Spalart-Allmaras turbulence model, M-T = Menter blended *k*-ω/*k*-ε model. SD = scalar artificial numerical dissipation, MD = matrix dissipation. "Corrected α Exp." refers to the new base-line set of simulation parameters, which had produced improved agreement with the experimental drag polar. (From Figure 20 of Melnik et al, 1995).

10.14.4 Turbine/Compressor

The study in Melnik et al (1995) of Turbine/Compressor flows involved eight different configurations. The grid convergence study involved 2-D slices with grid density ranging from 193 × 33 to 385 × 81. Even so, the results (Figure 10.14.4.1) showed "the strong dependence of the total pressure loss coefficient (mixed-out) on the grid spacing at an exit Mach number of 0.90."

The "fine mesh prediction of 94.1% peak efficiency agrees fairly well with the measured value of 93.4% and occurs at essentially the same mass flow relative to the choking flow (9.990 predicted vs. 0.989 measured)."

The team's assessment of the single code used noted that "ADPAC is relatively easy to apply by a first-time user," and gave this curious, virtually universal statement that "Moderate CFD experience is suggested to achieve optimal results." [I should hope so.] Less obviously, they noted the need for "serious debugging of less used options" [a Verification of Code issue, in present terminology], for better than an algebraic turbulence model [a Validation issue, in present terminology], and for "more consistent version control" [a Certification or Code QA issue, in present terminology].

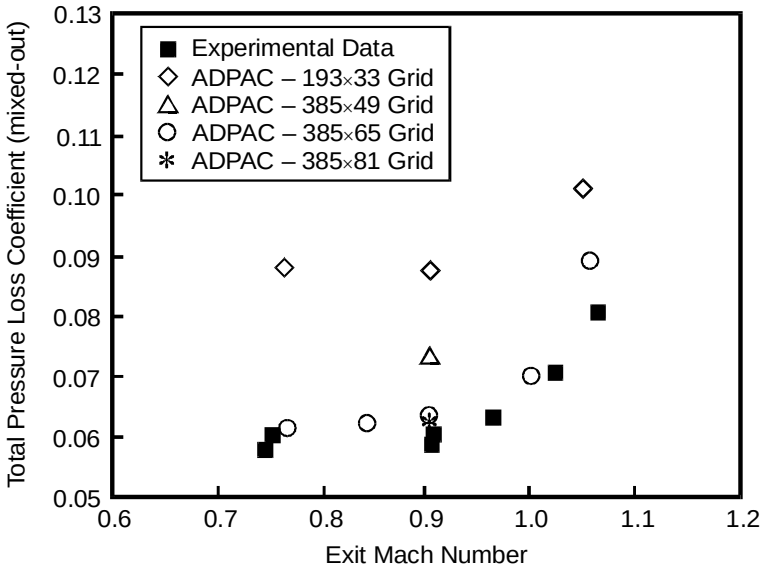


Figure 10.14.4.1. Loss Coefficient for a Turbine/Compressor Flow as a Function of Exit Mach Number, Showing the Effect of Grid Density. (From Figure 23 of Melnik et al, 1995).

.

10.14.5 Selected Observations of Melnik et al

A brief selection of observations from Melnik et al (1995) follows. Some are repetitious here, but deserve to be.

- "Certification is an expensive and time consuming process that is currently carried-out in piece-meal or ad-hoc fashion in the [aerospace] industry. Our experience in the project has demonstrated, at least to us, that a consortium or other type of teaming approach is a very effective way to accomplish code Certification. It is cost effective because it reduces duplication of effort, it accelerates technology transfer to the designers who are the ultimate users and it promotes code developers to do the right thing through the generation of consistent and broad based industry feedback."
- "Code assessment (at the Verification [of Calculations] step) requires computations on grids that are fine enough to reduce numerical errors below those from other sources. Our studies in the project clearly established that this is a necessary first step in the assessment process. Failure in the Verification step will, at best lead to inconclusive results, and at worst to misleading or erroneous assessments."..."In this project, because we were able to use grids that were sufficiently fine, we were able to observe first-hand the pitfalls of working with inadequately Verified codes [Calculations]." From our experience on this project it is clear that proper Verification [of Calculations] is an absolutely necessary first step in the code Certification process."
- The complete Certification process cannot be done in general or once-and-for-all but must be carried through for each general class of configurations...and/or physical phenomena...Each of these situations involves differing grid and physical modeling requirements that must be addressed in the [Verification of Calculations, Validation, and] Certification process."..."Certification has to be thought of as an ongoing activity..."

- "Data archiving is an important aspect of the code Certification process..."..."The data should be archived in a standardized form..."..."Lack of a process of archiving Certification data is hindering the Certification process and the transfer of CFD technology to the industrial design community."
- All the studies pointed to the inadequacy of simple algebraic turbulence models.
- "The prevailing view seems to be that CFD cannot be trusted for absolute values of drag. Our results, at least for the simple M100 wing/body combination, indicate that the TLNS3D [code] can predict drag to within 5% on a relatively crude 1.7 million point grid."..."This conclusion was reached only after a careful grid study in the Verification stage on the assessment."
- "All teams pointed to the critical need for additional experimental data for CFD code Certification."
- "Larger computing resources will be needed to treat problems of greater complexity..."
- "Support is needed for the acquisition of suitable experimental data as well as for the Certification process itself. Unfortunately, this NASA supported project is the exception and not the rule."

10.15 SIMULATION TEAM RESPONSIBILITIES IN VALIDATION/CERTIFICATION

Rizzi and Vos (1996) noted that the engineering activity of CFD simulations involves the interaction of three distinct teams.

1. Code Developers–Software engineers, mathematicians, and physical model developers.
2. Physical model experts who carry out Calibration and Validation.
3. Users, who are [generally] not experts in either the code nor modeling, but are application specialists.

Rizzi and Vos stated that, of the four steps (Verification, Validation, Calibration, Certification) required to build credibility, Verification [of the Code] and Certification are the responsibility of Team 1, the Code Developers. (Recall that, as described in Chapter 2, Rizzi and Vos do not include Validation in their concept of Certification.)

"The work of the second and third teams are interrelated because Calibration and Validation are specific to a particular design application and user community. It is the task of the second team to Calibrate a code, and through Validation determine its range of applicability, which then has to be transferred to the end users. Validation and Calibration are related. In Calibration you improve the results of your Validation, at the expense of a general Validation because you modified or specialized the physical model to make it work for a certain type of generic flow. In Validation you identify all the generic flow types that make up the complex flow and then choose the appropriate model to make the predictions. The reasoning behind this approach is inductive, and therefore must be Validated. The key activity, which is also the most demanding and expensive, is Validation."

10.16 SHIFTING RESPONSIBILITIES AND GRAY AREAS

Discussion of shifting responsibilities is a practical consideration, even though it might appear confrontational (vs. team-oriented). Consider the common situation (e.g., cited in Barber, 1996) wherein a Validation exercise shows discrepancy between CFD simulation and experiment, and the cause is agreed to be probe interference. That is, the experimental flow is perturbed significantly by the introduction of the experimental measuring apparatus.

Should the CFD modeler volunteer to model the perturbed flow about the probe?

If he does, the responsibility for the discrepancy shifts from the shoulders of the experimentalist (whose job it is to minimize probe interference effects, calibrate them, and if possible account for them in the data reduction process) to the shoulders of the modeler. Unfortunately, the probe flow introduces another scale of resolution, and possibly another dimension into the modeling problem. (For example, the undisturbed flow may be axisymmetric 2-D, but a boundary layer probe makes it fully 3-D.)

This is a QA and engineering management issue. The modeler's decision will be influenced by the professional environment, levels of trust (will "responsibility" become "blame"?) and personal values and goals.

More difficult assignment of responsibility will occur in gray areas. For example, in a comparison with wind tunnel data, if the experimenter claims uniform free-stream flow conditions, then any deviation from this condition is the responsibility of the experimenter. But it would not be unreasonable to expect a CFD code to accommodate some non-uniformity in free-stream conditions. (Unlike the probe example, this would not introduce another scale of flow variation, although it might destroy axisymmetry.) If the CFD code cannot include variable free-stream conditions (due to limitations on input data simplicity, memory requirements, or the fundamental formulation of the equations) then the responsibility (blame?) shifts towards the modeler.

10.17 WUA BENCHMARKS IN 1994 AND 1996

The World User Association in CFD sponsored conferences in 1994 and 1996 (Muller, 1994; Muller and Loffler, 1996) that included benchmark calculations by commercial code vendors. The problems were defined by a WUA committee, with suggestions solicited from true industrial users. In spite of much work and expertise, difficulties were experienced with inadequate problem definition. These included undefined geometry and undefined inflow boundary conditions needed for turbulence models. By commitment and attention to feedback from the first conference (de Vahl Davis, 1994), the committee succeeded in correcting many difficulties for the next meeting.

The test cases provided good range of industrial fluid dynamics, including a generic automobile model, flow in a chemical smoke flue, impinging turbulent jet flow with heat transfer, and a stirred reactor tank. The biggest difficulty with the WUA comparisons, common to all such exercises, was that any vagueness in problem specification (e.g., inflow turbulence quantities), no matter how realistically representative of the industrial modeling environment, means that the exercise becomes a comparison of the skills of analysts, when what is desired is a comparison of codes.

Note again the false criticism of turbulence models and CFD, previously discussed. No reasonable turbulence model requires more information (boundary conditions and initial conditions) than the physics. In fact, all turbulence models are simplifications of the true physics, i.e. the full Navier-Stokes equations, and therefore may be expected to require less information than required to rigorously define the problem. Thus, if a CFD turbulence model shows that the results of interest (e.g., distance to reattachment for a backstep separated flow problem) are sensitive to some parameters (such as the upstream turbulence kinetic energy distribution) and if this model has been Validated on this point, then the fact that some experiments have not measured this quantity indicates that the experimenters have been running a test with uncontrolled variables. (No wonder agreement between facilities is poor!) Nevertheless, these undefined variables from incomplete experiments impede Validation, making the agreement with experiment or lack thereof dependent more on engineering guesswork and intuition than on reliable data.

10.18 CFD TRIATHLONS

The CFD Triathlons organized by Freitas (1993c,1995b) and sponsored by the ASME were designed to apprise potential customers and users of the capabilities of CFD commercial codes. Rather than simulate complex industrial fluid dynamics problems with the difficulties noted above (Section 10.17), the vendors were invited to submit their simulations for fairly simple, unambiguously defined, somewhat academic problems. Freitas did not evaluate the results but just let the papers stand for themselves. Even though the problems were simple, some of the discrepancy can be attributed to the skill of the modelers as well as codes.

These observations come to mind.

1. Verification of Calculations were generally inadequate, with most of the papers reporting results from only one grid on all or some of the problems, or minimal non-quantified reference to other studies.
2. Often, CFD commercial codes show poor accuracy, usually due to the sacrifice of accuracy to robustness.
3. A general point, aside from the evaluation of commercial codes or research codes, etc. is that user training is required. (See also Rizzi and Vos, 1996, and Section 10.23 below.)
4. There was a wide range of computer times required to solve the same problem with roughly similar grid densities.
5. Some results were clearly erroneous, and others differed from one paper to another.
6. Although the flows were laminar, Validation (i.e., determination of whether or not the "right equations" were used) was still an issue, because of the assumption of 2-D flow in two of the problems.

10.19 CANADIAN CFD SOCIETY TEST CASE

A. Pollard organized a comparison exercise of commercial and research CFD codes for the 1995 meeting of the CFD Society of Canada (Thibault and Bergeron, 1995). The problem was one of turbulent heat transfer in the vicinity of stagnation point flow. It was intended to be a true prediction exercise (rather than "postdiction"), i.e. the experimental data was not made available to the participants beforehand. However, some of the participants recognized the problem specification and identified the journal paper with the experimental data. Not surprisingly, their results were best. (See Section 9.2.4.) As is often the case, some of the turbulence information required was missing from the experiment, so the problem specification had to be supplemented by the experience and intuition of the modelers.

As is always the case with non-trivial turbulent flow cases, the results were generally unsatisfactory. Agreement with experiment was generally poor, and agreement between different modelers using the same turbulence model was also poor. The exercise was worthwhile, again showing the unsatisfactory state of experimental data, especially for turbulence measurements, and the lack of Verification of Calculations.

10.20 WORKSHOPS

10.20.1 Older Workshops

Workshops have been a part of CFD for many years. A workshop for Verification and/or Validation may be described as "a collective learning process where a group of researchers actively working on a sharply defined topic meet to discuss in detail their problems and experiences and make direct comparisons and critical evaluation of their computational and experimental data." (Rizzi and Vos. 1996.) The 1980/81 Stanford Conference widely known as the "Stanford Turbulence Olympics" (Kline et al, 1981a,b) is a

premier example. Rizzi and Vos (1996) reviewed several such activities, and noted that a strong case exists for collaborative Validation, involving a number of different codes and at least two rounds of analysis. (See also Section 10.23 below.) Another good example is the transonic airfoil workshop of 1987 (Holst, 1987).

Marvin (1995) noted that an important aspect of Validation strategy is the "cataloging, documentation, and assessment of experimental databases. Such information can be used in workshops to assess code development and Validation, by CFD design groups, who must Validate codes used in the design process, and by planners of future experimental programs, who should be aware of existing data." He cited AGARD (1979, 1985), Kline et al (1981a,b), and Settles and Dodson (1991) as precedents of Validation Workshops. Porter (1996) also contains references to Primary Workshops, Databases, etc. The JANAFF Airbreathing Propulsion Subcommittee, Airframe Integration Panel has sponsored a series of workshops on CFD Code Validation/Verification and "has found the subject to be one of major interest and impact on the aerospace community as a whole." Proceedings are published yearly by the Chemical Propulsion Information Agency (CPIA) under JANAFF auspices, e.g. CPIA Publication 551, etc. See Porter (1996) for discussion.

10.20.2 § Lisbon III V&V Workshop

The series of three Lisbon Workshops (Eça and Hoekstra, 2004, 2006b, 2008) were exceptional experiences. The first focused on Calculation Verification for a realistic problems of RANS turbulence flow over a backstep. The Workshop organizers provided non-orthogonal boundary fitted grids, some of which were deliberately problematical in order to stress the uncertainty estimators. The second workshop also covered Calculation Verification but expanded to include Code Verification by MMS using manufactured solutions for several RANS models provided by the Workshop organizers and committee; these manufactured solutions in themselves constitute a valuable contribution to the V&V literature (Eça et al, 2007b,c). The third Workshop covered all three V&V aspects, using the often cited experimental data on the turbulent backstep flow by Driver and Seegmiller (1985) and the brief version of the Validation Uncertainty methodology of ASME ANSI Standard V&V20 as described in the following Chapter 11.

Besides these technical descriptions, all three Workshops were judged remarkably successful and a paradigm of Workshops in general by virtually all participants. The Workshop was narrowly focused on V&V from the modelers' viewpoint (rather than much discussion of experimental work). All the participants shared the same V&V ethic of striving for high quality work and conscientious effort to estimate uncertainties. Most notably, the organizers arranged for the ratio of discussion time to formal presentation time to ~ 1. That is, these were truly Workshops, not just technical paper presentations. Even the Proceedings and the AIAA summary papers (Eça et al, 200, 2007a, 2009) exhibit the deep level of discussion and sharing of experience that occurred during the two full days of each Workshop. Anyone planning to organize a V&V Workshop would do well to consider this approach.

10.20.3 § INL Nuclear Systems V&V Workshop

The "Verification and Validation for Nuclear Systems Analysis Workshop" (Imel and Mousseau, 2010) convened in Idaho Falls (USA) for five days in July 2008, sponsored by the U.S. Idaho National Laboratories and three other academic and research institutions. While some of the presentations were focused on V&V as intended, others were more general experimental papers with no presentation of experimental uncertainty. Only one experimental paper included thorough experimental uncertainty as described in an earlier publication (McIlroy et al, 2006). Not all participants shared the V&V ethic, e.g. declaiming responsibility for verifying commercial codes for their own high-consequence applications. Most participants agreed that the sessions devoted to discussion rather than formal presentation were most

valuable, and several experimentalists stated that they benefited from the Workshop atmosphere with an enhanced appreciation for the modelers' needs for accuracy and especially experimental uncertainty estimates.

10.20.4 § AIAA Drag Prediction Workshops

The AIAA Drag Prediction Workshops (http://aaac.larc.nasa.gov/tsab/cfdlarc/aiaa-dpw/) are an exemplary series of ongoing Workshops with a narrow focus. As an example of continually updated assessment of computational state-of-the-art, they are perhaps unmatched. See Section 10.24 for description and references.

10.21 AGARD 1988 VALIDATION OF COMPUTATIONAL FLUID DYNAMICS

In May 1988, AGARD (the NATO sponsored Advisory Group for Aeronautical Research and Development) hosted a conference/workshop entitled *Validation of Computational Fluid Dynamics* in Lisbon, Portugal. The proceedings (AGARD, 1988) covered 3 invited papers, 34 contributed papers, 16 poster papers, and a Round Table Discussion. The lengthy proceedings provided an overview of the status of aeronautical Validation, including terminology, as of 1988. Session topics included: CFD Validation Concepts; Airfoils; Vortex Flows; Wings/Wing Body; External High Speed Flows; Turbomachinery; and Intakes and Ducts.

The proceedings are worth perusal by anyone active in CFD aerodynamics, yet some deficiencies are noteworthy. The semantic problems go beyond the usual ones, approaching "epistemological chaos." Not all the papers even treat Validation - some just present CFD methods, some present only experimental data that might be used for Validation, but were not designed specifically for CFD Validation. Some are not well done, and few address general concepts. Some common themes emerged from the specifics of grunt work, including these.

- There is real value in using some definitions (though not overly precise) to distinguish concepts.
- Good experiments are hard to find.
- It is best to have experiments specifically designed for CFD Validation.
- As a corollary, cooperation is required (and was often evident) between computationalists and experimentalists.
- There is a wide range of opinion as to what one may honestly describe as "good agreement with experiments"
- Navier-Stokes codes are better than Euler codes for predicting separation. [!]
- Boundary layer transition was never modeled, neither in the CFD computations nor in the wind tunnel "modeling."

The candidate definitions presented have already been discussed in Chapter 2; in AGARD (1988), semantic clarity was lacking. Some of the participants certainly grasped the essential distinction between mathematics and science (e.g. see the remarks by K. Gersten and by L. Roberts in the Round Table Discussion) although no one used the precise terms Verification and Validation. (The conference was held in 1988, and agreed-upon distinctions were still emerging.) On the other extreme, even the standard terms consistency, stability, and convergence were misused. "Consistency" was confused with behavioral errors (Roache, 1998b) so that the desirable, but not at all necessary, property of an algorithm identically preserving uniform free-stream flow at all discretizations became confused with recovering of derivatives from differences. Likewise, "stability" was confused with parameter sensitivity. Definitions of "stages of

CFD code development" may be expected to be somewhat vague managerial terms, but the criteria of "value exceeds expectations" is irrelevant, having nothing to do with the code *per se* but only with previous managerial prognostications. The claim that [iterative] convergence acceleration schemes constitute a limiting factor to accuracy is not true; to the contrary, faster iteration convergence will make it easier to achieve near steady-state solutions and therefore may be expected to slightly improve accuracy. Empty statements are common, such as "today's computers do not permit total elimination of this error source" related to grid density - as if tomorrow's computers will? There was significant lack of consensus on the difficulty of gird generation, with one participant claiming that "It is now clear that the construction of grids around complex configurations is not a major problem anymore." This is at a conference in 1988! Generally, there were lots of lists with incomplete coverage. Many of the papers noted the now familiar need for convincing grid convergence in order to evaluate the experimental comparisons.

In the AGARD Proceedings, Bradley (1988) described four categories of experiments related to CFD development.

Experiments designed to
1. *understand flow physics*
2. develop *physical models* for CFD codes
3. *calibrate* CFD codes
4. *validate* CFD codes.

Each of these experiments should be designed differently with regards to variables measured, measurement resolution, accuracy required, etc.

On the quality of Validation experiments, the respected experimentalist S. Bogdonoff gave (in the Round Table Discussion) an attention-getting evaluation. "I am appalled at the use of experiments that were made over the past 40 years to do computational 'Validation'. Those experiments were carried out to do specific things for many reasons, but they were surely not designed for computational Validation. I know of very few experiments which were carried out 30 years ago or 20 years ago, or even 10 years ago that have the quality, the basic resolution, and the parameter variations that are required for computational Validation. I would suggest that developing a code and then looking for some experiments to Validate it is unreasonable. The experiments that you need to Validate a code, I believe, have to be designed specifically for that task. You should have computations to tell you what kind of details you need from the experiment, what range of parameters, what kind of instrumentation you need to really test if the code is worth anything or not." Bogdonoff also stated that, as an experimentalist, "the major contribution that I have gotten from computation...is that what I thought was important in the experiment was not the physics that turned out to be critical." This is strong testimony on the effectiveness of simulation, recalling the statement of Hamming (1962): "The purpose of computing is insight, not numbers."

In regard to what constitutes "good agreement," we am not talking about the differing needs for differing end uses, but differing standards of quality for the same end use. The laxness of some computors was simply shocking.

It should come as no surprise that Navier-Stokes codes are better than Euler codes for predicting separation. On the contrary, one would be forgiven for expressing surprise that use of Euler codes was even attempted, since inviscid flows do not separate. What happens is that discretization introduces viscous-like behavior. Often in Euler codes it is an explicit artificial viscosity that is very close to physical viscosity; sometimes it is implicit effects that are only analogous, not strictly equivalent, to viscosity; or sometimes it is not really viscous-like but just some kind of numerical error that allows separation to occur. If the body has a geometrical feature that practically fixes the location of separation (i.e., a sharp leading edge delta wing) then virtually *any* viscosity, physical or otherwise, gives a reasonable answer. Unfortunately, people

then push their luck, and try some rounding of the leading edge. If that works, why not a normal airfoil? The reason why not is that the artificial viscosity, and therefore the separation location and the entire solution, depends on the grid spacing. As the grid is refined, the solution does not converge to a physical answer. (This discourages grid refinement studies.) Nevertheless, attempts persist, and ill-conceived Euler solutions of separation are as persistent as 1st-order upstream differencing, or crabgrass. Of course, if Euler codes are built with zonal boundary layer submodels, grid convergence can be meaningful.

Boundary layer transition was never modeled because it is very difficult, both computationally and experimentally. Instead, transition was always induced artificially in the wind tunnel "modeling" with a trip wire or other device. This fixed transition location is then input to the CFD model, often with some blending required (over ~5 cells) to maintain stability. This is a wise Validation approach, because to use "natural" transition would hopelessly confuse the Validation procedure. Transition is known to be devilishly sensitive to wind tunnel idiosyncrasies, even in high-quality tunnels specifically designed for transition studies. In tunnels designed for gross measurement of aerodynamic coefficients, it is a very loosely controlled phenomenon, unrepeatable from one tunnel to another, or even from one run to another in the same tunnel. Also, the CFD codes (certainly those in 1988) did not have transition models in them. This does not mean that the modelers were ignoring the importance of transition, just that they did not know how to handle it. The Validation of a CFD code is then performed with the understanding that it cannot be applied to a new free-flight case without addressing the transition issue. Fortunately, tremendous progress has been made in the last few years in predicting transition (e.g., see Haynes et al, 1996; Reed and Saric, 1989; Reed et al, 1996). It may soon be possible to embed a transition module in full Navier-Stokes codes.

For details on what is involved in the careful design of experiments for CFD Validation, the AGARD (1988) papers by Elsenaar et al (1988), Firmin and McDonald (1988), and Kordulla et al (1988) are recommended.

10.22 CASE STUDY FOR CFD CODE VALIDATION METHODOLOGY

Aeschliman and Oberkampf (1997) presented details of their case study for CFD code Validation methodology. The project was a long term, coupled CFD and experimental effort initiated at Sandia National Laboratories–Albuquerque in 1990, and referred to as JCEAP for Joint Computational/Experimental Aerodynamics Program. The study involved Validation of CFD codes using experiments in long duration hypersonic (M = 7.84) wind tunnels, which experiments were specifically designed for the Validation. The methodology will be of interest far beyond hypersonic wind tunnels, however. Such a project represents a major commitment to quality work on the part of all personnel and management. As noted above in Section 10.6, synergism arises when experiments are designed specifically for CFD Validation, and much was learned about both computational and experimental techniques.

The geometry used was a spherically blunted cone with a sliced aft region and flaps at the rear of the slice. With varying flap angles, this relatively simple yet modular geometry generated a wide range of flow complexity, from simple attached flow to very complex flow with massive separation and strong shock wave/boundary layer interactions. The flow was laminar everywhere, as determined by flow visualization techniques. Especially pertinent to the evaluation of tunnel flow angularity and nonuniformity errors was that the model was rolled, to take data at 0, 90, 180 and 270 degrees, and its axial location within the test section was also varied. This unusually thorough approach used in conjunction with 96 pressure ports, various flap angles, and repeat runs produced a massive amount of comparatively high-resolution data. The authors performed a thorough experimental Uncertainty Analysis, obtaining an experimentally based statistical estimate of variance components of surface pressure measurements, including the effects of model geometry uncertainty.

Full details of the statistical methods and conclusions will be found in Aeschliman and Oberkampf (1997), and this report is highly recommended. They determined unambiguously that "the dominant contributor to uncertainty in these surface pressure measurements is due to the nonuniformity of the tunnel test section flow field. Although we had previously suspected this was the case, the present statistical analysis quantitatively demonstrates it." Their suggestion is that this result probably applies to other hypersonic facilities, and perhaps to transonic facilities as well.

10.23[115] Δ DYNAMIC DATABASES FOR VALIDATION

Rizzi and Vos (1996) presented early their vision for a process of systematic Code Validation that involves dynamic, interactive databases stored and accessed on the World Wide Web. The particulars came from their experience in the European aerospace community, but the general concepts are applicable to all areas of engineering and science involving the numerical solution of partial differential equations.

As noted earlier, Validation of a Code, if it has meaning at all (as opposed to Validation of a particular Calculation) depends on the concept of a nearby problem, and this concept will fail near parameter "transition" boundaries, which should be identifiable from knowledge of the general field. (See list in Chapter 9; see also the cautionary examples of difficulty cited earlier in Section 10.7.) The first requirement in the vision of Rizzi and Vos (1996) is for the establishment of a detailed taxonomy of flows for the subject application field, establishing for the database users sets of flow cases acceptable for Validation of "generic flow cases" (what are herein described as "nearby" flows). Rizzi and Vos stated: "The taxonomy is also the basis upon which the code user transfers credibility from previously Validated generic flow cases to the simulation at hand."

Other elements of their vision for the Dynamic Database for Validation are these.

- Creation of a detailed flow taxonomy for the applications area.
- Establishment of an electronic database on the World Wide Web.
- The database performs these functions:
- Archives the experimental data from trustworthy experiments;
- Provides easy access and full public scrutiny to the experimental data for Validation exercises;
- Provides easy access and full public scrutiny to the previous Validation computational data;
- Provides data manipulation (plotting) software;
- Allows for ongoing full public discussion of the information.

The essential first requirement for a taxonomy of flows follows what Rizzi and Vos denoted as the "Stanford Paradigm" (Kline et al, 1981a), which they consider (and most would agree, myself included) to be a "landmark event." They noted the following other features of the Stanford Paradigm, which "still stands today as complete in the sense of providing all elements necessary for understanding the state of the science and for carrying our CFD validation."

- Taxonomies that organize
  - flows,
  - methods of modeling,
  - numerics.
- Classification of experimental data and computational data through a review process.
- Comparison of all computational results with experimental data and with other computed results, case by case.

[115] The order of Sections 10.23 and 10.25 have been exchanged from VV1, and Section 10.24 is new..

- Examination, analysis, and discussion of the comparisons.
- Overall synthesis of the findings by distinguished experts.
- Storage of all data in specified machine format.

Rizzi and Vos also stated that the Stanford Conference was "a very professional attempt to give an overview of the state of the art in computing complex turbulent flows in 1981 using the experimental data base established for this purpose in 1980." While I agree whole-heartedly with this evaluation, I note some limitations of this very professional attempt. (I served on the Evaluation Committee of the Conference.) These limitations have a lesson for future projects as well. The project did not really represent the state of the art of computing turbulent flows because not everyone active in this area participated in the conference; a notable absentee was any representative from the Los Alamos group, who had in fact pioneered the development of two-equation turbulence models (Harlow and Nakayama, 1967), what had by the time of the Stanford Conference been re-named as the k-ε model. Of those who did participate, not all computed all the flows for which they were equipped. The most obvious reason is that participation in such an activity requires much effort, time, and money. Other understandable personal reasons are reluctance to subject one's models and work to so much public scrutiny and inevitable comparison, avoidance of confrontational discussion, disagreement and distrust about the evaluators, reluctance to loose one's competitive advantage by disclosure prior to journal publication, etc. The taxonomies of flows and modeling were excellent, but the attempted taxonomy of numerics proved to be inadequate because many methods are hybrids of the various taxonomic categories (e.g., implicit vs. explicit). Many participants felt the scope of the conference was overly ambitious, and that more narrowly focused problems would have been preferable; in fact, this has been the trend in later workshops (see Rizzi and Vos, 1996 for examples). Finally, as noted in Chapter 1 and elsewhere, the lack of Verification by the participants made firm conclusions on the relative merits of the turbulence models virtually impossible (Kline et al, 1981b).

Nevertheless, the Stanford Paradigm, as well as its experimental data base, still serves well, a remarkably long-lived success considering the advancements in computers, communications, and modeling in the intervening years. The following major additions in the Rizzi and Vos vision are enabled by advances in technology since the 1981 Stanford Conference.

- A *dynamic* (rather than static) database that
  - can be updated,
  - can be interrogated by the user with graphical tools,
  - can be searched electronically (via keyword files),
  - can provide some generation of synthesis documents [automatic report preparation],
  - can include reports, pictures, videos,
  - can include expert evaluation and advice on
    - appropriate level of modeling complexity for a user's needs,
    - specific computational codes, and
    - specific computational procedures,
    - thus providing on-line user training and
    - serving as an electronic "Handbook on How to Use CFD."

The updating feature is extremely important. As Rizzi and Vos noted, traditional methods of disseminating experimental data are actually quite inadequate. The artificial length restriction on journal articles means that the data published is often incomplete and that precision is greatly reduced because of small Figures. A dynamic, electronic archive removes these restrictions, and further allows access to "raw data" and details of the data reduction (numerical) techniques used to process the raw data into final data.

This would allow later refinement of the data reduction; for example, someone (the original experimenter, a reviewer, or someone attempting a Validation of a Code) could improve upon the ideal gas assumption made in the original data reduction by incorporating real-gas effects. Generally, the dynamic aspect of the database allows for the iterative nature of the scientific process to proceed more naturally, without the artificial constraints of hard-copy publications that sometimes cast errors in concrete. Describing previous static databases, Rizzi and Vos (1996) note that "In some cases the input description is incomplete and the output format for the results [has] been specified imprecisely."

The ASME *Journal of Fluids Engineering*, under the editorship of D. P. Telionis, has established a dynamic database at the Virginia Polytechnic Institute and State University. Selected experimental or numerical papers are accompanied by data deposited in the Data Bank, accessible at either telnet scholar.lib.vt.edu or http://scholar.lib.vt.edu/. This system also contains a bulletin board where readers can read discussions and contribute their own on specific papers.

A significant difficulty, at least in the aerospace sector and likely in other sectors of industrial fluid dynamics, is that some databases are not accessible from the public domain, e.g., the HAEDB system described in Rizzi and Vos, 1996. They noted: "In addition, the most sensitive data is saved off-line, and can only be accessed after intervention of the HAEDB manager."

The most important first step, the development of a taxonomy of experimental results, will be specific to sub-fields within the fields of electrodynamics, chemistry, biological systems, laser physics, structural dynamics, fluid dynamics, etc., etc. The fluid dynamics category would itself be much too broad for a single database. There is little in common between the fluid dynamics of ocean current modeling, free-surface wave interactions, groundwater flow and transport in porous media, dusty aerodynamic flows, hypersonics, free convection, etc. Within the category of Industrial Sectors, Rizzi and Vos (1996) note that ERCOFTAC (the European Research Community on Flows, Turbulence and Combustion Association) created an Industrial Advisory Committee. One of their tasks is to define the taxonomies for flows in different industrial sectors, such as Aerospace, Power and Energy, Process Engineering, Automotive, Environment,...

Even restricting the dynamic database to the sub-field of aerodynamics would be to broad. Restricting themselves to the aerospace sector and compressible flows that are either fully turbulent or for which the transition location is known, Rizzi and Vos (1996) offered "a glimpse of what would be a preliminary attempt to categorize features," presented here in Table 10.23.1.

Besides this preliminary taxonomy, Rizzi and Vos gave another list of phenomena pertinent to classification of fluid dynamics classes: "turbulence, relaminarization, retransition, catalyticity, chemical kinetics, combustion, shock interactions, shear-layer mixing, separation, secondary flows, etc." Obviously, this preliminary attempt was presented to give the flavor of the task; the result is neither unique nor complete, nor would it ever be - the *dynamic* aspect applies to the taxonomy itself as well as the data. As the authors stated, "The definition of classes is never completely precise and of necessity imperfect because single classes of features never occur isolated in a complex flow, but the categorization is an aid in establishing credibility and should be seen as such. The process is more difficult and less reliable without it."

Of course, one can easily construct a taxonomic tree and sort through it to the area of interest electronically. The point is that creation and maintenance of a dynamic database involves more than just a decision tree. As Rizzi and Vos (1996) pointed out, such a database requires a substantial investment to develop and maintain, including the cost elements shown in Table 10.23.2.

Such efforts require major funding and collaboration from interested groups. Rizzi and Vos (1996) stated that "funding on a governmental (or European) level is needed to build and maintain such a system." They gave examples of some electronic databases for aerodynamics in existence in 1996.

I. Wall-Bounded Flows
  I.1 attached
    I.1.a with pressure gradient
    I.1.b without pressure gradient
    I.1.c with shocks
    I.1.d with or without suction
    I.1.e with or without blowing
  I.2 separated
    I.2.a bubble type
    I.2.b with reattachment
    I.2.c without reattachment
II. Free Shear Layer Flows
  II.1 vortex flow
    II.1.a delta wing leading edge
    II.1.b forebody
  II.2 ...
III. Mixing Layers
IV. Flows with Confluence of Two Shear Layers
  IV.1 high-lift devices
    IV.1.a off-body recirculation zones
  IV.2 ...
V. Flows In Cavities
VI. Wake Flows

Table 10.23.1. Preliminary Taxonomy of Compressible Turbulent Flows of Interest in the Aerospace Sector. (Rizzi and Vos, 1996)

- Personnel costs of an *evaluation committee*
  (to assess the quality of each experiment)
- Initial investment in hardware and software
- Costs to run the Validation cases
- Support for a database *manager*
- Hardware replacement and/or upgrade
- Software maintenance

Table 10.23.2. Cost Elements of a Dynamic Validation Database. (Rizzi and Vos, 1996)

DATABASE	WWW URL (http://)
ERCOFTAC Fluid Dynamics Database	fluindigo.mech.surrey.ac.uk/
NAS Data Set Archive	www.nas.nasa.gov/NAS/DataSets/
MADIC/NASA Code Certification	hpccp-www.larc.nasa.gov/ccpintro.html
ASME Journal of Fluids Engineering	scholar.lib.vt.edu/ejournals/JFE/jfe.html
NPARC Alliance Validation Archive	info.arnold.af.mil/nparc/Validation_information.html

Table 10.23.3. URLs for Five Validation Databases

The annual AIAA "Year in Review" feature of Aerospace America includes updates on databases including fluid dynamics and other aerospace disciplines. Several current are given in Table 10..23.3.

A dynamic database for Code Verification of primarily FEM solid mechanics problems is NAFEMS (2006). The solutions (analytical or accurate numerical) given are for simplified problems. The *ASME Journal of Fluids Engineering* maintains an online database at http://scholar.lib.vt.edu/ejournals/JFE/data/JFE/. The NPARC Alliance database includes basic shapes and nozzle data of interest in fluid dynamics at http://www.grc.nasa.gov/WWW/wind/valid. The ERCOFTAC (European Research Community on Flow, Turbulence and Combustion) provides several databases accessed through http://ercofrac.mech.surrey.ac.uk. A database for turbulent heat transfer is http://www.jsme.or.jp/ted/HTDB/dathet.html.

Rizzi and Vos discussed some successes and some partial failures, both in the database technology and in the science. In the partial failure category, they noted examples of lack of compliance on the part of contributors, including failure to provide keyword files, geometry rescaling and coordinate system irregularities, and incomplete answers. They stated "although some of the problems appear trivial, they prohibited the automatic functioning of the data manipulation tools." Also, they noted the need for "efforts [to] avoid or at least explain major differences in results obtained by different computors with the same turbulence model." (My impression is that, in fluid dynamics, the communications technology is of better quality than the science. Also, my impression is that the uniformity of results from various structures codes is less than stated by Melnik et al, 1995 and by Rizzi and Vos, 1996.) But even with modern communications technology, it will not work to try to create a "database of everything." (At least, not in the immediate future. Later, an all encompassing "database of databases" is feasible in principle. See, for example, the online "Encyclopedia of Life" initiated by E. O. Wilson.)

Further discussions of these ideas for dynamic databases was given by Vos et al (2002). Oberkampf and Trucano (2007, 2008) also considered many of the same aspects, including search engines, standard computer languages and formats, etc. Note that their recommendations for SSB's (Strong Sense Benchmarks) for both Code Verification and Validation often appear excessive because their focus is not on individual or small scale V&V efforts. "The benchmark effort we describe is not feasible as a short-term task." Rather, their SSB's refer to long term dynamic database requirements. As they perceptively noted, given the high quality recommendations for dynamic database SSB's not only for V&V techniques but also for thorough documentation, "these benchmarks could be viewed as carefully documented step-by-step sample problems from which practitioners, new and experienced, could learn a great deal." They also noted that such dynamic databases, if public and widely used, will naturally evolve into international standards.

10.24 § STATISTICALLY ASSESSING STATE OF THE ART

Hemsch (2002a) developed a simple graphical framework for robust statistical evaluation of results from an AIAA Drag Prediction Workshop (Levy et al, 2002). The Workshop goal was to determine the state of the art for CFD transonic drag predictions of subsonic transports in an industrial setting. Three types of statistical graphical methods were used: (1) running records of individual outcomes, including centerlines and scatter bands, (2) histograms[116] of individual outcomes, and (3) analysis of means. The first two techniques have the advantage of displaying *all the data*, which addresses one of the biggest concerns in statistical analysis, that of possibly drawing incorrect conclusions from aggregated data. For a Workshop, there was a large and arguably statistically significant participation; e.g. for the cruise point calculations the participation included 14 codes, 35 solutions, 3 turbulence models, and roughly equal use of structured and unstructured grids. The statistical approach was based on "N-Version testing" to assess reproducibility as if the individual computed solutions were obtained from a replicated measurement process. In this respect it is similar to the (later) MV&V approach criticized in Section 5.13, but the difference is that the present approach is not used to evaluate grid convergence (Calculation Verification) and does not attempt Validation in the absence of experimental data. To the contrary, the experimental data used are high quality. However, the assessment results are somewhat corrupted by the same problems described in Section 5.13, namely, inability to distinguish discrepancies due to lack of grid convergence. Required grids were provided by the Workshop organizers but some participants used additional grids. That the effect was not isolated is shown by the fact that there was a significant difference in results for the two classes of grids, structured and unstructured, demonstrating that near grid independence was not achieved. (See Section 4.7.)

Judgment of relative success of the assessment of state of the art hinges on what one conceptually includes in state of the art. If the grids used (including those supplied by the Workshop organizers) are considered part of the state of the art, then the method of assessment is valid. Certainly any assessment of state of the art requires "obtaining results, in a collective sense, for diverse codes, grids, turbulence models, and observers" and "computational platforms." But if the computations were done well, either there would be no significant effect of grids (negligible numerical uncertainty estimates) or the effect of numerical uncertainties could be isolated from the other contributors like turbulence models, which should be the only essential contributor, i.e. the *model error* or *model form error* per se. Later AIAA Drag Prediction Workshops[117] have encouraged Calculation Verification. See http://aaac.larc.nasa.gov/tsab/cfdlarc/aiaa-dpw.

Any attempt to evaluate a state of the art for any area will be limited by the necessarily vague defining population. Participation in a Workshop is affected by many factors, including available funding, other commitments, etc. This is further complicated by discarding of "outliers" which, though necessitated by common sense, amounts to an *a posteriori* re-definition of the participating community. Aside from the emphasis in this book on systematic methods for V&V, it is probably true that the most practical informal

[116] There is no time element in the data, so the histogram *x*-axis is just an arbitrarily ordered integer designator for each contributor, a technique used in other Workshops as well, notably the Lisbon V&V Workshops (Eça et al, 2005, 2007a, 2009).

[117] Other references for the ongoing AIAA Drag Prediction Workshops include Levy et al (2003), Hemsch (2004), Hemsch and Morrison (2004), Rumsey et al (2005), Laflin et al (2005), Morrison and Hemsch (2007), Sclafani et al (2008), Mavriplis et al (2009), Vassberg et al (2007, 2008a,b). Vassberg et al (2008a) is a summary paper introducing six papers (e.g. Sclafani et al, 2008) from the Third DPW in a special issue of the AIAA *Journal of Aircraft*, Vol. 45, No. 3, May-June 2008.

way of assessing state of the art in computational PDEs is simply to run a few different codes and compare results, as suggested by Hemsch (2002b).

10.25[118] Δ JOINT CONSIDERATION OF EXPERIMENTAL AND SIMULATION UNCERTAINTIES

In Section 10.5, we noted the difficulty of possible cancellation of conceptual modeling errors and numerical errors (primarily inadequate grid convergence). "Issues related to the interaction of these two sources of errors have been of constant concern to the CFD and turbulence modeling communities because fortuitous cancellation of errors can lead to a very erroneous conclusion about Validation." (Rizzi and Vos, 1996.) The importance of performing Verification of Calculations *before* Validation was stressed. (See also Chapter 2.) However, this admonition does not address the question of what to do with the purely numerical error bar during the process of Validation. The safest approach is simply to reduce the purely numerical error bars to a level negligible compared to the experimental uncertainty. This may be uneconomical.

Coleman and Stern (1997) proposed an approach to a Methodology of Validation that included, for the first time, joint consideration of both experimental and simulation uncertainties. It was incorporated into the ITTC (2002) Manual V&V method. Theirs was a detailed, thoughtful and thought-provoking approach, elucidating the complexity and interactions of the combined experimental and numerical uncertainty issues. This later came to fruition in the ASME ANSI Standard V&V20 (see Chapter 11).

As a preliminary consideration, Coleman and Stern (1997) discussed hypothetical data-simulation comparisons as given in their Figure 1 (see also Coleman, 1996), similar to Figure 10.25.1. Parts *a* and *b* respectively show the data without and with experimental uncertainty bars. The authors stated that one might interpret Part *a* to indicate that Model 1 is superior to Model 2 since it seems to "capture the trend of the data." In contrast, when the experimental uncertainty is considered in Part *b*, "it is obvious that arguing for one method over another based on comparison with the experimental data is wasted effort since the predictions from both methods fall well within the data uncertainty." This negative evaluation was somewhat over-stated. It would be true if and only if the uncertainty distribution within the plotted range of experimental uncertainty (commonly called "error bars") was flat, i.e. a step function. (For example, if the "error bar" covered the data value D to ±10%, this flat distribution would give equal probability that the true value was at D +1%, D-3%, D+9%, etc. or at D itself.) In such a case, our probabilistic knowledge is indifferent to any values within the range of the "error bar" and therefore to the relative merits of Model 1 and 2, and the authors' assessment holds. Of course, we do not know what the appropriate probability distribution is, but most physical experiments surely have some preference for the data values reported, and the default model of probability is a Gaussian distribution. If, in the judgment of the analyst, this is the more likely distribution, then there probably is a trend in the true values, and Model 1 does indeed "capture the trend of the data." One could express this quantitatively as a weighted integral over the assumed probability distributions within the error bars; qualitatively, the naive preference for Model 1 is justifiable. (For discussions, see Roache, 1998c and Coleman and Stern, 1998.)

After this preliminary consideration, Coleman and Stern (1997) proceeded to a joint consideration of both experimental and simulation uncertainties. They defined the "comparison error" E as the difference between the experimental Data set value D and the value produced by the Simulation S.

[118] The order of Sections 10.23 and 10.25 have been exchanged from VV1.

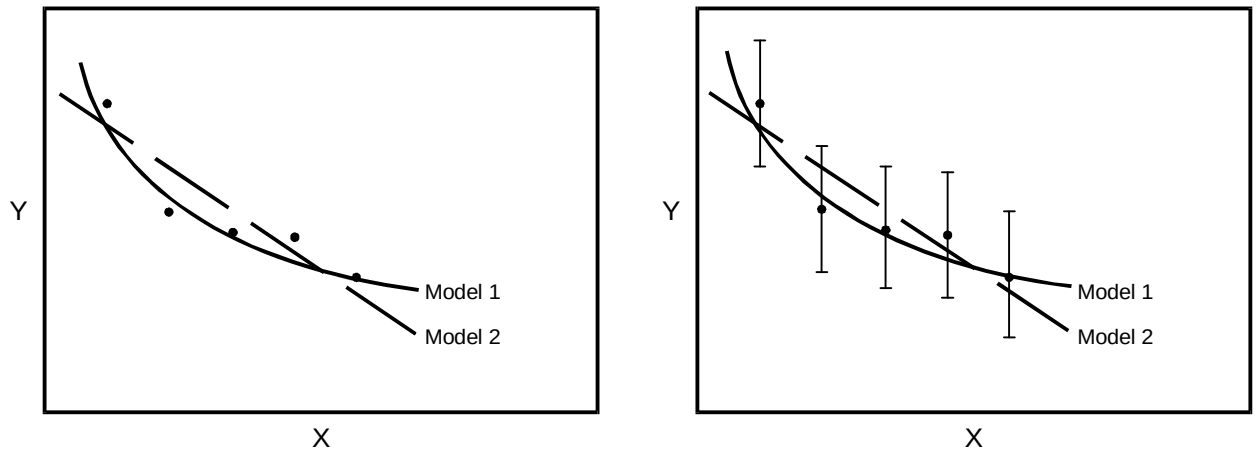


Figure 10.25.1 Comparison of experimental data and model predictions.

$$E = D - S \tag{10.25.1}$$

These may be point values (e.g., a drag coefficient C_D at given flight conditions) or 2-D plots (e.g., C_D vs. α) or multi-dimensional plots (e.g., C_D vs. α, *M*, *Re*, ...) *E* is then just what everyone would naively look at in a Validation exercise. There are experimental uncertainties associated not only with answer C_D but also with the parameters (α, *M*, *Re*, ...) so the experimental error "bar" about C_D should really be an *n*-dimensional error "box" about α, *M*, *Re*, etc. With an assumption of independence of error sources (the subtleties of which are not the main concern here) one can denote the uncertainty values *U* in the usual way, i.e. that the true value lies within $D \pm U$ in 95 times out of 100 (95% $\cong 2\sigma \cong$ 20:1 odds). Again assuming independence of the error statistics, the different sources of uncertainty were combined in an RMS sense. The key step in their approach was to define the metric for Validation, the Validation Uncertainty U_V as the combination of all uncertainties that we know how to estimate (i.e., all but the model form uncertainty). Their proposal was that this *Validation uncertainty U_V is the key metric in the Validation process*, and was to be used as follows.

If the "comparison error" $E = D - S$ is such that $|E|$ is less than the Validation uncertainty U_V,

$$|E| \le U_V \tag{10.25.2}$$

then the Validation has been achieved at the U_V level.

That is, the combination of all the errors in the experimental data set *D* and in the results of the simulation *S* is smaller than the estimated Validation uncertainty U_V, and Validation has been achieved at the U_V level.

The reasoning is that U_V is the key metric because it is the Validation "noise level" imposed by the uncertainties – *all* the uncertainties: uncertainties inherent in the experimental data set *and* uncertainties in the numerical solution (including parametric uncertainties as well as uncertainties due to the use of previous experimental data in the simulation model). A key point is that "one cannot discriminate once $|E|$ is

less than this, i.e., as long as $|E|$ is less than this one cannot evaluate the effectiveness of proposed model improvements."[119]

Not surprisingly, the ramifications of this pioneering work contained some paradoxes, already recognized by the authors in the original work and discussed further in Roache (1998a, Section 10.23; 1998c) and Coleman and Stern (1998). These paradoxes now are all circumvented and clarified by the removal of a specified error tolerance for Validation, or pass/fail criterion, from the concept of Validation and relocating it into Certification (or Qualification, or some other project-specific term) as discussed extensively in Chapter 2 and Appendix B. In fact, these considerations were contributors to the maturation of the Coleman and Stern approach using Validation Uncertainty in the ASME ANSI Standard V&V20, the subject of the following Chapter 11.

[119] If the view given earlier of the interpretation of Figure 10.25.1 in regards to Model 1 "capturing the trend in the data" is accepted, then the original statement would have to be tempered somewhat for an assumed Gaussian distribution rather than flat probability distribution, but basically, the point of the authors is well taken.

CHAPTER 11

§ VALIDATION UNCERTAINTY: ASME ANSI STANDARD V&V 20

11.1 § INTRODUCTION[120]

The most significant development in Validation since the publication of V&V1 in our opinion has been the publication in 2009 of ASME Committee PTC-61 (2009), *ANSI Standard V&V 20, Guide on Verification and Validation in Computational Fluid Dynamics and Heat Transfer* (or V&V20, in our shorthand citation). The methodology involves the application of the internationally accepted approach for *experimental* uncertainty to a total Validation Uncertainty including numerical uncertainty. It had long been recognized that computational methodology in Validation had lagged experimental methodology[121], and calls had been made for computationalists to learn from experimentalists, but it was not until Coleman and Stern (1997; see also Section 10.25) that an initial effort was made. This seminal work was incorporated into the ITTC (2002) Manual V&V method, and was later completed and superseded in V&V20. The methodology is not difficult to follow. It results in an estimate of total Validation Uncertainty, including the contributions of experimental, parametric, and numerical uncertainties. The interpretation that this approach enables is more useful than that of the earlier practice which we have described as an "error

[120] This is a new Chapter, not included in V&V1.

[121] This is not to imply that widespread experimental *practice* meets the standards, nor that the standards have been established for many years. In NIST TN 1297 (Kuyatt and Taylor, 1994) the Director of the U. S. National Institute of Standards and Technology noted that "there has never been a uniform approach at NIST to the expression of uncertainty" and it was not until 1992 that he established a NIST Ad Hoc Committee on Uncertainty Statements which led to TN 1297.

bar" approach (Chapter 10), i.e. presenting the computational results with numerical error bars (i.e. the GCI) and parametric uncertainties and comparing (usually in graphical presentation) to experimental results with experimental error bars. This older and simpler methodology is often adequate and is immediately intuitive; if all publications attained this level of completeness, this would be a success. But the total Validation Uncertainty methodology enables more precision in the interpretations, notably when the experimental and numerical error bars overlap, and provides more explicit guidance on whether or not model improvements can be pursued with present experimental data, or whether more accurate experiments are required prior to model improvements. The most informative presentation utilizes both the traditional[122] error bar approach and the Validation Uncertainty.

The V&V20 methodology is straightforward to apply in the most common situation of (a) using "expanded" uncertainty estimates $U_{95\%}$ and (b) assuming independence of experimental, parametric, and numerical uncertainties. Departure from (a) to use standard uncertainty u leads to some complications, since numerical uncertainties obtained by GCI and similar methods are not based on u but $U_{95\%}$. Departures from (b) require additional calculations that are straightforward conceptually (either chain rule numerical differentiation or Monte Carlo calculations) but are tedious to carry out (see the detailed presentation in V&V20). The methodology will be first described for u, following V&V20, then for $U_{95\%}$, followed by distinctions and further discussion of u vs $U_{95\%}$.

11.2 § V&V20 BACKGROUND AND MOTIVATION

The total Validation Uncertainty approach of ANSI Standard V&V20 is based on the concepts and definitions of error and uncertainty that have been internationally codified by the experimental community over several decades in the ISO Guide (ISO, 1995)[123]. This ISO guide was the culmination of an effort begun in 1977 by seven international organizations[124] to address the "lack of international consensus on the expression of uncertainty in measurement." The ISO Guide has been accepted as the *de facto* international standard for the expression of uncertainty in measurement. This acceptance includes promulgation by the U.S. National Institute of Standards and Technology (NIST; see Taylor and Kuyatt, 1994) and the publication ASME PTC 19.1-2005 "Test Uncertainty" (ASME, 2005) which is treated as a companion document by V&V20.

The approach of V&V20 applies these concepts to the errors and uncertainties in the experimental result and to the errors and uncertainties in the result from the computation. Thus, the uncertainties in the experimental value and in the simulation value are treated using the same process. Using the approach of the ISO Guide, for each error source (other than the simulation modeling error itself, which is the target of any validation effort) a standard uncertainty u is estimated such that u is the standard deviation of the parent population of possible errors from which the current error is a single realization. This allows estimation of a range within which the simulation modeling error lies.

The limited objective validation involves the comparison of computational solution and experimental data *for a specified variable at a specified validation point (experimental "set point")* for cases in which the conditions of the *actual experiment are simulated*, expressed as *the experiment "as run."* For example, in computational aerodynamics, computational results are to be validated against wind tunnel

[122] Oberkampf and Trucano (2007, 2008), who coined the faintly pejorative descriptor "viewgraph norm" to distinguish these from their initial efforts at statistical validation metrics, also agree that such "traditional graphical comparisons should be included" in Validation presentations.

[123] Commonly referred to as "the GUM" from the title "*Guide to the Expression of Uncertainty in Measurement*."

[124] See V&V20 for a list of organizations and details of the history of this international effort.

results, with no consideration of the applicability of the wind tunnel tests to free flight, which is a separate subject. The experiment "as run" is defined as the reality of interest, so the conditions of the actual experiment are the "validation point" that is simulated. Usually a validation effort will cover a range of conditions within a domain of interest; in fact, this is highly recommended. Consideration of the accuracy of simulation results at points within a domain other than the validation points (interpolation/extrapolation in a domain of validation) was not considered in V&V20, nor in V&V10 or other such documents. The subject has been treated as a matter of engineering judgment specific to each family of problems and therefore beyond the scope of a general methodology. Although this will remain true to some extent, there has been recent progress in this area, which will be treated briefly in Section 11.12.

Neither the present book nor V&V20 expends much effort to management considerations of V&V for large projects, which is the preoccupation of the AIAA Guide and V&V10, and which is treated in Oberkampf and Roy (2009). But V&V20 expresses the following understanding of programmatic issues for large V&V projects.

"*Ideally, as a V&V program is initiated, those responsible for the simulations and those responsible for the experiments should be involved cooperatively in designing the V&V effort.* The validation variables should be chosen and defined with care. Each measured variable has an inherent temporal and spatial resolution, and the experimental result that is determined from these measured variables should be compared with a predicted result that possesses the same spatial and temporal resolution. If this is not done, such conceptual errors must be identified and corrected or estimated in the initial stages of a V&V effort or substantial resources can be wasted and the entire effort compromised."

11.3 § ERRORS AND UNCERTAINTIES

Pertinent definitions and descriptions from ASME 19.1 (2006) and used in V&V20 follow.

- δ error (of measurement): "result of a measurement minus a true value of the measurand"
- u uncertainty (of measurement): "parameter, associated with the result of a measurement, that characterizes[125] the dispersion of the values that could reasonably be attributed to the measurand"

Also, from the first edition of ASME 19.1 (1984):

- u uncertainty (of measurement): "an estimate characterizing[126] the range of values within which the true value of a measurand lies"

Thus, an error δ has a particular *sign and magnitude*. It is assumed that any error whose sign and magnitude are known already has been removed by correction, so any remaining error is of unknown sign and/or[126] magnitude. Then an uncertainty u is estimated with the idea that $\pm u$ characterizes[126] the range containing δ. In experimental uncertainty analysis (ISO, 1995) u is the *standard uncertainty* and corresponds conceptually to an estimate of the standard deviation σ of the parent distribution from which δ is a single realization. This definition allows u to be calculated for repeat experiments (aleatory

[125] The term "characterize" is carefully chosen, and signifies *less* than what one might expect; the full implications may not be appreciated until later discussion in Section 11.9.1.

[126] Usually both sign and magnitude are unknown. If the sign is known, that information is lost in this most common approach.

uncertainty) by a simple Root-Mean-Square (RMS) operation, also described as Root-Sum-Square (RSS). For n replications of an experimental measurement f, the sample[127] standard deviation is calculated from

$$f_{mean} = \frac{1}{n}\sum_{i=1}^{n} f_i \tag{11.3.1a}$$

$$u = \left\{ \frac{1}{n-1}\sum_{i=1}^{n} [\, f_i - f_{mean} \,]^2 \right\}^{1/2} \tag{11.3.1b}$$

or for the "population" standard deviation,

$$u = \left\{ \frac{1}{n}\sum_{i=1}^{n} [\, f_i - f_{mean} \,]^2 \right\}^{1/2} \tag{11.3.1c}$$

as appropriate. It is significant to note that no assumption about the form of the parent distribution is associated with the definition of u based on standard deviation. Much can be accomplished in V&V without assuming a formal distribution.

11.4 § DEFINING VALIDATION UNCERTAINTY

Consider a variable of interest, for example, temperature T_o. Following V&V20, we denote the predicted value of T_o from the simulation solution as S, the value determined from experimental data as D, and the true but unknown value as T. The validation comparison error[128] E is defined as

[127] This most commonly used sample standard deviation includes the Bessel correction 1/(n-1) rather than 1/n. The population standard deviation using 1/n may be appropriate, and NIST TN 1297 uses another RMS averaged form in its Eq. (A-5). The distinction seems to be based on modeling philosophy, and appears vague and somewhat subjective to this non-statistician. If the analyst intends to characterize the spread or variability of the given data set, i.e. considering it to be the entire population, then the population standard deviation using 1/n is used. But if he intends to characterize the variability of a larger population from which the given data set is merely a sample, which is usually the case, then the sample standard deviation using 1/(n-1) is used. (The 1/n form would be a "biased estimator" although this is not necessarily bad. The 1/n form is also appropriate if the mean is calculated without using Eq. (11.3.1.a), using either the mean of the entire population, or perhaps an estimate from another sample.) Wolfram Math World notes that there is "widespread inconsistent and ambiguous terminology." I would add "overly subtle" and "sometimes deceptive," as when distinguishing between "sample standard deviation" using 1/(n-1) and "standard deviation of the sample" using 1/n. Some authors use 1/n, others use 1/(n-1), as a default definition, so that a report of a value for "standard deviation" without a qualifier of "sample" or "population" may be ambiguous. Even "sample" has been used inconsistently, sometimes for 1/n and other times for 1/(n-1). "Unbiased" dependably indicates use of 1/(n-1). Fortunately, for large n the distinction is insignificant compared to other vagaries of "uncertainty" for the applications considered herein.

[128] Eq. (11.4.1) actually defines E as merely a discrepancy rather than an error at this point in the development, but E can be argued to be properly an error by Eq. (11.4.6).

$$E = S - D \tag{11.4.1}$$

The error in the solution value S is the difference between S and the true value T.

$$\delta_S = S - \mathrm{T} \tag{11.4.2}$$

The error in the experimental value D is

$$\delta_D = D - \mathrm{T} \tag{11.4.3}$$

From Eqs. (11.4.1-3), E is expressed as

$$E = S - D = \left(\mathrm{T} + \delta_S\right) - \left(\mathrm{T} + \delta_D\right) = \delta_S - \delta_D \tag{11.4.4}$$

The validation comparison error E is thus the combination of all of the errors in the simulation result and the experimental result, and its sign and magnitude are known once the validation comparison is made.

All errors in the simulation solution S can be assigned to one of three categories (Coleman and Stern, 1997):

- error δ_{model} due to modeling assumptions and approximations
- error δ_{num} due to the numerical solution of the equations, and
- error δ_{input} due to errors in the simulation input parameters.

Thus

$$\delta_S = \delta_{model} + \delta_{num} + \delta_{input} \tag{11.4.5}$$

The objective of a validation exercise is to estimate δ_{model} to within an uncertainty range. Using Eq. (11.4.5) in (11.4.4), the comparison error *E* is

$$E = \delta_{model} + \delta_{\text{num}} + \delta_{input} - \delta_D \tag{11.4.6}$$

or

$$\delta_{model} = E - (\delta_{num} + \delta_{input} - \delta_D) \tag{11.4.7}$$

Once the simulation and the experiment are run, S and D are determined, and E is known (both sign and magnitude, from Eq. 11.4.1), but the signs and magnitudes of δ_{num}, δ_{input}, and δ_{D} are unknown. The standard uncertainties corresponding to these errors[129] are u_{num}, u_{input}, and u_{D}. Then the key step of the V&V20 methodology is defining the *validation standard uncertainty* u_{val} as an estimate of the standard deviation of the parent population of the combination of errors (δ_{num} + δ_{input} - δ_{D}). Considering the relationship shown in Eq. (11.4.7), the interval = (E ± u_{val}) *then characterizes an interval within which* δ_{model} *falls,* or

[129] In the conceptual approach of the ISO Guide (ISO, 1995), ASME PTC 19.1 (ASME 1986, 2006), and NIST TN 1297 (Taylor and Kuyatt, 1994) there is no distinction made in the *mathematical treatment* of errors that are "random" and those that are "systematic." After S and D have been determined, their values always differ by the same fixed amount from the true value; so δ_{D}, δ_{input}, δ_{num}, and δ_{model} *effectively* are all systematic errors and the uncertainties to be estimated are systematic standard uncertainties.

$$\delta_{model} \in [E - u_{VAL}, E + u_{VAL}]. \tag{11.4.8}$$

In other words, E and u_{val} are the validation metrics for the modeling error δ_{model}. *The estimation of u_{val} is thus at the core of the V&V20 methodology.*

11.5 § ESTIMATING VALIDATION STANDARD UNCERTAINTY

If the three errors δ_{num} , δ_{input} and δ_D are effectively independent, then the corresponding uncertainties u_{num} , u_{input} and u_D can be easily combined by the usual statistical assumption.

$$u_{val} = \sqrt{u_{num}^2 + u_{input}^2 + u_D^2} \tag{11.5.1}$$

Fortunately, this condition of effective independence is often practical. In the important case in which the validation variable T_o is *directly measured*, the assumption of effectively independent errors is generally reasonable. Unfortunately, however, in the common case in which the validation variable is determined using a data reduction equation, the experimental and computational values can be functions of shared variables, and δ_{input} and δ_D are not independent.[130]

Much of V&V20 (Section 5) is devoted to explanations with detailed examples of various cases of estimating u_{val} when the errors are not independent and u_{val} must be estimated in a tightly coupled procedure to avoid problems. One particular problem is the case where the effects of a parameter uncertainty should approximately cancel between u_{input} and u_D if treated correctly, but will contribute two terms if Eq. (11.5.1) is used. The variations include situations in which the validation variable is determined using a data reduction equation that combines multiple variables each directly measured, with or without shared error sources. The methods are conceptually simple, calculating the non-independent effects by tracking the mutual influences through data reduction equations using either of two approaches: local sensitivity equation evaluated by numerical differentiation (sometimes partially analytical), or global Monte Carlo methods. (These methods are also applicable to the estimation of all contributors to u_{input}.) However, the calculations are tedious, and the simpler use of Eq. (11.5.1) is expected to be more common. It is also true that the error of this simpler approach often will be conservative, since the dependent effects will often reduce the estimate of the Validation Uncertainty.

11.6 § PARAMETER ERRORS AND MODEL FORM ERRORS

Since the uncertainty contributions to u_{val} take into account all of the error sources in δ_{num}, δ_{input}, and δ_D, then δ_{model} includes only errors arising from modeling assumptions and approximations; these are the "model form" errors. For example, in a simple heat conduction problem, deviation from the correct value of a constant conductivity *K* would be part of the input parameter error δ_{input}, while deviation from the assumption of constant *K* (i.e. neglect of dependence of *K(x,y,z,T,...)* and neglect of tensor vs. scalar

[130] To a lesser extent, δ_{num} can be dependent on δ_{input} because adequate grid resolution depends on the parameters. Usually the parameter uncertainty range does not strongly affect δ_{num}. If it does, its dependency must also be accounted (V&V20).

conductivity would be part of model form error. In practice, numerous gradations[131] can exist in the choices of which error sources are accounted for in δ_{input} and which error sources are defined as an inherent part of the model form error δ_{model}. The term δ_{model} is the modeling error which we intend to assess. It is composed of the errors in the governing *continuum* equations of the model (e.g. the RANS model used) and errors due to any other non-ordered approximations such as inflow and outflow boundary conditions; these errors do *not* $\rightarrow 0$ as $\Delta \rightarrow 0$ (where Δ is a representative measure for the grid cell size). The error δ_{num} is composed of the ordered numerical errors; these errors *do* $\rightarrow 0$ as $\Delta \rightarrow 0$. Likewise, the error δ_{input} is composed of the (non-ordered) errors arising from the use of incorrect parameter values in the model equations. Note that the symbol δ_{num} is somewhat ambiguous in one respect. Errors that result from the imposition of finite computational boundaries, e.g., finite distance to a downstream outflow boundary (Section 6.10) or other far-field boundary (e.g. see Roache, 1998b), can be thought of as "numerical" in some sense, but they are not included in δ_{num}, because they are not ordered in Δ. Rather, they are included in δ_{model}.[132]

The code used will often have more adjustable parameters or data inputs than the analyst may decide to use (e.g., for a commercial code). The decision of which parameters to include in the definition of the computational *simulation* (conceptually separate from the *code*) is somewhat arbitrary. Some (even all[133]) of the parameters available may be considered fixed for the simulation. For example, an analyst may decide to treat parameters in a chemistry package as fixed ("hard-wired") and therefore not to be considered in estimating u_{input}, even though these parameters could have been accessed and had associated uncertainties. The point here is that the computational simulation which is being assessed consists of the code and a selected number of simulation inputs which are considered part of the simulation, while other simulation inputs have uncertainties that are accounted for in u_{input} and thus do not contribute to δ_{model}. See Appendix C of V&V20 for related discussion of specific and general senses of *model*, and parametric uncertainties vs. model form uncertainties.

This distinction is also required to explain the following paradox. As the analyst improves the thoroughness of a validation study by investigating parametric uncertainty more extensively, the total validation uncertainty will become larger, not smaller. Every additional parameter variation considered will add to δ_{input} . The resolution of the paradox lies in recognizing that, with every addition of a parameter uncertainty (e.g. considering variable conductivity of $K(x,y,z,T,...)$ instead of fixed K) one is changing the "model' under evaluation. In the limit of a strong model approach, with all parameter values hard-wired, there simply is no parametric uncertainty; $\delta_{input} = 0$ and $u_{input} = 0$.

---

[131] Virtually all model form errors could be parameterized and thereby included in δ_{input}. Even such disparate model forms as Darcy flow vs. fracture flow in porous media could be parameterized by (admittedly arbitrary) blending functions. An appreciation for such distinctions is essential to clarifying some muddled claims of philosophy of science, e.g. Popper (1980), Oreskes et al. (1994). See Appendix C of V&V1.

[132] As pointed out in Roache(1998b), the imposition of far-field boundaries at finite distances are reflexively thought of as inherently numerical, but it is just an accident of mathematics that boundary conditions "at ∞" happen to be easier to apply in many analytical solutions. There is nothing inherently "exact" about applying the free-stream conditions of constant velocity and fluid properties "at ∞" to model free flight when (for example) an exponential atmosphere model would be a better approximation to nature.

[133] If all parameter values are considered fixed in the model, this is the limit of what has been termed a strong-model approach; see Section 2.2.

11.7 § ESTIMATING PARAMETRIC UNCERTAINTIES

Considering the relationship shown in Eq. (11.5.1), obtaining an estimate of u_{val} requires estimates of u_{num}, the standard uncertainties in the experiment that contribute to u_D, and the standard uncertainties in all input parameters that contribute to u_{input} . Obviously, estimates of the standard uncertainties of all of the input parameters treated as variable (rather than hard-wired) in the model are required. Then u_{input} is determined from propagation by either (1) using a sensitivity coefficient (local) method that requires estimates of simulation solution sensitivity coefficients, or (2) using a Monte Carlo (sampling, global) method that makes direct use of the input parameter standard uncertainties as standard deviations in assumed parent population error distributions. Details are presented in Section 3 of V&V20. The standard uncertainty in the experimental result u_D is determined using well-accepted techniques developed by the international community over a period of decades and is discussed in Section 4 of V&V20. The estimate u_D is the standard uncertainty appropriate for D. It includes all effects of averaging, all random and systematic components, and effects of any correlated experimental errors/uncertainties and any other factors that influence D and u_D. As noted previously, when D and u_D are used in the validation comparison, any random uncertainty components have been effectively frozen and u_D is calculated as a *systematic* standard uncertainty.

A comprehensive and highly recommended end-to-end example of the application of the V&V20 methodology is presented and discussed in Section 7 of V&V20. For a recommended overview of sensitivity analysis and uncertainty propagation, see Blackwell and Dowding (2006).

11.8 § INTERPRETATION OF VALIDATION RESULTS USING STANDARD UNCERTAINTIES

The power of the V&V20 approach based on total Validation Uncertainty becomes evident when the interpretation of Validation results is considered.

Note again: once a validation effort reaches the point where the simulation value *S* and the experimental value *D* of a validation variable have been determined, then the sign and magnitude of *E (= S - D)* are known.

The validation uncertainty u_{val} is an estimate of the standard deviation of the parent population of the combination of all errors except the modeling error $(\delta_{num} + \delta_{input} - \delta_D)$ in *S* and *D*. Techniques for estimation of the uncertainty components u_{num}, u_{input}, and u_D that combine to give u_{val} have been discussed, and evaluation of u_{val} from those uncertainty components is accomplished by Eq. (11.5.1) in the simplest and often realistic cases, or by more elaborate methods as required.

We repeat here Eqs. (11.4.7-8) as

$$\delta_{model} = E - (\delta_{num} + \delta_{input} - \delta_D) \tag{11.8.1}$$

$$\delta_{model} \in [E - u_{VAL}, E + u_{VAL}]. \tag{11.8.2}$$

The interval of Eq. (11.8.2) will then "characterize" an interval within which δ_{model} falls; E and u_{val} are the validation metrics for the modeling error δ_{model}. Thus *E* is an estimate of δ_{model}, and u_{val} is the standard uncertainty of that estimate.

11.8.1 § Interpretation of Validation Results Using E and u_{val} Without Assumptions Made about Error Distributions

If one has only an estimate for the validation uncertainty u_{val} and not an estimate of the probability distribution associated with $(\delta_{num} + \delta_{input} - \delta_D)$, an interval within which the value of δ_{model} falls with a given probability cannot be estimated without further assumption. One can make the following statements, however.

Case 1. If

$$|E| >> u_{val} \tag{11.8.1.1}$$

then probably

$$\delta_{model} \approx E \tag{11.8.1.2}$$

Case 2. If

$$|E| \leq u_{val} \tag{11.8.1.3}$$

then probably

$$O(\delta_{model}) \leq O(\delta_{num} + \delta_{input} - \delta_D). \tag{11.8.1.4}$$

That is, probably δ_{model} is of the same order as, or less than, $(\delta_{num} + \delta_{input} - \delta_D)$.

The significance of distinguishing these two cases is illustrated when one considers not just the evaluation of a computational model, but the possibility of improving the model. From a practical standpoint, in the first case one has information that can possibly be used to improve the model, i.e. reduce the modeling error. But in the second case, the modeling error is within the "noise level" imposed by the numerical, input, and experimental uncertainties and formulating model "improvements" is more problematic. That is, an analyst could hardly justify[134] tuning parameters in the model form used, or changing the model form, without first reducing some or all of δ_{num} , δ_{input} , and δ_D. Respectively these would require (a) improved computational accuracy and/or uncertainty estimation, and/or (b) reduced parametric uncertainty (perhaps by using more complete dependency than the independence assumed for Eq. (11.5.1) or by further analysis justifying reduced input parameter ranges), and/or (c) improved experiments. Obviously, one could consider reducing error source that is dominant, or most accessible, or cheapest to improve.

This interpretation of Case 2 is more evident with the total Validation Uncertainty methodology of V&V 20 than with the older error bar approach, and constitutes a major advantage of the approach in our opinion (e.g. see Pelletier, 2008). An example of similar interpretation using probabilistic (rather than standard) Validation Uncertainty will be given in Section 11.10.

11.8.2 § Interpretation of Validation Results Using E and u_{val} With Assumptions Made about Error Distributions

In order to estimate an interval within which δ_{model} falls with a given probability or degree of confidence, an assumption (with possible ranges of justification) about the probability distribution of the

[134] There is no justification if the error distribution is rectangular. However, without necessarily assuming a Gaussian distribution, one would usually expect the model to be improved by tuning that drives the simulation value S towards the center of the distribution. See discussion and References in Section 10.25. This would be a secondary improvement of confidence. A common engineering rule of thumb is to cease tuning when S is within 1-σ of the center of the distribution (Hills, 2008).

combination of all errors except the modeling error must be made. This then allows the choice of a coverage factor *k* (ISO, 1995; Coleman and Steele, 2009) such that

$$U_{xx\%} = k\,u \tag{11.8.2.1}$$

where $U_{xx\%}$ is called the *expanded uncertainty*. For example, one might be able to say that ($E \pm U_{95\%}$) defines an interval within which δ_{model} falls about 95 times out of 100 (i.e., with 95% probability) when the coverage factor has been chosen for a level of 95% "confidence" (see footnote #34 on page 122).

To obtain a perspective on the order of magnitude of *k*, consider the three parent error distributions used as examples in the ISO Guide (ISO, 1995).

(a) A uniform (rectangular) distribution[135] with equal probability that δ lies at any value between –A and +A so that $\sigma = A/\sqrt{3}$.

(b) A triangular distribution symmetric about δ = 0 with base from –A to +A so that $\sigma = A/\sqrt{6}$.

(c) A Gaussian distribution with standard deviation σ.

With a decided (assumed or justified) distribution, one can choose a coverage factor *k* such that ($\delta_{num} + \delta_{input} - \delta_D$) certainly (or almost certainly) falls within $\pm k(u_{val})$.

(a) If ($\delta_{num} + \delta_{input} - \delta_D$) is from the uniform distribution, then 100% of the population is covered for $k = 1.73$.

(b) If ($\delta_{num} + \delta_{input} - \delta_D$) is from the triangular distribution, then 100% of the population is covered for $k = 2.45$.

(c) If ($\delta_{num} + \delta_{input} - \delta_D$) is from the Gaussian distribution, then 95.44% of the population is covered for the most commonly used[136] value $k = 2$.

Other less commonly values for the Gaussian distribution are $k = 1$ for 68.26%, $k = 3$ for 99.7%, $k = 3.5$ for 99.95%, and $k = 4$ for 99.99%.

These results are not very sensitive to the form of the distribution, thus making uncertainty analysis practical. With these comparisons, one can conclude that for error distributions roughly in the family of the three distributions considered, it is highly likely that δ_{model} almost certainly falls within the interval $E \pm k(u_{val})$, where *k* is typically a number in the range of 2 to 3.

11.9 § ESTIMATING VALIDATION PROBABILISTIC UNCERTAINTY

Eq. (11.5.1) for combining standard uncertainties extends to "expanded" or probabilistic uncertainties with the same requirement that the errors δ_{num}, δ_{input} and δ_D are effectively independent.

---

[135] The Eurachem/CITAC (2007) guide on measurement uncertainty gives the following guidelines. If ± limits are given without a confidence level and extreme values appear likely, a rectangular distribution is appropriate. If ± limits are given without a confidence level and extreme values appear unlikely, a triangular distribution is appropriate.

[136] The value $k = 2$ is strongly recommended at the U. S. National Institute of Standards and Technology (Taylor and Kuyatt, 1994, paragraph 6.5). "To be consistent with current international practice, the value of *k* to be used at NIST for calculating *U* is, by convention, $k = 2$. Values of *k* other than 2 are only to be used for specific applications dictated by established and documented requirements."

$$U_{val} = \sqrt{U_{num}^2 + U_{input}^2 + U_D^2} \tag{11.9.1a}$$

$$U_{val,95\%} = \sqrt{U_{num,95\%}^2 + U_{input,95\%}^2 + U_{D,95\%}^2} \tag{11.9.1b}$$

Likewise, the methods for treating dependence are applicable to *U*'s (local sensitivity or global Monte Carlo methods). However, there is a subtlety involved that arises from the mixing of standard and expanded uncertainties, discussed below.

11.9.1 § Standard Uncertainty vs. "Expanded" or Probabilistic Uncertainty

The early assumption of Gaussian distributions in so ingrained in the education of engineers and scientists that it requires mental effort to separate Gaussian distribution from other aspects of statistics. Standard uncertainty u is commonly associated with 1-σ or ~68% confidence, and 2-σ with ~ 95% confidence, so people naturally think of $2u \sim U_{95\%}$. The distinction between u and $U_{95\%}$ is implicitly treated as just a matter deciding on the value of the coverage factor k, with $k = 1$ producing u with ~68% confidence, $k = 2$ producing $U_{95\%}$ with ~ 95% confidence, etc. In this common interpretation, the symbol u could just as well be written as $U_{68\%}$. This is wrong.

In fact, without the assumption of Gaussian distribution, u and $U_{68\%}$ are fundamentally different *kinds* of uncertainty definitions. Until a distribution is decided (assumed, or justified) u is not merely a smaller or less inclusive uncertainty than $U_{95\%}$. Following the internationally accepted practice for experimental work[137], u is defined (Eq. 11.3.1) as the (sample) *standard uncertainty* and corresponds conceptually to an estimate of the standard deviation σ of the parent distribution from which δ is a single realization.[138] At the beginning of Section 11.3, we described u (of measurement): "parameter, associated with the result of a measurement, that characterizes the dispersion of the values that could reasonably be attributed to the measurand" and we noted in a footnote that the term "characterize" is carefully chosen; the implications will now be discussed.

Contrary to a natural assumption, "characterize" here does not imply any probability (not even in a Bayesian sense). We cannot say generally, i.e. without assuming a distribution, that the true value "probably" lies within the interval [*mean* ± u] even for the minimalist interpretation of probability > 50%. This perhaps counter-intuitive statement is demonstrated by the following example. Consider the set of five measurements (normalized about the median value): {0.9, 0.9, 1.0, 1.1, 1.1}. The mean value is 1. The (population) variance is calculated as

$$\sigma^2 = [2(0.9 - 1)^2 + (1\text{-}1)^2 + 2(1.1\text{-}1)^2]/5 = 0.04/5 = 0.008 \tag{11.9.1.1}$$

so the estimated standard uncertainty u, or (population) standard deviation, is

---

[137] See ISO Guide (ISO, 1995), ASME PTC 19.1 (ASME 1986, 2006); NIST TN 1297 (Taylor and Kuyatt, 1994), Coleman and Steele (2009), and others.

[138] To be unnecessarily precise, this involves estimates of estimates: σ is standard uncertainty of the infinite sequence (frequentist) or parent population, and s is the sample standard uncertainty which provides an estimate of σ. Then uncertainty u is an estimate of s, which is an estimate of σ.

$$u = \sigma = \sqrt{0.008} = 0.08944... \quad (11.9.1.2)$$

Then the interval

$$[mean \pm u] = [1 \pm 0.08994] = [0.9106, 1.0894] \quad (11.9.1.3)$$

includes only one of the five measurements (the value 1.0). Thus the probability of [*mean* ± *u*] containing a random draw from these experimental values is a mere 20%.

In experimental work, standard uncertainty u (or u_D in Eq. 11.5.1) is naturally and conveniently calculated (at least for aleatory uncertainty, less so for systematic uncertainty). So to obtain what is needed, i.e. a probabilistic statement of uncertainty, it is natural to start from u, which strictly speaking has no probability associated with it, then to assume a distribution (usually Gaussian, although as noted the probabilistic results are not very sensitive to the type of distribution), then to "expand" u by a factor $k > 1$ to calculate some "expanded" uncertainty $U_{xx\%}$, typically Gaussian with $k = 2$ giving $U_{95\%}$. Thus the terminology "expanded" uncertainty makes sense, referring to a probabilistic uncertainty obtained by expanding from a standard uncertainty. However, the terminology is unfortunate because it is sometimes possible to obtain a probabilistic uncertainty without calculating u and without assuming any distribution. The Grid Convergence Index or GCI is just such a probabilistic uncertainty for numerical errors,

$$GCI = U_{num,95\%}. \quad (11.9.1.4)$$

The confirmation of the GCI has been obtained (Chapter 5) by examining data on simulation solutions S for a large number (many hundreds) and keeping track of cases of "failure" (true numerical value outside the predicted interval) and success, with no need to assume any distribution. It is a powerful fact that the **empirical validity of the GCI does not depend on any distribution**. (There is no fundamental reason why a similar approach could not be used for aleatory experimental variation, but this is not common practice.)

Since u is fundamentally calculated as a standard deviation, and since u is claimed to "characterize" the dispersion of the values yet does not imply any probability (even in a Bayesian sense), one might be justified in asking the following. Why does a standard deviation deserve the name "uncertainty"? Good question. If standard deviation had been given the symbol d instead of u, it would be more clear that there is a conceptual leap involved in proceeding from standard deviation d to probabilistic uncertainty $U_{xx\%}$. Nevertheless, use of the term standard *uncertainty* (and the symbol u instead of d) is not only widespread but institutionalized. (See list of organizations in Section 11.2 and V&V20.) It is not going to go away.[139]

In summary, the calculation of *standard* uncertainty u_D for experimental data does not depend on any assumption of Gaussian or other distribution, nor does the calculation of *probabilistic* uncertainty $U_{num,95\%}$ for numerical error. The trouble is that we need to use the same kind of uncertainties to calculate a total Validation Uncertainty, using either *all* standard uncertainties u for Eq. (11.5.1) or *all* probabilistic uncertainties $U_{xx\%}$ for Eq. (11.9.1). To do so, we will need to assume a distribution at some point. And unfortunately, we cannot casually go back and forth between u and $U_{xx\%}$ because the assumption of a Gaussian distribution (or even a qualitatively similar distribution) is demonstrably *not* justifiable for the best behaved numerical solutions, as we will now discuss.

[139] Given the fact that most analyses involve assumptions of distributions, usually Gaussian, which enables the two-way transition $u \Leftrightarrow U_{68\%}$, this use does less harm than other standing problems areas of statistics, e.g. null hypothesis testing.

11.9.2 § Combining Standard Uncertainty and Probabilistic Uncertainty[140]

This sub-section is somewhat convoluted, and a reading is not necessary if the analyst intends finally to base the total Validation Uncertainty on the probabilistic uncertainty $U_{val,95\%}$, e.g. using Eq. (11.9.1b). In this most common case, the reader may jump to the next Section 11.9.3.

As noted, the Grid Convergence Index or GCI (Chapter 5) for the fine grid solution is just a Factor of Safety F_s applied to the (generalized) Richardson error estimate $|E_1|$ as in Eq. (5.6.1) or any other ordered error estimate (e.g. Section 7.5). For not very thorough numerical studies, using only two grids and only the theoretical order of convergence (say $p = 2$), the recommended value is $F_s = 3$. It may be expected that this hypothetical ensemble of computational cases will include some that are not in the asymptotic range of grid resolution, some that display oscillatory convergence, etc. For these cases, it might be reasonable to assume a Gaussian distribution and $k = 2$. We would not use this k to calculate an "expanded" numerical uncertainty $U_{num,95\%}$ since we already know that $U_{num,95\%}$. = GCI. Rather, we might use $k = 2$ to work backwards to calculate a *contracted* estimate[141] of u_{num}, or more precisely $U_{num,68\%}$.

$$u_{num} \sim U_{num,68\%} = U_{num,95\%} / k = \text{GCI} / 2 \quad = (F_s / 2) \times |E_1| \tag{11.9.2.1}$$

For the recommended value of $F_s = 3$, this gives $u_{num} = 1.5|E_1|$, a not unreasonable value. Intuitively, since $|E_1|$ generally corresponds with 50% confidence (see Section 5.14.2), then it is not unreasonable to expect that $1.5|E_1|$ could be consistent with 68% confidence or $U_{num,68\%}$.

The difficulty arises for the best numerical work, using three or more grids to establish a stable observed p that agrees with theoretical p, and monotonic convergence. In such cases, the recommendation is $F_s = 1.25$. Blindly using Eq. (11.9.2.1) would then lead to $u_{num} = 0.625|E_1|$, which is not only unreasonable but contradictory: an intended 68% confidence interval smaller than a 50% confidence interval. The cause is evidently the assumption of a Gaussian distribution for the errors. In these best behaved numerical cases, monotonic convergence indicates that the true (numerical) solution is outside the set of all grid solutions, i.e. it is extrapolated. (See related discussion in Section 5.13.) Thus it is not reasonable to assume that the error distribution is even qualitatively Gaussian about the fine grid solution f_1; rather, it is likely a shifted Gaussian, i.e. Gaussian not about the fine grid solution f_1 but about the Richardson extrapolated solution $f_{\text{extrap}} = f_1 - E_1$. (Roache, 2003a)

None of these difficulties arise if the analyst decides to use probabilistic uncertainties $U_{95\%}$ for all terms, i.e. Eq. (11.9.1) rather than (11.5.1). Any standard uncertainties (u_{input} and u_D) are "expanded" to 95% probabilistic uncertainties and used in Eq. (11.9.1). (See, for example, the Proceedings of the Third Lisbon V&V Workshop, Eça and Hoekstra, 2008.) If the experiments are already reported using $U_{95\%}$, this step is not needed. If, however, the analyst needs to use standard uncertainties everywhere, an estimate of u_{num} must be obtained from the GCI using a coverage factor k_s appropriate for the *shifted* Gaussian. A rough estimate can be obtained by linear interpolation between the known values.

---

[140] I am indebted to Dr. V. Romero for his perceptive review of the initial public draft release of V&V20, which review led to these insights.

[141] This approach is recognized as "Type B" evaluation of u by NIST TN 1297 (Taylor and Kuyatt, 1994) in paragraph 4.3. Type B evaluations of u are not obtained by direct statistical methods, and are "usually based on scientific judgment." It is noted that "Type A evaluations of uncertainty based on limited data are not necessarily more reliable than soundly based Type B evaluations." There is not always a simple correspondence between Types A and B and "random' and "systematic."

k_s = Multiplier of $\|E_1\|$	Coverage level
1.25	95% (empirical)
1.10	68% (target, interpolated)
1.00	50% (ordered error estimate)

Another estimate was obtained by Hills (2008) (see also V&V20, Section 2) by analyzing a shifted Gaussian, giving a multiplier $k_s = 1.15$. This larger value of k_s is slightly more optimistic when contracting from GCI to u_{num} , i.e. it gives a slightly smaller estimate of u_{num}.[142]

Continuing this process for greater uncertainty coverage, e.g. targeted 99% coverage, would be difficult to justify. The GCI has been established for 95% confidence, and it would be problematical to naively extend it to (say) 99%.

11.9.3 § Summary Procedure for Probabilistic Validation Uncertainty

If the analyst intends finally to base the total validation uncertainty on the probabilistic uncertainty $U_{val,95\%}$ using Eq. (11.9.1b), the summary procedure is simple. The probabilistic numerical uncertainty $U_{num,95\%}$ = GCI. If the experimentalist has provided probabilistic data uncertainty $U_{D,95\%}$ it is used directly. Otherwise, if the experimentalist has provided data standard uncertainty u_D, the analyst must assume a distribution (probably Gaussian) and corresponding coverage factor (probably $k = 2$) to expand to $U_{D,95\%} = k \times u_D$. (See Section 11.8.2 above for other distributions and coverage factors.) Similarly for uncertainties of input parameters, which probably have been calculated as standard uncertainties u_{input} and need to be expanded to probabilistic uncertainties of input parameters $U_{input,95\%} = k \times u_{input}$. These terms are used in Eq. (11.9.1b) to calculate the probabilistic total Validation Uncertainty $U_{val,95\%}$.

11.10 § INTERPRETATION OF VALIDATION RESULTS USING PROBABILISTIC UNCERTAINTIES

Here we use the more specific and common engineering target U_{val} with 95% level of confidence, rather than standard uncertainty u_{val}. We note again that once a validation effort reaches the point where the simulation value S and the experimental value D of a validation variable have been determined, the sign and magnitude of E *(= S - D)* are known. But now, $(E \pm U_{val})$ defines an interval within which δ_{model} falls, with ~95% coverage or "level of confidence."

$$\delta_{model} \in [E - U_{VAL}, E + U_{VAL}], \quad confidence\,level \sim 95\% . \tag{11.10.1}$$

(This contrasts to Eq. (11.8.2) for u_{val}, which does not have a confidence level specified.)

Case 1. If

$$|E| >> U_{val} \tag{11.10.2}$$

then probably $\delta_{model} \approx E$.

[142] Note that both $k_s = 1.10$ or 1.15 are conservative if the u_{num} calculated this way is later "expanded" back to an estimate of $U_{num,95\%}$ using $k = 2$ along with other Gaussian-distributed terms. For the case of u_{input} , $u_D << u_{num}$, it produces the more conservative estimate $U_{num,95\%} = (k / k_s) \times$ GCI.

Case 2. If

$$|E| \le U_{val} \tag{11.10.3}$$

then probably δ_{model} is of the same order as, or less than, $(\delta_{num} + \delta_{input} - \delta_d)$.

In the first case one has information that can possibly be used to improve the model (i.e., reduce the modeling error). In the second case, however, the modeling error is within the "noise level" imposed by the numerical, input, and experimental uncertainties, and formulating model improvements is more problematic. (See the discussion in Section 11.8.1 after Eq. 11.8.1.3.)

The Third Lisbon V&V Workshop (Eça and Hoekstra, 2008; Eça et al, 2009) provided a good example of such interpretation, and is worth quoting at length. The problem was the classic CFD computation of 2-D turbulent flow over a backstep using the Spalart - Allmaras model, for which there were six submissions at the Workshop. The following quotes are from Eça et al (2009).

...

"Close to the wall, |E| is significantly larger than U_{val} indicating that there is a model deficiency in the prediction of the near-wall backward flow. One step height above the bottom wall, U_{val} becomes similar to |E| due to the increase of the numerical uncertainty showing that the numerical prediction must be improve to draw consistent conclusions about the modeling error. In the center of the computational domain (roughly 2 to 6 step heights) |E| is significantly smaller than U_{val}. In this case, the main contribution of U_{val} is the experimental uncertainty, which in this case is 2% of the incoming flow velocity. In the vicinity of the wall |E| is again larger than U_{val}. However, at this location all the submitted results exhibited the same trend, independent of the turbulence model adopted. We believe that the problem is related to the inlet boundary conditions. There is no experimental information available for the top wall boundary layer. As a guess, the Workshop organization proposed to use equal boundary-layer profiles at the top and bottom walls. The Validation exercise seems to indicate that this was not the proper choice

...

The inclusion of the Validation exercise was an excellent addition to the Workshop. The ASME Validation procedure is clearly a step forward compared to the "standard" graphical comparison between experiments and numerical predictions. It allowed to point out limitations in the modeling, but also to show deficiencies in the numerical simulations and/or in the experiments."

11.11 § MODEL QUALITY VS. VALIDATION QUALITY[143]

It is easy to lose sight of a fundamental fact, related to the easy confusion of error and uncertainty. If U_{val} is unacceptably large, this says *nothing* about poor quality of the model[144].

The magnitude of U_{val} does not reflect upon the quality of the model.

The magnitude of U_{val} increases because of poor computational work, poor parameter estimation, and poor experiments, not from a poor model. It does not depend on δ_{model}. The model quality and the validation quality are different values. The development of a model creates δ_{model} and the performance of a validation exercise (including the execution of the experiment and the *use* of the model in the simulations) creates U_{val}.

A poor quality model combined with a high quality validation exercise leads to $|E| >> U_{val}$ and therefore trustworthy $\delta_{model} \approx E$. If δ_{model} is excessively large for any reasonable application, the result

[143] This material is not explicitly part of the V&V20 document but is implicit in it.

[144] To avoid semantic confusion it is essential that *model* here refers to the continuum model, not including the grid.

(certainly for Certification, and arguably for Validation) is a well-founded rejection of the *poor quality model* enabled by a *high-quality validation exercise*.

In the reverse situation, we could have a high-quality model, with δ_{model} smaller than any forseen application needs (or even a perfect model with $\delta_{model} = 0$), yet the validation exercise could fail because of excessive U_{val} (due to poor computational work, parameter estimation, and/or experiments). Fortunately, in the V&V20 methodology, this does not lead to a false negative evaluation of the model accuracy, but only to the very useful information that well-founded conclusions about model quality cannot be made unless improvements are made, not in the model, but in the validation exercise itself.

This distinction is required for the next section on extending the domain of validation.

11.12 § EXTENDING THE DOMAIN OF VALIDATION[145]

As noted in Section 11.2, consideration of the accuracy of simulation results at points within a domain other than the validation points, i.e. interpolation/extrapolation in a domain of validation, was not considered in V&V20, nor in V&V10 or other such documents. Yet a practical and justifiable position is that useful Validation is not accomplished until the domain of validation is extended away from the experimental validation set points. We refer to a new point, other than a validation set point, at which one intends to perform a new computation, as an **application point**.[146] If an analyst needs information at an application point that coincides with a validation set point in parameter space, modeling is not much needed since he has the experimental values there.[147]

Extending the domain of validation by interpolation/extrapolation involves two components: *how* to interpolate, and *what* to interpolate. The latter is the more subtle issue.

11.12.1 Interpolation Methods

Interpolation is not a glamorous subject, but those who are involved in practical applications acknowledge that it is not trivial, either, especially in parameter spaces of high dimension *P*. The process is simplified if all the parameters are uniformly incremented, producing a *P*-dimensional regular cartesian space. If not, then even linear interpolation involves considerable mathematics. Analysts are free to consider higher order polynomial based interpolations, or physics-based interpolations (e.g. interpolations based on a functional form of an exponential when interpolating over altitude for atmospheric properties, or using the functional form of perturbation solutions from constant-property problems, etc.) or others. Experienced analysts working in difficult problem areas often restrict the process to linear interpolation (or even to zeroth-order) achieving lower formal accuracy in some metric but avoiding behavioral or mimetic errors (Roache, 1998b) such as undershoots (interpolated values exceeding extrema of the original data), entropy violations, mass conservation errors, etc.

---

[145] The material in Section 11.12 is not part of the original V&V20 document, but at the time of this writing it is under more extensive development (including examples) by the ASME V&V 20 Committee, and is planned to be published in one of a series of supplements to V&V20.

[146] A term suggested by Dr. R. Hills of the ASME V&V 20 committee.

[147] A model could still have value at a Validation set point. For example, if we had ten validation set points and the model agreement was good at nine but poor at the application/set point, an analyst might be justified in trusting the model result over the experimental result. Or a model could be used to choose between disparate experimental results at almost the same set point. However, this is a fine distinction. Usually, validation needs to be extended beyond the discrete set points to be useful.

Also, linear functional form is often used for extrapolation, and zeroth-order is the more prudent choice. Although extrapolation has a bad reputation (and some analysts would categorically reject it) it is sometimes required and justifiable. For high parameter space dimension *P*, the distinction between interpolation and extrapolation is not so sharp as in the commonly considered 1-D problems. For *P*-dimensions, interpolation would be limited to the *P*-dimensional convex hull around the data points nearest to the application point, but this can exclude some of the best data from the basis, i.e. an application point nearest to original data points but outside the convex hull. Even in 1-D, extrapolation may be easier to justify than interpolation. For example, two data points at $x = 0$ and 10 would support interpolation to $x = 5$ with a distance to the data points of $|\Delta x| = 5$, but would better support extrapolation to $x = -1$ with a distance of only $|\Delta x| = 1$. The danger is that some parameter transition boundary might exist at $x = 0$ that would restrict to interpolation. This cannot be known by general theories for interpolation/extrapolation but only by physics particular to the problem. But note that such boundaries can also exist interior to the given data points (e.g. x = Mach number = 1 is a dangerous boundary across which to interpolate). Interpolation/extrapolation subprograms can be tailored to special physics and the basic algorithms can be modified by post-processing to avoid or minimize mimetic errors.

The choice of interpolation/extrapolation algorithms is left to the analyst, and we now consider issues particular to the Domain of Validation.

11.12.2 Estimating δ_{model} by Interpolation

We consider what we are given at the **Validation set points**, and what we expect to accomplish in extending the Domain of Validation. Consider the schematic of Figure 11.12.2.1, showing a single parameter M (perhaps Mach number) with a sequence of experimental validations at four Validation set points M = 1,2,3,4 each with f, δ_{model}, and U_{val}. We then perform a new simulation at M = A, an **application point** which is between M = 1 and 2. Clearly we want to have an estimate of δ_{model} and uncertainty U at A. We distinguish known values at A by subscript A, and interpolated values there by subscript Ai.

At the Validation set points M = 1,2,3,4, the analyst has only been given experimental values for the validation variables f and estimates of model error δ_{model} and total Validation Uncertainty U_{val} (or standard uncertainty u_{val}).

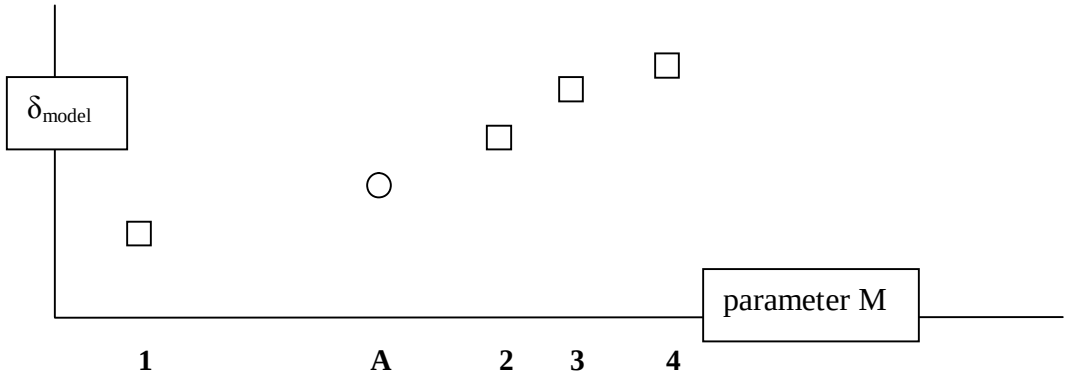


Figure 11.12.2.1. Schematic of Interpolation in the Domain of Validation.

We expect to achieve an estimate of the model error δ_{model} at the application point A. We need an estimate of the error and total uncertainty of our new computation at the application point, denoted as U_A. However, note that U_A cannot be estimated simply by interpolation of U_{val} from the Validation set points data, since these data generally have no relation to whatever grid resolution and algorithm accuracy we might use for our application at A. The quality of the Validation exercises to determine U_{val} will place a *limit* on U_A , but we cannot expect high quality computations at A, where there are no experimental data, to improve the Validation uncertainty.[148]

It does makes sense to interpolate δ_{model}. This is a property of the model, not just of the Validation. (See discussion in Section 11.11.)

It does not make sense to interpolate the solution S. We do not interpolate S, because we calculate a *new* solution at A, denoted S_A.

So we interpolate for $\delta_{model,Ai}$ from the known data. If we use all four known Validation set points 1,2,3,4 and if we use linear or quadratic interpolation, it will be over-determined [149], so we use a least squares fit of the "model of a model" and the standard deviation of the fit (e.g., Eça and Hoekstra, 2006a, 2009b) gives a standard uncertainty term u_{fit}. At each of the four Validation set points, we have been given (from the Validation exercise)

$$\delta_{model} = E \pm U_{val} = S - D \pm U_{val} \tag{11.12.2.1}$$

In the simplest case of approximately independent error terms, this was determined from Eq. (11.5.1), repeated here as

$$\delta_{model} = S - D \pm \{U_{num}^2 + U_{input}^2 + U_D^2\}^{1/2} \tag{11.12.2.2}$$

It does not make sense to interpolate for experimental values D_{Ai} . If we had no modeling, it would now make sense to interpolate for D_{Ai} , but this is replaced by the new computational value S_A and the interpolated values for $\delta_{model,Ai}$.[150]

It does not make sense to interpolate for experimental uncertainty U_D. There is no intermediate experiment. There is generally no relation between U_{D1} and U_{D2} or any others; they even could be associated with different experiments in different facilities. In any case, there is no experiment at A, so there is no associated experimental uncertainty at A.

It does not make sense to interpolate U_{num} . That will be determined by the new simulation at A, which could be at higher or lower numerical accuracy than any of the Validation set points 1,2,3,4. Besides, there is generally no relation between $U_{num,1}$ and $U_{num,2}$ or any others. For the same reasons, it does not make sense to interpolate U_{input} .

---

[148] Improved computations *at* a Validation set point could not be used to extend the Domain of Validation but rather to re-execute and improve the Validation at the same set point. A division of labor is assumed, in which the results of the Validation exercise are handed-off to the analyst. These results consist essentially in estimates of δ_{model} and U_{val} , not details of the intermediate steps, which can be quite elaborate (V&V20) if Eq. (11.5.1) is not applicable, i.e. if the three errors δ_{num} , δ_{input} and δ_D are not effectively independent.

[149] Often the term "interpolation" is restricted to determining a local function that identically passes through the given data. Here, we use it more generally to also refer to "curve fitting" which is not so restricted.

[150] In a meaningful sense of modeling philosophy, we are using a computational model to interpolate between experimental set point values.

11.12.3 Estimating $U_{\delta model}$ by Interpolation

It does make sense to interpolate U_{val} , but the interpretation is aided with a change in notation. Recall that U_{val} is intended to be the uncertainty for the estimate of δ_{model}. So we can use the somewhat clumsy but more suggestive notation $\delta_{model} \pm U_{\delta model}$. We evaluate $U_{\delta model} = U_{val}$ and interpolate for $U_{\delta model,Ai}$.[151]

The new computation result S_A cannot reduce uncertainties at the validation set points 1,2,3,4. It can only *add new uncertainties* at the application point A. The relation is

$$U_{\delta model,A} = \{U_{\delta model,Ai}^2 + U_{fit}^2 + U_{num,A}^2 + U_{input,A}^2\}^{1/2} \qquad (11.12.3.1)$$

where U_{fit} is assumed = 2 u_{fit} for want of any better assumption.

11.12.4 Reporting New Modeling Results

How does the modeler report new results at the application point A ?

(1) If the modeler decides to use the corrected solution[152] S_{CA} at A, with

$$S_{CA} = \{S_A + \delta_{model,Ai}\} \qquad (11.12.3.2)$$

then the reported solution and uncertainty would be

$$S_{CA} \pm U_{\delta model,A} \qquad (11.12.3.3)$$

(2) If the modeler decides to use just the new calculated solution S_A at A, without correcting for the interpolated model error, then the reported solution and uncertainty would be

$$S_A \pm \text{some combination of } \{U_{\delta model,A} \text{ and } |\delta_{model,Ai}|\} \qquad (11.12.3.4a)$$

At the time of this writing, a justified combination has not been determined. A possibly justifiable combination of unlike terms U and δ would use RMS,

$$S_A \pm \{U_{\delta model,A}^2 + \delta_{model,Ai}^2\}^{1/2} \qquad (11.12.3.4b)$$

A more conservative combination would be

$$S_A \pm \{U_{\delta model,A} + |\delta_{model,Ai}|\} \qquad (11.12.3.4c)$$

---

[151] If we do not interpolate for some $U_{\delta model,Ai}$ then the erroneous conclusion could be reached that there is only interpolation error for δ_{model} but no uncertainty at the new calculation point A. We could use a highly accurate numerical solution so $U_{num,A} \sim 0$ and we could perform a very accurate parameter estimation so $U_{input,A} \sim 0$. As already noted, it makes no sense to interpolate U_D . If we used linear interpolation, then $u_{fit,A} = 0$. So our nonsense estimate would be $U_{\delta model,Ai} \sim 0$. Obviously we need to interpolate for $U_{\delta model,Ai}$.

[152] The pros and cons of using S_{CA} or S_C are reminiscent of those for using the solution corrected by Richardson Extrapolation f_c vs using the fine grid solution f_1, discussed in Section 5.6.1.

11.12.5 Division of Responsibilities

The interpolation (non-unique) of δ_{model} and $U_{\delta model}$ could be done continuously (not just at a specific application point) to establish a continuum extended Domain of Validation. It could be done by either the people responsible for the Validation exercises, or by the user addressing a new problem at an application point. If it is the former, they must emphasize that this $U_{\delta model}$ cannot be interpreted as the total uncertainty at a new application point, U_A , but only as a contributor to U_A . The total U_A will be evaluated from Eq. (11.12.3.5) and is the responsibility of the modeler.

In closing, we note that even a very well done Validation exercise leading to small U_{val} at all experimental set points and an accurate model giving small δ_{model} do not absolve the modeler from responsibility for conscientious grid convergence and parametric uncertainty studies at new application points in the Domain of Validation.

PART IV

BROADER ISSUES

Parts II and III of this book covered Verification and Validation, which (to oversimplify) involve mathematics and science, respectively. The broader issues in this Part IV assume Verification and Validation, and involve (to oversimplify again) engineering and management practice.

CHAPTER 12

QUALITY ASSURANCE ISSUES

What the hell is quality? What *is* it?

Zen and the Art of Motorcycle Maintenance: An Inquiry into Values
Robert M. Pirsig, William Morrow Publishing, New York, 1974

12.1 INTRODUCTION

This Chapter covers some general issues involved with formal Quality Assurance activity that may affect the practicing analyst, without getting into details of particular QA programs.[153] The context in this chapter, as in all this book, is that of non-real-time numerical solution of partial differential equations. (See Chapter 2 for definitions and connotations.)

Definitions of "Quality Assurance" are not very informative. The following *description* is adapted from Rechard et al (1991,1992).

QA as related to computer software refers to a formal management system, the goal of which is to ensure that the software consistently does what it is supposed to do to meet the expectations of the user (often referred to as the "customer", i.e., recipient, purchaser, or beneficiary of the work). For the type of projects considered in this book (non-real-time applications), the primary customer expectation is that the software not deceive the user, either by the software producing a meaningless answer or by the developer overstating software capabilities. Hence, QA means that a software user has a reasonable degree of assurance that the software will correctly perform the stated capabilities (e.g., provide a satisfactory solution to the mathematical model) when given the proper input. The degree of assurance must be commensurate to the complexity, importance, and risk of failure of the analysis.

[153] Condensed from Chapter 11 of V&V1.

Beyond this very general description, many factors of code QA could be listed, but will not be pursued here. For examples and further discussion, see Mehta (1996) and Sections 11.6-7 of V&V1. Some terms are obvious and commonplace, others might require elaboration to be legalistically specific, but all are at least suggestive of the elusive term "quality."

12.2 QUALITY ASSURANCE (QA) VS QUALITY WORK

As noted in Chapter 2, Section 2.11 (repeated here), it is worth making the distinction between the technical term "Quality Assurance" or QA vs. "quality work." QA enthusiasts like to talk as though the two were equivalent, but formal QA is largely a system of paperwork (e.g., see ISO, 1991) run by managers. A project can meet all the formal QA requirements and still be low quality (or flatly erroneous). On the other hand, high quality work can and has been done without a modern and formal QA program. QA can, of course, be helpful for quality work. (If nothing else, it can be used to encourage management to support quality work.) Even the formality and paperwork are helpful in issues like version control. But if allowed to run amuck, formal QA can mire real work in forms and procedures and definitions, impeding genuine quality rather than fostering it. For example, formal QA procedures may require expensive re-running of an entire suite of confirmation problems whenever a code is modified; this can discourage minor code fixes and improvements. I have seen QA reviewers complain and require written justification (with committee approvals, documentation of resolution, archiving of correspondence on approved forms, etc.) for a change in input data that produced numerical value changes from 3.0 to 2.999998. And this, in a geophysical problem wherein the parameter ranges were sampled over three orders of magnitude!

12.3 QA VS CREATIVITY

Early and excessive insistence QA issues (e.g., early documentation) will smother creativity. It is analogous to the situation of true mathematicians insisting on rigorous proofs and theorems for every stage of development in an applied mathematics arena. A quote from an old paper by Biot (1956) on the relation between pure and applied mathematics is equally applicable to the relation between QA and creative algorithm and code development.

One could understand the feelings of the artist who undoubtedly benefits from the scientific study of colors but who would be constantly reminded of proceeding with rigorous conformity to the dictates of physics and psychology.

An enlightened QA system should encourage and enable quality programming developed in a flexible, professional, and *progressive* atmosphere, rather than mandate every desirable characteristic in a legalistic fashion.

12.4 QA AND TEMPERAMENT TYPES

It is this smothering by paperwork that creates tension and gives QA a bad name among creative scientists and engineers. There are sound management reasons for QA, yet there are also deeply rooted personality and temperament issues involved as well. In the Myers-Briggs temperament classification system (Myers, 1962; Keirsey and Bates, 1984), the typical QA temperament type is SJ, whereas the creative scientist type is typically NF, diagonally across the matrix from SJ. These two types do not work well together; they get on each others' nerves. Yet they need each others' traits. Big trouble occurs when

one or the other type does not recognize his or her own temperament traits as simply idiosyncrasies, and tries to enforce them on others. QA types will even prefer crude and inaccurate algorithms, just because they are easier to document clearly and to understand. (This statement is *not* an exaggeration on my part. I have experienced strong requests from regulators, unqualified in either the science involved or in the simulation algorithms, to drop sophisticated algorithms, no matter that they are efficient and accurate, in favor of childish and inaccurate spread-sheet type of algorithms which any accountant could understand.)

In spite of these statements, and my own temperament type (which, as the reader may have guessed by now, is NF), I do believe in the usefulness of a formal QA system for large projects, provided that the QA system is designed and applied with some common sense, and that personal power trips are avoided.

12.5 PREVALENCE OF ERRORS IN SCIENTIFIC SOFTWARE: USE OF STATIC ANALYZERS

Hatton (1997) performed a huge suite of tests over 5 years on commercial, government and university scientific software to evaluate the prevalence of errors. The results were horrifying.

12.5.1 The "T" Experiments of Hatton

Hatton's projects were in two categories. The "T1" experiment measured defects without running the codes, by performing static "deep flow analysis" that "looks for inconsistent or undefined use of language." The software analysis programs QAC and QA Fortran were used to gauge "formal consistency," e.g., inconsistency in variable types and/or numbers between code components (functions, subroutines). The T1 experiment covered over 3 million executable lines of scientific software written in C and Fortran, from "many industries and applications areas." The "T2" experiments covered a single applications area and involved running the codes. The results of the T2 experiments were terrible but, I believe, not so representative - the applications area was seismic data processing, a notoriously difficult area involving vary large ill-conditioned matrices and fine-grained input data. On the other hand, I believe the poor results of the T1 experiments to be more generally representative of scientific software.

"More than 100 codes in some 40 applications areas" were tested in T1. No details could be given on the types of codes, although the applications areas included "graphics, nuclear engineering, mechanical engineering, chemical engineering, civil engineering, communications, databases, medical systems, and aerospace," earth science, and control. We may presume that many of these codes involved our interest in this book, numerical solution of partial differential equations. "The age of the codes was evenly spread between 1 and 20 years." All were "mature," i.e. "in regular use by their intended end users." Significantly, "internationally standardized systems of quality control" [presumably ISO 9000 or something like it] "were comprehensively represented."

Once again, it is necessary to introduce semantic distinctions. In Hatton's terminology, the "defects" measured in T1 are called *faults*, while those defects found in T2 by actually running the codes are called *failures*. Failures can arise through a simple defect (i.e., a programming mistake) but *could* also be caused by non-programming defects such as an inappropriate [or just plain incorrect] algorithm, or variations in floating-point implementations. However, Hatton's results indicate that for his T2 tests, in which a common algorithm was used across codes, the only significant source of such inaccuracy *failures* was in fact "simple defects" or *faults*. Hatton noted that "the terms 'defect', 'error', and 'bug' are often used as a generic term for either fault or failure, dependent on context."

Briefly, Experiment T1 showed that "C and Fortran codes, independent of the application area, are riddled with statically detectable faults. For example. calling-sequence inconsistencies (also known as

interface inconsistencies) occurred at the average rate of one in every 7 interfaces in the Fortran codes we tested, and one in every 37 interfaces in C."

What can we do with such numbers?

First or all, Hatton does *not* suggest using C instead of Fortran. As he says, "In C, the delights of pointers add many new ways of getting it all wrong. In C++, even more pitfalls await the unwary..." About half of the higher incidence of interface errors in the Fortran codes is easily normalized away, since the Fortran components on average were about 2.5 times as large and had about 2.5 times as many arguments. I venture the opinion that the higher prevalence of errors in Fortran than in C (and the longer components) results from simple skewing in time; more old code was written in Fortran. (And there is nothing as ugly as really old Fortran code.) Hatton stated that "The net result of changing languages appears to be that the overall defect density appears to be about the same. In other words, when a language corrects one deficiency, it appears to add one of its own."

Second, the presence of these *faults* does not always indicate failures. Most codes with many options were built with the intention of performing a rather limited subset of the options. It is this limited subset that has been given the most attention and exercised. Often, code authors will build in an option more with a view towards future expansion, rather than immediate application, and test it only cursorily. Even with a commercial code, a potential client will usually be advised as to whether the features of interest have been well-exercised or not. Also, some *faults* are simply not errors at execution time, e.g. failure to initiate variables *can* be correct if restricted to computer systems or compiler options that automatically zero all variables. (It is still not good coding practice.) Even inconsistencies in interfaces may not be a true error; I have written correct code which only requires the first few arguments for some function options, and no error results from not including a list of unused arguments in some calls. It could be argued that such a call with a full list of unused variables is less readable. (I do not argue this, anymore, and have added dummy zero's, properly floating point or integer, to the argument list, just to avoid questions and flags from code analyzers.) However, Hatton addressed this in a statistical fashion, asking experts in the code applications areas to weight the severity of the faults, from 5% to 100%. A 5% severity rating represented a low probability that the *fault* would result into a *failure*, "for example, casting a pointer to an integral type in C." A 100% severity weighting meant "effective certainty that failure will take place in a reasonable software life-cycle," but no guarantee. Briefly, their evaluation was that the situation was serious. Some fluids engineering codes happily were on the low end of the weighted fault distributions, while one *nuclear engineering* code was off-scale with an "awe-inspiring 140 weighted static faults per 1,000 lines of code..." [If the mean weighting for a fault was about 50%, this would indicate one coding fault every 4 or so lines of code!] As Hatton says, this package "amounted to no more than a very expensive random number generator."

Third, Hatton's results give evidence that (e.g., see Chapter 3) there will always be a continuing requirement for Confirmation exercises.

12.5.2 Use of Static Analyzers

Fourth, and most important, Hatton's survey indicated the importance of performing static tests on codes, and raises an obvious question. If Hatton and his collaborators could perform these static tests, why did not the code developers do it first?

Good question. One might try to argue that the static analysis tools were not available when the old code was developed. However, this applies only to the oldest Fortran code, since such static analyzers have been commonly available at least since the early 1980's. Then why is not the analyzer applied after the

fact, *now*, to old code? Because (I believe) while it is simple to turn on the analyzer, it is difficult to go through the results, especially for old code. In my limited experience, static analyzers typically present many "false positives," flags that formally *could* be a problem, but are not. The most common (in my experience) has been the flag that states "tests for equality between floating point variables may not be meaningful." Of course, they may *be* meaningful, and are probably not even dangerous when the test is not for simple equality but for "less (greater) than or equal," which construction still elicited flags. (Similar flags occur with some compiler options.) There are other false positive flags, or flags that are true positive but *intended*. For example, one might build a code data structure to accommodate future expansions, including more generality in a key sub-program than is presently required. This can result in error flags of un-initialized variables, "dead" code segments that can never be accessed, etc. In a *good* QA program (which *must* include such static tests) all these flags have to be addressed, either by changing the code (with subsequent headaches of re-running of test cases, new documentation, etc.) or by explaining on QA forms why they are *false* positives, which is perhaps as big a headache, and often leaves a residue of doubt. In the template QA system outlined later, scientific codes at the C level (Candidate for "A," or QA-approved, level) are required to be processed by a static analyzer. However, because these runs typically produce many false positive warnings, and consequently require significant effort and judgment from the Code developer to process the results, its use is not to be *required* at all stages of code development, although early use is recommended (see below).

These static analyzers can also output various indices of the code complexity. For old and ugly Fortran code, the numbers are high, and indicate the terrible state of the art at the time the codes were written. However, I do not believe in arbitrary limits on these complexity indices, nor on the validity of code comparisons based on them. If you want a simple (non-complex) code, you use a simple algorithm. It is a sad but true fact that very efficient algorithms tend to be inherently complex, and that general purpose codes will have many option combinations. I have had QA aficionados suggest dropping a complex but efficient 3-D semi-coarsening multigrid algorithm in favor of simple point relaxation, at a cost of 2–3 orders of magnitude in efficiency for large problems. This is too high a price to pay for the aesthetic of coding simplicity.

The static analyzers can also determine adherence to structured programming and other programming standards, if anyone is interested. Hatton noted that "experiment T1 proved conclusively that attempts to maintain programming standards were risible," i.e. laughable, hilarious. One can only imagine the output of error flags!

In spite of these sad stories of additional work required to use static code analyzers, I believe that

- such static tests should be required in any formal QA program,
- static tests should be used by any code builder or user interested in true quality, and
- static tests should be used early and throughout the lifespan of a code.

It may take a day of drudgery to wade through the analyzer output for a moderate size scientific package, but the discovery of one genuine bug will be relatively painless compared to releasing and using faulty scientific code. As in the case of Verification of mathematical correctness of codes and of calculations, there is a time to put aside creativity, and just think like an accountant for awhile.

12.6 CODE DOCUMENTATION

User acceptance is highly dependent on documentation. In my experience, code documentation is second to no other factor in user acceptance, not even ease of use. (Accuracy is not even in the race, for most users.) The widespread acceptance of the USGS groundwater flow code MODFLOW (McDonald and Harbaugh, 1988) is apparently due to its admirable documentation, as well as its availability. (This is not

to say that it is not accurate or efficient.) Although not flawless or 100% complete, the MODFLOW documentation is definitely methodical and extensive, and is totally devoid of the deliberate obscurantism often seen in scientific journals, using the minimum mathematical sophistication necessary to communicate the ideas to the reader.

The essential user documentation may cover only the input/output data, but full documentation, essential to maintenance, involves internal documentation as well. In each subprogram, internal documentation should describe all the parameters. It would be convenient if the internal documentation was complete and self-contained, but references to other internal or external documentation will be the more usual case. Commercial codes should be paragons of documentation, but unfortunately are often not.

12.7 COMMERCIAL CODES AND THEIR USERS

Most users of commercial codes are primarily concerned with issues other than accuracy. Also, the complexity of the option tree in a general-purpose commercial code makes code accuracy Verification of all option combinations difficult. Even if a code option set is Verified for numerical accuracy, it still remains to Verify the accuracy for a particular calculation, as described in Part II. Some commercial codes include a single grid error "estimator," better referred to as an error indicator; see Chapter 7. As noted therein, dependable error estimation with a practical definition of error (an error really of interest to an engineering calculation) is not yet available. It could be, with an automatic grid coarsening feature, but the typical programming environment is not conducive to inclusion of this feature. Although there are exception and signs of improvement, generally speaking, commercial code accuracy is in a sorry state (not in regard to flexibility or robustness, but accuracy).

The situation is difficult for anyone committed to the QA party line of responsiveness to the "customer." A commercial code user once told me that my concern with numerical accuracy was "Neanderthal," and that the "purpose of calculations is to motivate the cutting of metal." A director of a very successful consulting firm, specializing in simulation of enhanced oil recovery processes, stated that in some 20 years of experience he had *never* had a request from a customer for enhanced numerical accuracy. Even dealing with academic researchers, the experience of myself and colleagues has been disappointing. We have seen more concern over trivial issues of input data structure than numerical accuracy. Not that input data structures and ease of use are not important, but that the customer-users are too *easy* to please-they are happy if these problems are solved, with no concern for numerical accuracy. Commercial code builders cannot be faulted too much for not including dependable accuracy checks if customers are not concerned.

12.8 CODE TO CODE COMPARISONS

Given the suspicion of some regulators, it is inconsistent and surprising that they often like to use Code-to-Code comparisons as part of Code QA or Certification. Certainly, if a new code agrees with an old code that has already been shown to be of high quality, confidence is gained in the new code. In actual practice, such comparisons have often been misused, notably in the groundwater flow community. The practice is acceptable if one code is a trustworthy standard-not merely *old*, or commonly used, but accurate. Also, more limited use of Code-to-Code comparisons is often efficient, e.g. in testing the installation of a new solver, in which case the comparison is on efficiency only, and the Code-to-Code comparison is really used as a debugging device or an aid to confidence building. In general, Code-to-Code comparisons are suspect, and are no substitute for rigorous code accuracy Verification by comparison with exact solutions (see Chapter 3).

12.9 QA FOR LARGE PUBLIC POLICY PROJECTS

QA at its best is a flexible philosophy, at its worst a bureaucratic morass. It seems clear that the need for a comprehensive software QA system increases as the size of an engineering project increases, and as the public becomes more involved, e.g. in pressing environmental issues such as hazardous or nuclear waste disposal. The activity of "stakeholders" who may not be either technically qualified nor intellectually honest puts extreme pressures on QA management, especially in a litigious society such as the United States of America. Regulators may be more competent than the general public, but still technically unqualified, and more intrusive. (It is challenging to try to explain difficult algorithmic issues like conservation in an Approximate Factorization algorithm to people who are unfamiliar with conservation laws or Gaussian elimination.) This political situation leads to excessive rigidity and records-keeping in a QA system, beyond what the technical requirements might be. For anyone involved in such projects, the inevitable results will include reduced personal productivity and increased job stress.

My recommendations are

1. resist micro-management from unqualified regulators who are not supposed to be even macro-managing, and
2. discuss technical questions with members of the general public only if they understand high-school level science and mathematics. (Democracy will not work with an uneducated populace.)

12.10 QA OF ANALYSES

Since computational PDE codes are so extensively used in modern engineering and science projects, it is perhaps worth emphasizing again that having the best code imaginable is not a guarantee of an adequate analysis. The analysis or design of an engineering project involves much more than the use of a (perhaps excellent) set of codes. Therefore, existence of a formal QA system for Codes or Software is not sufficient. If the QA approach is to be effective, there also must be formal QA procedures to review and evaluate the analyses themselves, and probably the resultant reports (e.g., Rechard et al, 1992).

12.11 QA / CERTIFICATION OF USERS AND REGULATORS

Porter (1996) and others have also considered the possibility of Certification not just of codes, but of *code users*. Porter noted that "the utility of CFD in [a particular Air Force project] will be as dependent on the skill and experience of the analysts as on the codes themselves." To me, the most remarkable thing about this statement is that it had to be made at all. It should be patently obvious, but evidently is not.

It is our opinion based on some experience in a major project (WIPP PA Dept., 1992; Helton et al, 1995, 1996) involving sensitive public policy decisions, that *specific user training and Certification* (not just token Certification based on general credentials such as M. S. degree and 5 years of experience) is a much needed interpretation of the QA concept. Furthermore, it should be extended to *include regulators*.

12.12 ASSESSMENT OF CODES? OR USERS?

Strazisar and Denton (1995) presented a "CFD Code Assessment in Turbomachinery–A Progress Report." Intended to be (in the terminology used in this book) a Validation comparison of CFD codes, it became (in my assessment) a QA exercise on user training. The results are worth considering in some detail.

A Working Group in Turbomachinery, set up by the International Gas Turbine Institute, designed the assessment as a true *prediction* exercise, being of the opinion that comparisons with already published data-sets "do not provide an objective view of CFD capability." With the same tactful phrasing, the organizers stated "An objective view is not provided when a code is run by its developer." However, 8 of the 12 choosing to participate used their own codes. The test data-set on a transonic axial flow compressor rotor (what the organizers considered to be "almost the simplest example of a practical machine") was taken at NASA-Lewis. The data set was not distributed (internally or externally) until the 11 "blind" simulations were submitted; the 12th simulation was "non-blind" and performed to test feasibility of the project. After the original submission of results, the participants were free to perform additional postdiction simulations.

Tangential to the main thrust of the exercise, the organizers provided a valuable service to future algorithm/code developers by polling several major turbomachinery designers on their goals for CFD accuracy, shown below.

- Pressure rise ± 1–2%
- Temperature rise ± 1–2%
- Efficiency ± 0.5–1.0%
- Rotor exit flow angle ± 1–2°

However, the designers acknowledged beforehand that these levels were not presently realizable, but were thought to be "ultimately possible." The designers also noted two more encouraging aspects of design simulations.

- "Accurate prediction of *change* between two configurations is more important than prediction of *absolute levels*." [See discussion earlier in Chapter 8 on "Inadequacy of Single Grid Calculations for Parameter Trends" and in Chapter 9 on "Trends, Computational and Experimental."]
- "[A]ccurate prediction of the *shape* of radial distributions is more important than absolute accuracy." [This is an interesting variant on the "trends" question.]

Of the 12 submissions, 10 used the algebraic Baldwin-Lomax turbulence model and 2 used k-ε. Most (9 of 12) used H-type grids [which are known to be difficult to use accurately]. The number of grid points varied from 37,000 to 1,050,000; this factor of nearly 30 is not so suspicious as it may seem at first glance, since this represents only about a factor of 3 in each of the 3 coordinate directions. However, no mention (let alone details) of any Verification of Calculation is made in the article. Thus, it is *not* surprising to read the first conclusion of the organizers; "It has been surprisingly difficult to draw firm conclusions from the numerical predictions," thus echoing the first conclusions of the 1981 Stanford Turbulence "Olympics" held 14 years earlier (see discussion in Chapters 1, 9, 10). No numbers on accuracy are reported, only general assessments. The statement that "overall performance was reasonably well predicted by most methods" [some with postdiction] is difficult to reconcile with another statement, that "overall performance can be *greatly affected* [emphasis added] by the details of the turbulence modeling." Again echoing the 1981 Stanford Turbulence Olympics, "no conclusions can be drawn as to the best [turbulence] model."

The hopeless confusion is further evident in the statements about H-grids. "Unfortunately those using more complex grids always used more grid points and so it is not possible to identify a particular grid type as the source of any improvements." This statement seems to take it for granted that Verification of Calculations was not done adequately. If it were, there would be *no significant effect of grid type on the results*. What it would effect is *efficiency*, with other grid topologies (O-type grids and especially C-type grids) achieving the acceptable level of grid convergence with fewer grid points, if previous general

experience is any guide. All joking aside (see the CFD parody in V&V1, Appendix D), the mathematical solution is not supposed to depend on the coordinate system!

The ultimate confusion (and ultimate justification for the parody) comes from these statements. "Three of the participants used the same method...[and]...code, and it is significant that they all obtained different results both for overall performance and for flow details. This occurred even though two of the simulations used a similar number of grid points. It must be concluded from this that the results are not only code dependent but also user dependent." Many comparison exercises have come to the same (presumably tongue-in-cheek) conclusion. (See Hutton, 2006, cited in Section 10.8.) It simply states the obvious: that different users *who use different code parameters* will obtain different results. This conclusion hardly requires such an extensive comparative exercise.

The authors clearly understood the real issue, when they stated: "It is clear that even tighter controls of the CFD code parameter space are necessary in future test case exercises." To put it another way, QA/Certification of users and Analyses (as well as codes) is required.

12.13 § VALIDATION CLAIMS WITH USER-SPECIFIED INPUT PARAMETERS

The legitimacy of claims of a "Validated code" vs. a Validated model becomes clouded with the presence of user-specified input parameters. As noted previously, strictly speaking, it is a computational model that is validated, rather than a code. But a computational model must be incorporated into a code in order for the model to be exercised and produce results, and it is common and not too misleading to speak of Validation of a code, when one means Validation of the computational model in the code. The trouble is that a code rarely is built to completely specify a computational model; the model form is certainly specified, but the parameters of the model are usually a mix of hard-wired values for some parameters and user-input values for others. (See also Sections 6.31 and 9.18.) The claim of "Validated code" or model can only apply to a stated range of parameters, but the interpretation and connotation is influenced by the accessibility of such parameters. For example, one could have a code with a computational model involving 10 parameters, 4 hard-wired and 6 user-input, and a well documented validation study supporting a claim that the code/model is "Validated." But then a trivial change in coding that switched one of the 4 hard-wired parameters (say the *i-th* parameter x_i) to user input specification would require a new qualification of the claim of "Validated code." The original claim did not need to specify the parameter range for x_i because it was hard-wired, and therefore effectively part of the model form. The new claim would have to specify the acceptable range for x_i ; if no sensitivity study were done, the claimed "range" would have to be restricted to a single number, i.e. the previously hard-wired value of x_i , and QA documentation would need to be changed to reflect this new status.

12.14 OTHER QA ASPECTS

Chapter 11 of V&V1 covers other QA aspects that may be of some interest to code developers, but are not covered here because of primary interest of code users. These aspects include the following: terms, factors and components of QA (see also Mehta, 1996); division of labor, code levels, code sources (see Rizzi and Vos, 1996 and Porter, 1996); desirable but not required code characteristics in QA; the very important subject of code documentation; undocumented options; built-in automatic user error detection; designing for code maintenance; software certification and ISO 9000 standards; and an example of specific QA Procedures and a QA system template adapted from our work in Rechard et al (1991, 1992) on the WIPP project.

12.15 CONCLUDING REMARKS ON QA

Like any bureaucracy, a formal QA system can be frustrating. Like most bureaucracies, it is a necessary evil for large projects. Provided that participants exhibit some patience, tolerance, and common sense, a formal Quality Assurance system can actually contribute to true "quality work," especially for large engineering projects. It can also provide creative scientists, mathematicians, and engineers with a different kind of challenge, and can provide its own sense of satisfaction.

CHAPTER 13

CONCLUSIONS

13.1 THE OVERALL PROCESS FOR QUANTIFICATION OF UNCERTAINTY

This book has treated Verification and Validation, and the general area of Quantification of Uncertainty, in Computational Science and Engineering and related disciplines that utilize the numerical solution of partial differential equations. Included are the activities of Verification of numerical accuracy of Codes, Verification of Calculations, Validation of Codes (or the conceptual and mathematical models on which the code is built), and the broader areas of Code Certification and QA. Certainly, more development work remains to be done, especially in removing burden of Verification of Calculations from Users of commercial codes, and certainly Validation is and will remain an ongoing activity. But there is no excuse for not doing a convincing, credible treatment of the numerics. We *now* have algorithms, codes, methodology, and computer power to do a good job on the numerical accuracy. Note that the accuracy issues associated with "nonlinear dynamics of numerical methods" (see Chapter 8) are easily handled by straightforward techniques, and the "basins of attractions" phenomena relate only to code iteration convergence and robustness. While certainly important to Code Users, these are not *accuracy* issues.

Techniques are presently available to generate an exact solution (the Method of Manufactured Solutions, Chapter 3) for virtually any PDE code including commercial codes. This method (or another equally complete method) should be used first to rigorously Verify the numerical accuracy of a Code, before Verifying the individual calculations. The Verified code order (say 2nd order) can then be used to verify an actual calculation over 2 grids, or, preferably, the observed order for an actual calculation can be determined experimentally by calculations over a minimum of three grids.

Also, good practice requires that one investigate changing the time step even for steady state and for adaptive time step methods if nonlinear time stepping algorithms like 4th order nonlinear Runge-Kutta have been used.

Once the mathematical veracity of the code and the calculation have been established (i.e., Verification), then one can proceed to attempt Validation of the code (or better, the conceptual model embodied in the code) by comparison to experiments. The Validation is ad hoc, particular to the physical experiment and a (usually small, but hopefully usable) parameter range around the particular experiment. The voices of extensive experience clearly warn the practitioner (or team): do not attempt the Validation process until you have confidence in both Verifications.

The various approaches to Verification of Calculations can be compared on the basis of their reliability, flexibility, and work required. Single grid error estimators are perhaps most efficient, and are appealing because they do not require additional grid generation, but only if the analyst assumes the convergence rate to be known. Observed convergence rates can only be calculated with solutions on multiple grids. Also, single grid error estimators do require additional code development and numerical experimentation to correlate (calibrate) their predictions with more straightforward grid convergence studies. This work must be repeated (re- calibrated) for any change in the mathematical model, which becomes a serious issue in code maintenance. Nevertheless, these methods are attractive for large suites of calculations such as those required in Monte Carlo studies and optimization studies. This re-calibration of the error estimator is not required with grid convergence studies and error estimation (or banding) based on Richardson Extrapolation. Also, generalized Richardson Extrapolation-based methods such as the Grid Convergence Index are the most reliable, are the most quantifiable, are specific to whatever practical error measure is of interest, and are adaptable to various order of accuracy; they can be used to Verify and extract the observed order of convergence for a calculation. Methods using separate PDEs to transport the errors require the most work, and are fragile; they are interesting and powerful but not quite ready for general recommendation.

The convincing Verification of a Calculation sometimes involves subtleties, but generally does not involve difficult concepts or mathematics. It is somewhat tedious and requires attention to detail. Melnik's advice is realistic: "Think like an accountant." I would add, "Go do your homework."

The level of confidence required depends on the project, i.e. on the intended use of the results. As we noted in the 1986 ASME Journal of Fluids Engineering Policy Statement (see Chapter 1), many industrial calculations will probably continue to be single grid calculations without an error estimate, with adequacy of the grid resolution estimated by intuition and experience. But the "practical engineer" must recognize that computational PDE practitioners with an extensive experience base usually cannot infer grid resolution requirements from ostensibly nearby calculations. This is a task for the code user, not the code builder. It is just as difficult when a commercial code is used, no matter what the advertising promises may imply. Furthermore, although merely predicting trends produced by changes in parameters will generally require less grid resolution than predicting absolute values, the user cannot be casual; under-resolved calculations cannot be expected to always produce correct trends, even if the solutions are qualitatively correct. There exist counter-examples showing that grossly or even moderately under-resolved solutions do not show the correct trends in some quantities of interest. Also, such trends or sensitivity studies may require tighter iteration convergence criteria than absolute values do.

Compared to just obtaining a single grid solution, it requires (very) roughly twice as much work to Verify a new calculation, if we are using a previously Verified code and the same parameter range, and if the Verification is positive on the first try. If the new problem exhibits new features (compared to the previously Verified problem), perhaps three times the effort will be required

13.2 FULFILLING THE PROMISE OF COMPUTATIONAL POWER

With the often noted tremendous increases in computer speed and memory, and with the less often acknowledged but equally powerful increases in algorithmic accuracy and efficiency, a natural question suggests itself. What are we doing with the new computer power? with the new GUI and other set-up advances? with the new algorithms? What *should* we do? Shall we continue to produce more pseudo-solutions, or shall we finally solve the problems correctly? I suggest that, when all things are considered, including the inevitable fruits of quick-and-dirty calculations - anxieties and guesswork and misdirections and distrust and tarnished reputations, both for individuals and for the computational community in general–that it is simpler and easier to "just do it!" *Get the right answer*.

The statement of Ferziger (previously quoted in Chapter 3) bears repeating.

"*... the frequently heard argument 'any solution is better than none' can be dangerous in the extreme. The greatest disaster one can encounter in computation is not instability or lack of convergence but results that are simultaneously good enough to be believable but bad enough to cause trouble."*

References and Bibliography

Abdel-Salam, A. and Chrysikopoulos, C. V. (1994), "Analytical Solutions for One-Dimensional Colloid Transport in Saturated Fractures," *Advances in Water Resources*, Vol. 17, 1994, pp. 283–396.

Adams, E. E. and Gelhar, L. W. (1992), "Field Study of Dispersion in a Heterogeneous Aquifer 2. Spatial Moments Analysis," *Water Resources Research*, Vol. 28, 1992, pp. 3293–3307.

AEA Technology (1994), *CFDS-Flow3D User Guide*, Computational Fluid Dynamics Services, Building 8.19, Harwell Laboratory, Oxfordshire OX11 ORA, United Kingdom.

Aeschliman, D. P., Oberkampf, W. L., and Blottner, F. G. (1995), "A Proposed Methodology for Computational Fluid Dynamics Code Verification, Calibration, and Validation," *Proceedings 16th International Congress on Instrumentation in Aerospace Simulation Facilities*, WPAFB, Ohio, 95–CH3482–7, July 1995, pp. 27.1–27.13.18–21. See also Aeschliman and Oberkampf (1997a,b).

Aeschliman, D. P. and Oberkampf, W. L. (1997), "Experimental Methodology for Computational Fluid Dynamics Code Validation," SAND95–1189, Sandia National Laboratories, Albuquerque, New Mexico, September 1997.

Aeschliman, D. P. and Oberkampf, W. L. (1998), "Experimental Methodology for Computational Fluid Dynamics Code Validation," *AIAA Journal*, Vol. 36, No. 5, 1998, pp. 733-741.

AGARD (1979), "Experimental Data Base for Computer Program Assessment," NATO Advisory Group for Aeronautical Research and Development, Fluid Dynamics Working Group 04, AGARD AR–138, May 1979.

AGARD (1985), "Test Cases for Inviscid Flow Field Methods," NATO Advisory Group for Aeronautical Research and Development, Fluid Dynamics Working Group 07, AGARD AR–211, May 1985.

AGARD (1988), *Validation of Computational Fluid Dynamics*, Lisbon, Portugal, 2–5 May 1988, NATO Advisory Group for Aeronautical Research and Development, AGARD CP 437, Volume 1: Symposium Papers and Round Table Discussion. Volume 2: Poster Papers. December 1988.

AGARD (1994), *Assessment of Wind Tunnel Data Uncertainty*, NATO Advisory Group for Aeronautical Research and Development, Advisory Report AGARD–AR–304, 1994.

Aharoni, A. (1995), "Agreement Between Theory and Experiment," *Physics Today*, June 1995, pp. 33–37.

AIAA (1994), "Editorial Policy Statement on Numerical Accuracy and Experimental Uncertainty," *AIAA Journal*, Vol. 32, No. 1, January 1994, p. 3.

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

AIAA (2006), Editorial Policy Statement, January 2006, all AIAA Journals.

AIAA Guide, see AIAA (1998)

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

Allen, J. S. and Cheng, S. I. (1970), "Numerical Solution s of the Compressible Navier-Stokes Equations for the Laminar Near Wake," *Physics of Fluids*, Vol. 13, No. 1, January 1970, pp. 37–52.

American Nuclear Society (1987), *Guidelines for the Verification and Validation of Scientific and Engineering Computer Programs for the Nuclear Industry*, ANSI/ANS–10.4–1987, American Nuclear Society, 555 North Kensington Avenue, La Grange Park, Illinois 60525, USA.

Anderson, W. K. and Batina, J. T. (1988), "Accurate Solutions, Parameter Studies, and Comparisons for the Euler and Potential Flow Equations," in AGARD (1988), Paper No. 14.

Anon. (2006), Proc. Workshop on the Future of Modeling and Simulation for Combustion Applications, Pittsburgh, PA, Feb. 21-23, 2006.

Anon. (2008), Proc. Verification and Validation for Nuclear Systems Analysis Workshop, Idaho National Laboratory**,** Idaho Falls, Idaho, 21-25 July 2008.

Appel, J. R. and Gunzburger, M. D. (1997), "Difficulties in Sensitivity Calculations for Flows with Discontinuities," *AIAA Journal*, Vol. 35, No. 5, May 1997, pp. 842–848.

Armaly, B. F., Durst, F., Pereira, J. C. F., and Schonung, B. (1983), "Experimental and Theoretical Investigation of Backward-Facing Step Flow," *Journal of Fluid Mechanics*, Vol. 127, p. 473.

Ashford, O. M. (1985), *Prophet or Professor? The Life and Work of Lewis Fry Richardson*, A. Hilger, Boston.

Ashford, O. M. et al (1993), Eds., *The Collected Papers of Lewis Fry Richardson*, (2 Volumes), Cambridge University Press, Cambridge.

ASME (1984), ANSI/ASME *Measurement Uncertainty for Fluid Flow in Closed Conduits*, MFC–2M–1983, ASME 1984, American National Standards Institute/American Society of Mechanical Engineers, New York.

ASME (1986), ANSI/ASME *Measurement Uncertainty*, PTC 19.1–1985, ASME 1986, American National Standards Institute/American Society of Mechanical Engineers, New York.

ASME (1993a), *ASME Journal of Fluids Engineering*, Vol. 115, September 1993, pp. 339–340.

ASME (1993b), *ASME Journal of Heat Transfer*. Vol. 115, February 1993, pp. 5–6.

ASME (1994), Editorial Board, "Journal of Heat Transfer Editorial Policy Statement on Numerical Accuracy," *ASME Journal of Heat Transfer*, Vol. 116, November 1994, pp. 797–798.

ASME (2002), "The Role of Risk Analysis in Decision Making," ASME Board of Governors Position Paper ID #02-15, April 2002. www.asme.org/gric/ps/2002/02-15.html.

ASME (2006), ANSI/ASME PTC 19.1-2005, *Test Uncertainty*. ASME, New York, NY.

ASME Committee PTC-60 (2006), *ASME V&V 10-2006. Guide on Verification and Validation in Computational Solid Mechanics*, 29 December 2006.

ASME Committee PTC-61 (2009), *ANSI Standard V&V 20, Guide on Verification and Validation in Computational Fluid Dynamics and Heat Transfer*.

ASCE/EWRI (2009), see Wang et al. (2009)

Avva, R. K., Kline, S. J., and Ferziger, J. H. (1988), "Computation of Turbulent Flow over a Backward-Facing Step-Zonal Approach," AIAA 26th Aerospace Sciences Meeting, January 11-14, 1988/Reno, Nevada.

Babuska, I. , Strouboulis, T. and Upadhyay, C. S. (1994), "A Model Study of the Quality of A Posteriori Error Estimators for Linear Elliptic Problems. Error Estimation in the Interior of Patchwise Uniform Grid of Triangles," *Computer Methods in Applied Mechanics and Engineering*, Vol. 114, 1994, pp. 307–378.

Babuska, I. , Strouboulis, T. Gangaraj, S. K. and Upadhyay, C. S. (1997), "Pollution Error in the *h*-Version of the Finite Element Method and the Local Quality of the Recovered Derivatives," *Computer Methods in Applied Mechanics and Engineering*, Vol. 140, 1997, pp. 1–37.

Babuska, I. and Strouboulis, T. (2001), *The Finite Element Method and its Reliability*, Oxford University Press, Oxford, UK.

Bachalo, W. D. and Johnson, D. A. (1979), "An Investigation of Transonic Turbulent Boundary Layer Separation Generated on an Axisymmetric Flow Model," AIAA Paper 79–1479, July 1979.

Baptista, A. M. (1987), *Solution of Advection-Dominated Transport by Eulerian-Lagrangian Methods using the Backward Methods of Characteristics*, Ph.D. Thesis, Dept. of Civil Engineering, Massachusetts Institute of Technology, 1987.

Barber, T. J. (1996), "The Role of Code Validation and Certification in the Design Environment," AIAA Paper 96–2033, 27th AIAA Fluid Dynamics Conference, New Orleans, Louisiana, 17–20 June 1996.

Barton, I. E. (1995), "A Numerical Study of Flow over a Confined Backward-Facing Step," *International Journal for Numerical Methods in Fluids*, Vol. 21, pp. 653–665.

Beam, R. and Warming, R. F. (1976), "An Implicit Finite Difference Algorithm for Hyperbolic Systems in Conservation Law Form," *Journal of Computational Physics*, Volume 22, 1977, pp. 87–110.

Bear, J. (1972), *Dynamics of Fluids in Porous Media*, American Elsevier, New York, 1972.

Benek, J. A., Kraft, E. M., and Lauer, R. F. (1996), "Validation Issues for Engine/Airframe Integration," AIAA Paper 96-2031, 27th AIAA Fluid Dynamics Conference, New Orleans, Louisiana, June 17–20, 1996.

Bennett, J. R. and Campbell, J. E. (1987), "Accuracy of a Finite-Difference Method for Computing Lake Currents," *Journal of Computational Physics*, Vol. 68, 1987, pp. 262–271.

Bergstrom, J. and Gebart, R. (1997), "Estimation of Numerical Accuracy for the 3D Flow Field in a Hydraulic Turbine Draft Tube," *Computers and Fluids*, to appear.

Bingham, F. W. (1990), *Software Quality Assurance Plan*, SLTR 89–2001, Nuclear Waste Repository Technology Department, Sandia National Laboratories, Albuquerque, New Mexico.

Biot, M. A. (1956), "Applied Mathematics: And Art and a Science," *Journal of the Aeronautical Sciences*, May 1956, pp. 406–410.

Blackwell, B. and Dowding, K. (2006), "Sensitivity Analysis and Uncertainty Propagation of Computational Models", Chapter 14 of *Handbook of Numerical Heat Transfer*, Second Edition, W J. Minkowycz, E. M. Sparrow, and J. Y. Murthy, eds., Wiley, New York.

Blackwell, B., Dowding, K., and Modest, M. (2009), "Cylindrical Geometry Verification Problem for Enclosure Radiation", *AIAA Journal*, to appear.

Blottner, F. G., "Finite Difference Methods of Solution of the Boundary Layer Equations," *AIAA Journal*, Vol. 8, No. 2, 1970, pp. 193–199; see also, "Finite-Difference Methods for Solving the Boundary Layer Equations with Second-Order Accuracy," *Proceedings of the Second International Conference on Numerical Methods in Fluid Dynamics*, Lecture Notes in Physics, Vol. 8, edited by M. Holt, Springer-Verlag, New York, 1971.

Blain, C. A., J. J. Westerink, and R. A. Luettich, Jr. (1994), "The Influence of Domain Size on the Response Characteristics of a Hurricane Storm Surge Model," *Journal of Geophysical Research*, Vol. 99, C9, 1994, pp. 18467–18479.

Blottner, F. G., and Roache, P. J. (1971), "Nonuniform Mesh Systems," *Journal of Computational Physics*, Vol. 8, No. 3, 1971, pp. 498–499.

Blottner, F. G. (1977), "Introduction to the Proceedings of the 3rd AIAA Computational Fluid Dynamics Conference," *Proceeding of the 3rd AIAA Computational Fluid Dynamics Conference*, Albuquerque, New Mexico, June 27–28, 1977, pp. 1–2.

Blottner, F. G. (1989), "Accurate Navier-Stokes Results for the Hypersonic Flow over a Spherical Nosetip," AIAA Paper 89–071, June 1989.

Blottner, F. G. (1990), "Accurate Navier-Stokes Results for the Hypersonic Flow over a Spherical Nosetip," *Journal of Spacecraft and Rockets*, Vol. 27, No. 2, March–April 1990, pp. 113–122. See also AIAA Paper 89–071, June 1989.

Bobbitt, P. J. (1988), "The Pros and Cons of Code Validation," AIAA Paper 88–2535. Also, NASA TM 100657, July 1988.

Boehm, B. W. (1981), *Software Engineering Economics*, Prentice-Hall, 1981.

Bogdonoff, S. M. (1993), "Computational Validation for Hypersonics: An Experimentalist's Perspective," *Workshop on Hypersonic Flows for Reentry Problems*, Antibes, France, January 1993.

Boggs, J. M., Young, et al (1992), "Field Study of Dispersion in a Heterogeneous Aquifer. 1. Overview and Site Description," *Water Resources Research*, Vol. 28, 1992, pp. 3281–3291.

Boggs, J. M. and Adams, E. E. (1992), "Field Study of Dispersion in a Heterogeneous Aquifer. 4. Investigation of Adsorption and Sampling Bias," *Water Resources Research*, Vol. 28, 1992, pp. 3325–3336.

Bond, R. b., Ober, C. C., Knupp, P. M., and Bova, S. W. (2007), "Manufactured Solution for Computation Fluid Dynamics Boundary Condition Verification," *AIAA Journal*, Vol. 45, No. 9, September 2007, pp. 2224-2236.

Book, D. L., Boris, J. P., and Hain, K. (1975), "Flux-Corrected Transport II: Generalizations of the Method," *Journal of Computational Physics*, Vol. 18, 1975, pp. 248–283. See also Boris, J. P., Naval Research Laboratory, NRL – MR–2357, November 1971.

Botella, O. and Peyret, R. (1998), "Benchmark Spectral Results on the Lid-Driven Cavity Flow," *Computers and Fluids*, Vol. 27, No. 4, pp. 421-433.

Boulding, K. E. (1978), *Ecodynamics: A New Theory of Societal Evolution*, Sage Publications Ltd. London.

Bradley, R. G. (1988), "CFD Validation Philosophy," in AGARD (1988).

Bradshaw, P. (1975), "Computers vs. Wind Tunnels," in Letters, *Astronautics and Aeronautics*, September 1975, p. 6.

Bradshaw, P. (1992), "Final Report on AFOSR 90–0154, Collaborative Testing of Turbulence Models," Mechanical Engineering Department, Stanford University, Stanford, California.

Brandt, A. (1977), "Multi-Level Adaptive Solutions to Boundary-Value Problems," *Mathematics of Computation*, Vol. 31, 1977, pp. 333–390.

Bras, R. L. and Rodriguez-Iturbe, I. (1985), *Random Functions and Hydrology*, Addison Wesley, New York.

Briley, W. R. and McDonald, H. (1977), "An Implicit Numerical Method for Multi-Dimensional Compressible Navier-Stokes Equations," *Journal of Computational Physics*, Vol. 24, No. 4, 1977, pp. 372–397.

Briley, W. R., and McDonald, H. (1980), "On the Structure and Use of Linearized Block Implicit Schemes," *Journal of Computational Physics*, Vol. 34, No. 19, pp. 54–73.

Brinster, K. (1995), personal communication.

Brock, J. S. (2005), "ASC Verification Project Bibliography, December 2005", LA-UR-05-9250, Los Alamos National Laboratories, NM, December 2005.

Brown, D. L. and Minion, M. L. (1995), "Performance of Under-Resolved Two-Dimensional Incompressible Flow Simulations, *Journal of Computational Physics*, Vol. 122, 1995, pp. 165–183.

Bruneau, C. H. and Saad, M. (2006), "The 2D Lid-Driven Cavity Problem Revisited," *Computers and Fluids*, Vol. 35, pp. 326-348

Burggraf, O. R. (1966), "Analytical and Numerical Studies of the Structure of Steady Separated Flows," *Journal of Fluid Mechanics*, Vol. 24, Part 1, 1966, pp. 113–152.

Burton, K., Viceconti, M., Olsen, B., Nolte, L.-P. (2005), "Extracting clinically relevant data from finite element simulations," Editorial, *Clinical Biomechanics*, Vol. 20, p. 451-454.

Cadafalch, J., Pérez-Segarra, C. C., Cónsul, R. and Oliva, A. (2002), "Verification of Finite Volume Computations on Steady State Fluid Flow and Heat Transfer", *ASME Journal of Fluids Engineering*, Vol. 124, pp. 11-21, March 2002.

Cadafalch, J. (2002), personal communication.

Cantekin, M. E. and Westerink, J. J. (1990), "Non-Diffusive $N + 2$ Degree Petrov-Galerkin Methods for Two-Dimensional Transient Transport Computations," *International Journal for Numerical Methods in Engineering*, Vol. 11, 1990, pp. 397– 418.

Carey, G. F. (2006), "Computational Geometry, Grid Generation, and Adaptive Grids, Chapter 15 of *Handbook of Numerical Heat Transfer*, Second Edition, W J. Minkowycz, E. M. Sparrow, and J. Y. Murthy, eds., Wiley, New York.

Carrica, P. M., Wilson, R.V., and Stern, F. (2007), "An Unsteady Single-phase Level

Set Method for Viscous Free Surface Flows," *International Journal of Numerical Methods in Fluids*, Vol. 53, No. 2, pp. 229-256.

Caruso, S. C., Ferziger, J. H., and Oliger, J. (1985), "Adaptive Grid Techniques for Elliptic Flow Problems," Report No. TF–23, Thermosciences Division, Stanford University, Stanford, California, 1985.

Caswell, H. (1988), "Theory and models in ecology: a different perspective," *Ecological Modeling*, Vol. 43, pp. 33-44.

Celik, I. and Freitas, C. J. (1990), Eds., *Benchmark Test Cases for Computational Fluid Dynamics*, ASME FED Vol. 93, Book No. H00598–1990.

Celik, I. (1993), "Numerical Uncertainty in Fluid Flow Calculations: Needs for Future Research, *ASME Journal of Fluids Engineering*, Vol. 115, June 1993, pp. 194–195.

Celik, I. and Zhang, W-M. (1993), "Application of Richardson Extrapolation to Some Simple Turbulent Flow Calculations," in Celik et al (1993), pp. 29–38.

Celik, I., Chen, C. J., Roache, P. J., and Scheuerer, G. (1993), Eds., *Symposium on Quantification of Uncertainty in Computational Fluid Dynamics*, FED Vol. 158, ASME Fluids Engineering Division Summer Meeting, Washington, D. C., June 20–24, 1993.

Celik, I. and Karatekin, O. (1995), "Numerical Experiments on Application of Richardson Extrapolation to a Turbulent Flow with Separation," in Johnson and Hughes (1995), pp. 37–43. See also *ASME Journal of Fluids Engineering*, Vol. 119, No. 3, September 1997, pp. 584–590.

Celik, I. and Zhang, W-M. (1995), "Calculation of Numerical Uncertainty Using Richardson Extrapolation: Application to Some Simple Turbulent Flow Calculations," *ASME Journal of Fluids Engineering*, Vol. 117, September 1995, pp. 439–445.

Celik, I. and Hu, G. (2003), "Further Refinement and Benchmarking of a Single-Grid Error Estimation Technique", AIAA-2003-0628, *41st AIAA Aerospace Sciences Meeting*, Reno, NV, 6-9 January 2003.

Celik, I and Li, J. (2005), "Assessment of numerical uncertainty for the calculations of turbulent flow over a backward facing step," *International Journal for Numerical Methods in Fluids*, Vol. 49, pp. 1015-1031.

Celik, I., Li, J., Hu, G., and Shaffer, C. (2005), "Limitations of Richardson Extrapolation and Some Possible Remedies," *ASME Journal of Fluids Engineering*, Vol. 127, pp. 795-805.

Celik, I., Klein, M. and Janicka, J. (2006), "Assessment Measures for LES Applications," *Proc. 2006 ASME Joint U.S. European Fluids Engineering Summer Meeting*, July 17-20, Miami Florida.

Celik, I. B., Ghia, U., Roache, P. J., Freitas, C. J., Coleman, H. W., and Raad, P. E. (2008), "Procedure for Estimation and Reporting of Uncertainty due to Discretization in CFD Applications", *ASME Journal of Fluids Engineering*, Editorial.

Chang, S. and Haworth, D. C. (1995), "Kinetic-Energy-Balance Based Solution-Adaptive Mesh Refinement," in Johnson and Hughes (1995), pp. 7–12.

Chang, S. and Haworth, D. C. (1997), "Adaptive Grid Refinement Using Cell-Level and Global Imbalances," *International Journal for Numerical Methods in Fluids*, Vol. 24, pp. 375–392.

Chapman, D. R., Mark, H., and Pirtle, M. W. (1975), "Computers vs. Wind Tunnels for Aerodynamic Flow Simulations," *Aerospace America*, April 1975, pp. 22–35.

Cheng, S. I. (1970), "Numerical Integration of the Navier–Stokes Equations, *AIAA Journal*, Vol. 8, No. 12, December 1970, pp. 2115–2122.

Claus, R. W. and Vanka, S. P. (1992), "Multigrid Calculations of a Jet in Crossflow," *Journal of Propulsion Power*, Vol. 8, pp. 185–193, 1992.

Coakley, T. J. and Huang, P. G. (1992), "Turbulence Modeling for High Speed Flows," AIAA Paper 92–0436, January 1992.

Coleman, H. W. and Steele, W. G. (1995), "Engineering Application of Experimental Uncertainty Analysis," *AIAA Journal*, Vol. 33, No. 10, October 1995, p. 1888.

Coleman, H. W., Steele, W. G., and Taylor, R. P. (1995), "Implications of Correlated Bias Uncertainties in Single and Comparative Tests," *ASME Journal of Fluids* Engineering, Vol. 117, December 1995, pp. 552–556.

Coleman, H. W. (1996), "Uncertainty Considerations in Validating CFD Codes with Experimental Data," AIAA Paper 96–2027, 27th AIAA Fluid Dynamics , New Orleans, Louisiana Conference, June 17–20, 1996.

Coleman, H. W., and Stern, F. (1997), "Uncertainties and CFD Code Validation", *ASME Journal of Fluids* Engineering, Vol. 119, December 1997, pp. 795–803.

Coleman, H. W., and Stern, F. (1998), "Reply by the Authors" *ASME Journal of Fluids* Engineering, Vol. 120, September 1998, pp. 636-637.

Coleman, H. W., Stern, F., Di Mascio, A., and Campana, E. (2001), "The Problem with Oscillatory Behavior in Grid Convergence Studies," *ASME Journal of Fluids* Engineering, Vol. 123, p. 438.

Coleman, H. W. (2003), "Some Observations on Uncertainties and the Verification and Validation of Simulations," *ASME Journal of Fluids Engineering*, Vol. 125, No. 4, p. 733-735, July 2003.

Coleman, H. W. and Steele, W. G. (2009), *Experimentation and Uncertainty Analysis for Engineers*. Third Edition. Wiley, New York, NY.

Comini, G. Cortella, G. and Manzan, M. (1995), "A Streamfunction-Vorticity-Based Finite-Element Formulation for Laminar-Convection Problems," *Numerical Heat Transfer, Part B*, Vol. 28, pp. 1–22.

Condie, K. G, McCreery, G. E., McIlroy, H. M. Jr., and McEligot, D. M., (2005), "Development of an Experiment for Measuring Flow Phenomena Occurring in a Lower Plenum for VHTR CFD Assessment," INL/EXT-05-00603, 21 September 2005.

Conte, S. D. and de Boor, C. (1965), *Elementary Numerical Analysis: An Algorithmic Approach*, McGraw-Hill, New York, 1965, p. 313.

Coorevits, P., Ladeveze, P., and Pelle, J.-P. (1995), "An Automatic Procedure with a Control of Accuracy for Finite Element Analysis in 2D Elasticity," *Computer Methods in Applied Mechanics and Engineering*, Vol. 121, 1995, pp. 91–120.

Cosner, R. R. (1994), "Issues in Aerospace Application of CFD Analysis," AIAA Paper 94–0464, AIAA 32nd Aerospace Sciences Meeting, Reno, Nevada, January 10–13, 1994.

Cosner, R. R. (1995), "CFD Validation Requirement for Technology Transition," AIAA Paper 95–2227, 26th AIAA Fluid Dynamics Conference, San Diego, California, June 19–22, 1995.

Cowen, R. (2008), "Standard model gets the right answer,", *Science News*, 20 December 2008, p. 13.

Davies, G. B. and Johnston, C. D. (1984), "Comment on **'**Contaminant Transport in Fractured Porous Media: Analytical Solutions for a System of Parallel Fractures'," *Water Resources Research*, Vol. 20, No. 9, September 1984, pp. 1321–1322.

Davis, R. T. (1975), "Introduction to the Proceedings of the 2nd AIAA Computational Fluid Dynamics Conference," *Proceedings of the 2nd AIAA Computational Fluid Dynamics Conference*, Hartford, Connecticut, June 19–20, 1975, p. 1.

Davis, P. J. (1996), "Weather, War, and Mathematics", Book Review, *SIAM News*, November 1996, p. 4.

de Marsily, G., Combes, P., and Goblet, P. (1992), "Comment on 'Ground-Water Models Cannot be Validated," *Advances in Water Resources*, Vol. 15, 1992, pp. 367–369.

de Neuman, B. (1989), Ed., *Software Certification*, Elsevier Applied Science Publishers Ltd., Barking, United Kingdom.

Department of Defense (1996), *DoD Modeling and Simulation (M&S) Verification, Validation, and Accreditation (VV&A)*, DoD Instruction 5000. 61, April 29, 1996. Re-issued 13 May 2003.

de Vahl Davis, G., Jones, I. P., and Roache, P. J. (1979), "Natural Convection in an Enclosed Cavity: A Comparison Problem," *Computers and Fluids*, Vol. 7, 1979, pp. 315–316.

de Vahl Davis, G. and Jones, I. P. (1983), "Natural Convection in a Square Cavity: A Comparison Exercise," *International Journal for Numerical Methods in Fluids*, Vol. 3, No. 3, pp. 227–248.

de Vahl Davis, G. (1983), "Natural Convection of Air in a Square Cavity: A Bench Mark Numerical Solution," *International Journal for Numerical Methods in Fluids*, Vol. 3, No. 3, 1983, pp. 249–264.

de Vahl Davis, G. (1994a), "Evaluation of Benchmark Problems ," in Muller, A. (1994), ed., *Proceedings of the Basel World CFD User Days 1994, Second World Conference in Applied Computational Fluid Dynamics*, Basel, Switzerland, 1–5 May 1994.

de Vahl Davis, G. (1994b), "Going for Quality in CFD Codes," *Proceedings 2nd World Conference in Applied Computational Fluid Dynamics*, World User Association–CFD, Basel, Switzerland, 1–5 May 1994, pp. 18.1–18.6.

Devenport, W., Powell, K and Power, G (1996), "Fluid Dynamics," *Aerospace America*, December 1996, pp. 16–17.

Dietrich, D. E., Roache, P. J. and Marietta, M. G. (1990), "Convergence Studies with the Sandia Ocean Modeling System," *International Journal for Numerical Methods in Engineering*, Vol. 11, 1990, pp. 127–150.

Dietrich, D. E., and Roache, P. J. (1991), "An Accurate Low Dissipation Model of the Gulf of Mexico Circulation," *Proceedings International Symposium on Environmental Hydraulics*, 14–16 December 1991, University of Hong Kong, J. H. W. Lee, et al, eds., H. Balkeema, Amsterdam.

Dolling, D. S. (1996), "Considerations in the Comparison of Experimental Data with Simulations–Consistency of Math Models and Flow Physics," AIAA Paper 96–2030, 27th AIAA Fluid Dynamics Conference, New Orleans, Louisiana, June 17–20, 1996.

Driver, D. M. and Seegmiller, H. L., 1985, "Features of a Reattaching Turbulent Shear Layer in Divergent Channel Flow," *AIAA Journal*, Vol. 23, No. 2, pp. 163-171. ERCOFTAC Classic Database C-30.

Driver, D. M. (1991), "Reynolds Shear Stress Measurements in a Separated Boundary Layer," AIAA Paper 91–1787, June 1991.

Dudebout, R. and Fahs, S. (1996), "Evaluation of Workstation Performance by Running a CFD Benchmark Code," CFD Society of Canada Bulletin, No. 7, Spring 1996, pp. 1–5.

Durlofsky, L. J. (1991), "Numerical Calculation of Equivalent Grid Block Permeability Tensors for Heterogeneous Porous Media," *Water Resources Research*, Vol. 27, 1991, pp. 699–708.

Durr, S. Fodor, Z, et al (2008), "Ab Initio Determination of Light Hadron Masses," *Science*, Vol. 322, No. 5905, 21 November 2008, pp. 1224-1227.

Durst, F., and Pereira, J. C. F., "Time-Dependent, Backward Facing Step Flow," *ASME Journal of Fluid Engineering*, Vol. 110, No. 3, 1988, pp. 289–296.

Dwyer, H. A., Kee, R., and Sanders, B. (1980), "Adaptive Grid Method for Problems in Fluid Mechanics and Heat Transfer," *AIAA Journal*, Vol. 18, pp. 1205–1212.

Eça, L. and Hoekstra, M. (2000a), "An Evaluation of Verification Procedures for Computational Fluid Dynamics," IST Report D72-7, Instituto Superior Tecnico (Lisbon), June 2000.

Eça, L. and Hoekstra, M. (2000b), "On the Application of Verification Procedures in Computational Fluid Dynamics," *2nd MARNET Workshop*, MARIN Maritime Research Institute Netherlands, 2000.

Eça, L. and Hoekstra, M. (2002a), "Verification Procedures for Computational Fluid Dynamics On Trial," IST Report D72-14, Instituto Superior Tecnico (Lisbon), July 2002.

Eça, L. and Hoekstra, M. (2002b), "An Evaluation of Verification Procedures for CFD Algorithms," *Proc. 24th Symposium on Naval Hydrodynamics*, Fukuoka, Japan, 8-13 July 2002.

Eça, L., Vaz, G. B., Falcao de Campos, J. A. C., and Hoekstra, M., (2004), *AIAA Journal*, Vol. 42, No. 12, December 2004, p. 2401.

Eça, L. and Hoekstra, M., eds. (2004), *Proc. Workshop on CFD Uncertainty Analysis*, 21-22 October 2004. Instituto Superior Técnico, Lisbon, Portugal.

Eça, L., Hoekstra, M., and Roache, P. J. (2005), "Verification of Calculations: an Overview of the Lisbon Workshop," AIAA Paper No. 4728, AIAA Computational Fluid Dynamics Conference, Toronto, June 2005.

Eça, L. and M. Hoekstra, M. (2006a), "On the Influence of the Iterative Error in the Numerical Uncertainty of Ship Viscous Flow Calculations," Proc. 26th Symposium on Naval Hydrodynamics, Rome, Italy, 17-22 Sept. 2006.

Eça, L. and Hoekstra, M. eds. (2006b), *Proc. Second Workshop on CFD Uncertainty Analysis*, 19-20 October 2006. Instituto Superior Técnico, Lisbon, Portugal.

Eça, L. and M. Hoekstra, M. (2007a), Evaluation of Numerical Error Estimation Based on Grid Refinement Studies with the Method of Manufactured Solutions," Report D72-42, MARIN, may 2007.

Eça, L. and M. Hoekstra, M. (2007b), "Code Verification of Unsteady Flow Solvers with the Method of Manufactured Solutions," Paper No. ISOPE-2007-565, International Society of Offshore and Polar Engineers.

Eça, L., Hoekstra, M., and Roache, P. J. (2007a), "Verification of Calculations: an Overview of the 2nd Lisbon Workshop," AIAA Paper No. 2007-4089, 2nd Workshop on CFD Uncertainty Analysis, AIAA Computational Fluid Dynamics Conference, Miami, June 2007.

Eça, L., M. Hoekstra, M., Hay, A., and Pelletier, D. (2007b), "A Manufactured Solution for a Two-Dimensional Steady Wall-Bounded Incompressible Turbulent Flow," *International Journal of Computational Fluid Dynamics*, Vol. 21, March-May 2007, pp. 175-188.

Eça, L., M. Hoekstra, M., Hay, A., and Pelletier, D. (2007c), "On the Construction of Manufactured Solutions for One and Two-Equation Eddy-Viscosity Models," *International Journal for Numerical Methods in Fluids*, Vol. 54, 2007, pp. 119-154.

Eça, L. and Hoekstra, M. eds. (2008a), *Proc. Third Workshop on CFD Uncertainty Analysis*, 23-24 October 2008, Instituto Superior Técnico, Lisbon, Portugal.

Eça, L. and Hoekstra, M. (2008b), "Testing Uncertainty Estimation and Validation Procedures in the Flow Around a Backward Facing Step," in Eça and Hoekstra (2008), *Proc. Third Workshop on CFD Uncertainty Analysis*, 23-24 October 2008, Instituto Superior Técnico, Lisbon, Portugal.

Eça, L. and Hoekstra, M. (2008c), "Overall Comments on the Workshop," in Eça and Hoekstra (2008), *Proc. Third Workshop on CFD Uncertainty Analysis*, 23-24 October 2008, Instituto Superior Técnico, Lisbon, Portugal.

Eça, L., Hoekstra, M., Roache, P. J. and Coleman, H. (2009), "Code Verification, Solution Verification and Validation: an Overview of the 3rd Lisbon Workshop," AIAA Paper No. 2009-3647, 19th AIAA Computational Fluid Dynamics, San Antonio, Texas, June 2009.

Eça, L., and Hoekstra, M. (2009a), "On the numerical accuracy of the prediction of resistance coefficients in ship stern flow calculations," *Journal of Maritime Science and Technology*, DOI 10.1007/s00773-008-0003-8.

Eça, L., and Hoekstra, M. (2009b), "Evaluation of numerical error estimation based on grid refinement studies with the method of manufactured solutions," *Computers and Fluids*, doi:10.1016/j.compfluid.2009.01.003. See also Eça and Hoekstra (2007a).

Eça, L. (2009), personal communication 12 March 2009.

Eça, L. (2009b), personal communication 22 May 2009.

Ekaterinaris, J. A. and Menter, F. R. (1994), "Computation of Separated and Unsteady Flows with One and Two-Equation Turbulence Models," AIAA Paper 94–0190, January 1994.

Ekaterinaris, J. A. (1995), "Numerical Investigation of Dynamic Stall of an Oscillating Wing, *AIAA Journal*, Vol. 33, No. 10, October 1995, pp. 1803–1808.

Eklund, D. R., Fletcher, D. G., Hartfield, R. J., Northam, G. B., and Dancey, C. L. (1995), "A Comparative Computational/Experimental Investigation of Mach 2 Flow over a Rearward-Facing Step," *Computers and Fluids*, Vol. 24, No. 5, pp. 593–608.

Elizalde-Blancas, F., Karaismail, E., and Celik, I. B. (2008), "Application of Error Scaling Method in Conjunction with GCI," *Proc. 3rd Workshop on CFD Uncertainty Analysis*, Lisbon, 23-24 October 2008, Instituto Superior Técnico, Lisbon. L. Eça, and M. Hoekstra, eds.

Elsenaar, A., Hjelmberg, L., Butefisch, K., and Bannink, W. J. (1988), "The International Vortex Flow Experiment" in AGARD (1988), Paper No. 9.

Emmons, H. (1970), "Critique of Numerical Modeling of Fluid–Mechanics Phenomena," *Annual Review of Fluid Mechanics*, Vol. 2, 1970, pp. 15–36.

Emmons, H. W. (Chairman), (1981),"Evaluation Committee Report," *Proceedings of the 1980–81 AFOSR–HTTM–Stanford Conference on Complex Turbulent Flows, Taxonomies, Reporter's Summaries, Evaluation, and Conclusions, Vol.* 11, Thermosciences Division, Mechanical Engineering Department, Stanford University, Stanford, CA, 1981, pp. 979–986.

Englund, E. and Sparcs, A. (1991), *Geostatistical Environmental Software User's Guide*, U. S. EPA 600/8–91/008.

Engquist, B. and Sjogreen, B. (1996), "Numerical Convergence Rate in the Presence of Shocks," to appear. (See Yee and Sweby, 1996.)

Ethier, C. R. and Steinman, D. A. (1994), "Exact Fully 3-D Navier-Stokes Solutions for Benchmarking," *International Journal for Numerical Methods in Fluids*, Vol. 19, 1994, pp. 369–375.

Eurachem/CITAC (2007), "Guide: Quantifying Uncertainty in Analytical Measurement,", Section 8.1. http://www.measurementuncertainty.org/mu/guide

Ewing, R. E. (1990), "A Posteriori Error Estimation," *Computer Methods in Applied Mechanics and Engineering*, Vol. 82, pp. 59–72.

Fasel, H (1990), "Numerical Simulation of Instability and Transition in Boundary Layer Flows," in D. Arnal and R. Michel (Eds.), *Laminar-Turbulent Transition*, Springer-Verlag, Berlin.

Ferziger, J. H. (1981), *Numerical Methods for Engineering Application*, John Wiley & Sons, New York.

Ferziger, J. H. (1988), "A Note on Numerical Accuracy," *International Journal for Numerical Methods in Fluids*, Vol. 8, pp. 995–996.

Ferziger, J. H. (1993), "Estimation and Reduction of Numerical Error," *ASME-FED Vol. 158, Symposium on Quantification of Uncertainty in Computational Fluid Dynamics*, ASME Fluids Engineering Division, Summer Meeting, Washington, D.C., June 20–24, 1993, Celik., I, Chen, C. J., Roache, P. J., and Scheurer, G., Eds., pp. 1-8.

Ferziger, J. H. and Peric, M. (1996), "Further Discussion of Numerical Errors in CFD," *International Journal for Numerical Methods in Fluids*, Vol. 23, pp. 1263–1274.

Firmin, M. C. P., and McDonald, M. A. (1988), "The Design of the GARTEUR Low Aspect-Ratio Wing for Use in the Validation of Shear Layer and Overall Flow Prediction Methods," Paper P5 in AGARD (1988), Vol. 2.

Fishman, G. S. and Kiviat, P. J. (1968), "The statistics of discrete-event simulation," *Simulation*, April 1968, pp. 185-195.

Fong, J. T. and de Wit, R., (Eds.), (2004), *Proc. Workshop on Verification and Validation of Computer Models for Design and Performance Evaluation of High-Consequence Engineering Systems*, U.S. National Institute of Standards and Technology, Gaithersburg, MD, 8-9 Nov. 2004.

Fong, J. T., Filliben, J. J., Fields, R. J., deWit, R., and Bernstein, B. (2004), "A Reference Benchmark Approach to V&V of Computer Models of High-Consequence Engineering Systems," Chapter 4, Preprints of 41 Invited Papers of Fong and de Wit (2004).

Freitas, C. J. (1993a), "Editorial," *Journal of Fluids Engineering*, Vol. 115, No. 3, September 1993, pp. 339–340.

Freitas, C. J. (1993b), "Editorial Policy Statement on the Control of Numerical Accuracy," *ASME Journal of Fluids Engineering*, Vol. 115, No. 3, September 1993, pp. 339–340.

Freitas, C. J. (1993c), Ed., *The CFD Triathlon: Three Laminar Flow Simulations by Commercial CFD Codes*, FED Vol. 160, ASME Fluids Engineering Conference, Washington, DC, June 20–24, 1993.

Freitas, C. J. (1995a), "The End of Justification, the Beginning of Implementation," *ASME Journal of Fluids Engineering*, Vol. 117, No. 1, March 1995, p. 9.

Freitas, C. J. (1995b), "Perspective: Selected Benchmarks for Commercial CFD Codes," *ASME Journal of Fluids Engineering*, Vol. 117, June 1995, pp. 208–218.

Freitas, C. J., U. Ghia, I. Celik, H. Coleman, P. Raad, and P. J. Roache (2003), "ASME'S Quest to Quantify Numerical Uncertainty," AIAA Paper 2003-0627, AIAA 41st Aerospace Sciences Meeting, January 2003, Reno, Nevada.

Freitas, C. J. (2008), personal communication.

Freitas, C. J. (2009), personal communication, 13 May 2009.

Fujii, K., Gavali, S., and Holst, T. L., "Evaluation of Navier-Stokes and Euler Solutions for Leading-Edge Separation Vortices," *International Journal for Numerical Methods in Fluids*, Vol. 8, 1988, pp. 1319–1329.

Ghia, U., Ghia, K. N., and Shin, C. T. (1982), "High-*Re* Solutions for Incompressible Flows Using the Navier-Stokes Equations and a Multigrid Method," *Journal of Computational Physics*, Vol. 48, 1982, pp. 387–411.

Ghia, K. N., Osswald, G. A., and Ghia, U. (1989), "Analysis of Incompressible Massively Separated Viscous Flows using Unsteady Navier-Stokes Equations," *International Journal for Numerical Methods in Fluids*, Vol. 9, p. 1025.

Ghia, U. (2008), personal communication.

Golub, G. H. and Ortega, J. M. (1992), *Scientific Computing and Differential Equations*, Academic Press, San Diego.

Gourlay, R. and Morris, J. Ll. (1968), "Deferred Approach to the Limit in Non-Linear Hyperbolic Systems," *Computing Journal*, Vol. 2, 1968, pp. 95–101.

Grasso, F., Leone, G., and Delery, J. M. (1994), "Validation Procedure for the Analysis of Shock–Wave/Boundary–Layer Interaction Problems," *AIAA Journal*, Vol. 32, No. 9, September 1994, pp. 1820–1827.

Gray, W. (1993), Ed., Special Issue devoted to "Research Perspectives in Hydrology," *Advances in Water Resources*, Vol. 17, 1994.

Grenda, J. M., Schwer, D. A. and Merkle, C. L. (1996), "Use of Analytical Solutions in Validating Unsteady CFD Codes," AIAA Paper No. 96–2164, June 1996.

Grenier, R. R., Luettich, R. A. and Westerink, J. J. (1995), "A Comparison of the Nonlinear Frictional Characteristics of Two-Dimensional and Three-Dimensional Models of a Shallow Tidal Embayment," *Journal of Geophysical Research*, Vol. 100, No. C7, pp. 13719–13735.

Gresho, P. M. and Lee, R. L. (1981), "Don't Suppress the Wiggles–They're Telling You Something," *Computers and Fluids*, Vol. 9, No. 2, 1981, pp. 223–255.

Gresho, P. M. and Taylor, C. (1994), "Editorial," *International Journal for Numerical Methods in Fluids*, Vol. 19, 1994, p. iii.

Grismer, M. J. and Powers, J. M. (1992), "Comparison of Numerical Oblique Detonation Solutions with an Asymptotic Benchmark," *AIAA Journal*, Vol. 30, No. 12, December 1992, pp. 2985–2987.

Grismer, M. J. and Powers, J. M. (1996), "Numerical Predictions of Oblique Detonation Stability Boundaries," *Shock Waves*, Vol. 6, pp. 147–156.

Hagen, S. C. (1997), "Finite Element Grids Based on a Localized Truncation Error Analysis," Ph.D. Dissertation, Department of Civil Engineering and Geological Sciences, University of Notre Dame, September 1997.

Hagen, S. C., Westerink, J. J. and Kolar, R. L. (1997), "Finite Element Grids based on a Localized Truncation Error Analysis," *International Journal for Numerical Methods in Fluids*, in revision.

Haitjema, H. M. (1995), *Analytic Element Modeling of Groundwater Flow*, Academic Press, New York.

Hall, D. J. and Zingg, D. W. (1995), "Viscous Airfoil Computations Using Adaptive Grid Redistribution," *AIAA Journal*, Vol. 33, No. 7, July 1995, pp. 1205–1210.

Hamming, R. W. (1962), *Numerical Methods for Scientists and Engineers*, McGraw-Hill Book Company, New York. "Chapter N + 1. The Art of Computing for Scientists and Engineers."

Harlow, F. H. and Fromm, J. E. (1965), "Computer Experiments in Fluid Dynamics," *Scientific American*, Vol. 212, No. 3, pp. 104–110.

Harlow, F. H. and Nakayama, P. I. (1967), *Physics of Fluids*, Vol. 10, 1967, p. 2323.

Hatton, L. (1997), "The T Experiments: Errors in Scientific Software," *Computational Science and Engineering*, Vol. 4, No. 2, April–June 1997, pp. 27–38.

Haworth, D. C., El Tahry, S. H., Huebler, M. S., and Chang, S. (1990), "Multidimensional Port-and-Cylinder Flow Calculations for Two- and Four-Valve-Per-Cylinder Engines: Influence of Intake Configuration on Flow Structure," SAE Paper No. 900257.

Haworth, D. C. (1993), personal communication.

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

Hay, A. and Pelletier, D. (2007), "Code and Solution Verification of an Adaptive Finite Element Turbulent Flow Solver," AIAA Paper 2007-4202, 18th AIAA Computational Fluid Dynamics Conference, Miami, Florida, June 2007.

Hay, A. and Pelletier, D. (2008), "Verification and Validation of Turbulent Flow Calculation using an Adaptive Finite Element Solver", *Proc. 3rd Workshop on CFD Uncertainty Analysis*, Lisbon, 23-24 October 2008, Instituto Superior Técnico, Lisbon. L. Eça, and M. Hoekstra, eds.

Haynes, T. S., Reed, H. L., and Saric, W. S. (1996), "CFD Validation Issues in Transition Modeling," AIAA Paper 96–2051, 27 AIAA Fluid Dynamics Conference, New Orleans, Louisiana, 17–20 June 1996.

Hazelrigg, G. A. (2003), "Thoughts on Model Validation for Engineering Design," DETC2003/DTM-48632, Proceedings of DETC.03, ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference,

Chicago, Illinois, USA, September 2-6, 2003.

Helton, J. C., Anderson, D. R.,... Roache, P. J., et al (1995), "Effect of Alternative Conceptual Models in a Preliminary Performance Assessment for the Waste Isolation Pilot Plant," *Nuclear Engineering and Design*, Vol. 154, 1995, pp. 251–344.

Helton, J. C., Anderson, D. R.,... Roache, P. J., et al (1996), "Uncertainty and Sensitivity Analysis Results Obtained in the 1992 Performance Assessment for the Waste Isolation Pilot Plant," *Reliability Engineering and System Safety*, Vol. 51, 1996, pp. 53–100.

Hemez, F. and Brock, J. (2008), "Verification of a Lagrangian Hydrodynamics Code," LA-UR-08-4197, Los Alamos National Laboratory, Los Alamos, NM, 2008.

Hemez, F. (2008), personal communication.

Hemsch, M. J. (2002a), "Statistical Analysis of CFD Solutions from the Drag Prediction Workshop," AIAA-2002-0842, *40th AIAA Aerospace Sciences Meeting*, Reno, NV, January 2002. See also *Journal of Aircraft*, Vol. 41, pp. 95-103, 2004.

Hemsch, M. J. (2002b), personal communication.

Hemsch, M. and Morrison, J. H. (2004), "Statistical Analysis of CFD Solutions from 2nd Drag Prediction Workshop," pp. 4951-4981, 42nd AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, 2004.

Hetu, J.-F. and Pelletier, D. H. (1992), "Adaptive Remeshing for Viscous Incompressible Flows," *AIAA Journal*, Vol. 30, 1992, pp. 1986–1992.

Hills, R. G. (2008), personal communication.

Hinkleman, M. (1995), Private Communication cited as Ref. 18 in Benek et al (1996).

Hoekstra, M. and Eça, L. (1999), "An Example of Error Quantification of Ship-Related CFD Results," MARIN Maritime Research Institute Netherlands, *7th Numerical Ship Hydrodynamics Conference*, Nantes, France, July 1999.

Hoekstra, M. and Eça, L. (2000a), "An Example of Error Quantification of Ship-Related CFD Results," MARIN Maritime Research Institute Netherlands, 2000.

Hoekstra, M., Eça, L., Windt, J. and Raven, H. (2000b), "Viscous Flow Calculations for KVLCC2 and KCS Models using the PARNASSOS Code," Proc. Gothenburg 2000, *A Workshop on Numerical Ship Hydrodynamics*, Gothenburg, Sweden.

Holst, T. L. (1987), "Viscous Transonic Airfoil Workshop Compendium of Results," AIAA Paper 87–1460, 1987. See also *Journal of Aircraft*, Volume 25, No. 12, December 1988, pp. 1073–1087.

Horgan, J. (1995), "From Complexity to Perplexity," *Scientific American*, Vol. 272, No. 6, June 1995, pp. 104–109.

Hortmann, M., Peric, M. and Scheurer, G. (1990), "Multigrid Benchmark Solutions for Laminar Natural Convection Flows in Square Cavities," ASME FED Vol. 93, *Benchmark Test Cases for Computational Fluid Dynamics*, I. Celik and C. J. Freitas, Eds., Book No. H00598, June 1990, pp. 1–6.

Huang, H. and Li, M. (1997), "Finite-Difference Approximation for the Velocity-Vorticity Formulation on Staggered and Non-Staggered Grids," *Computers and Fluids*, Vol. 26, No. 1, 1997, pp. 59–82.

Hugger, J. (1997), "The Equilibrium Method for Local A Posteriori, Pointwise Estimation of the Finite Element Error," *Computer Methods in Applied Mechanics and Engineering*, Vol. 145, Nos. 3–4, 1997, pp. 263–283.

Hugo, R. J. (1995), "Quantifying the Spatio-Temporal Effects of Optically-Active Turbulent Flowfields on a Coherent Optical Wave," Ph.D. Dissertation, Department of Aerospace and Mechanical Engineering, University of Notre Dame, 1995.

Hugo, R. J., Jumper, E. J., Havener, G, and Stepanek, S. A. (1995), "Time-Resolved, Aero-Optical Measurements of a Wavefront Aberrated by a Compressible Free Shear Layer," AIAA Paper 95–1979, AIAA 26th Plasmadynamics and Lasers Conference, San Diego, California, June 19–22, 1995.

Hull, M. (1997), The Creationism Debate: Behind the Ignorance that Confuses Religion with Science," *Crosswinds*, Albuquerque, New Mexico, U. S. A, March 1997, pp. 6–7.

Hunt, J. C. R. (1998), "Lewis Fry Richardson and His Contributions to Mathematics, Meteorology, and Models of Conflict", *Annual Reviews of Fluid Mechanics*, Volume 30, 1998, pp. xii–xxxvi.

Hutchings, B. and Iannuzzelli, R. (1987), "Benchmark Problems for Fluid Dynamics Codes," *Mechanical Engineering*, Vol. 109, No. 6, June 1987, pp. 54–58.

Hutton, A. G. (2006), "Quality and Reliability in CFD," Plenary Presentation, FEDSM2006, ASME Joint U.S. European Fluids Engineering Summer Meeting, 17-20 July 2006, Miami, Florida, U.S.A.

Hyman, J., Shashkov, M. and Steinberg, S. (1997), "The Numerical Solution of Diffusion Problems in Strongly Heterogeneous Non-Isotropic Materials," *Journal of Computational Physics*, Vol. 132, p. 130–148.

IAHR (1982), "Proceedings IAHR Workshop on Expanding Channel Problem," Rome, Italy, 24 June 1982.

Ignat, L., Pelletier, D. and Ilinca, F. (1998), "Adaptive Computations of Turbulent Forced Convection," *Numerical Heat Transfer, Part A*, Vol. 34, pp. 847-871, 1998.

Ilinca, A., Camarero, R., Trepanier, J. Y., and Reggio, M. (1995), "Error Estimator and Adaptive Moving Grids for Finite Volumes Schemes," *AIAA Journal*, Vol. 33, No. 11, November 1995, pp. 2058–2065.

Ilinca, F., D. Pelletier, D. and F. Arnoux-Guisse, F. (1997a), "An Adaptive Finite Element Scheme for Turbulent Free Shear Flows," *International Journal of Computational Fluid Dynamics*, Vol. 8, pp. 171-188, 1997.

Ilinca, F., Pelletier, D. and Garon, A. (1997b), "An Adaptive Finite Element Method for a Two-Equation Turbulence Model in Wall-Bounded Flows," *International Journal for Numerical Methods in Fluids*, Vol. 24, 1997, pp. 101–120.

Imel, G. and Mousseau, K. eds. (2010), Special Issue on Verification and Validation, *Nuclear Science and Engineering*, Spring 2010.

ISO (1991), *Quality Management and Quality Assurance Standards–Part 3: Guidelines for the Application of ISO 9001 to the Development, Supply, and Maintenance of Software*, First Edition, International Organization for Standardization, ISO 9000–3, Geneva, Switzerland, 1991, reprinted 1993.

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

ITTC (2002), *ITTC - Quality Manual*, "CFD General Uncertainty Analysis in CFD Verification and Validation Methodology and Procedures," International Towing Tank Conference. http://ittc.sname.org/2002_recomm_ proc/7.5-03-01-01.pdf

Jameson, A. and Martinelli, L. (1996), "Mesh Refinement and Modeling Errors in Flow Simulation," AIAA Paper 96–2050, 27th AIAA Fluid Dynamics Conference, New Orleans, Louisiana, June 17–20, 1996.

Jankowski, D. (1985), personal communication.

Jay, F. (1984), Ed., *IEEE Standard Dictionary of Electrical and Electronic Terms*, Institute of Electrical and Electronic Engineers, ANSI/IEEE Std. 100–1984.

Johns, B., Dube, S. K., Mohanty, U. C. and Rao, A. D. (1983a), "Simulation of Storm Surges Using a Three-Dimensional Numerical Model: An Application to the 1977 Andhra Cyclone," *Quarterly Journal of Research of the Meteorology Society*, Vol. 109, 1983, pp. 211–224.

Johns, B., Dube, S. K., Mohanty, U. C. and Rao, A. D. (1983b), "On the Effect of Bathymetry in Numerical Storm Surge Simulation Experiments," *Computers and Fluids*, Vol. 11, 1983, pp. 161–174.

Johnson, D. A. and King, L. S. (1985), "A Mathematically Simple Turbulence Closure Model for Attached and Separated Turbulent Boundary Layers," *AIAA Journal*, Vol. 23, No. 11, November 1985, pp. 1684–1692.

Johnson, D. A. and Coakley, T. J. (1990), "Improvements to a Nonequilibrium Algebraic Turbulence Model," *AIAA Journal*, Vol. 28, No. 11, November 1990, pp. 2000–2003.

Johnson, C., Rannacher, R., and Boman, M. (1995), "Numerics and Hydrodynamic Stability: Toward Error Control in Computational Fluid Dynamics," *SIAM Journal on Numerical Analysis*, Vol. 32, 1995, pp. 1058–1079.

Johnson, R. W. and Hughes, E. D. (1995), eds., *Quantification of Uncertainty in Computational Fluids Dynamics–1995*, Joint JSME–ASME Fluid Mechanics Meeting, ASME FED Vol. 213, Hilton Head, South Carolina, August 14–18, 1995.

Johnson, L. T. (1996), *The Real Jesus: The Misguided Quest for the Historical Jesus and the Truth of the Traditional Gospels*, Harper, San Francisco, p. 172.

Joslin, R. D. and Street, C. L. (1992), "Crossflow Disturbance Evolution on Swept Wings Computed by Spatial DNS," *Bulletin of the American Physical Society*, Volume 37, p. 1778.

Jumper, E. J., Hugo, R. J., and Cicchiello, J. M. (1994), "Turbulent-Fluid-Induced Optical Wavefront Dynamics: Near-and Far-Field Implications," AIAA Paper 94–2547, AIAA 18th Aerospace Ground Test Conference, Colorado Springs, Colorado, June 20–23, 1994.

Jumper, E. J. and Hugo, R. J. (1995), "Quantification of Aero-Optical Phase Distortion Using the Small-Aperture Beam Technique," *AIAA Journal*, Vol. 33, No. 11, November 1995, pp. 2151–2157.

Kacker, R. and Jones, A. (2003), "On use of Bayesian statistics to make the *Guide to the Expression of Uncertainty in Measurement* consistent," *Metrologia*, Vol. 40, pp. 235-248.

Kamm, J. R. (2002), personal communication.

Kamm, J. R., Rider, W. J., and Brock, J. S. (2003), "Combined Space and Time Convergence Analyses of a Compressible Flow Algorithm," AIAA Paper 2003-4241.

Karniadakis, G. E. (1995), "Toward a Numerical Error Bar in CFD," *ASME Journal of Fluids Engineering*, Vol. 117, No. 1, March 1995, pp. 7–9.

Kasser, J. L. (2006), *Philosophy of Science*, Part 3, Lectures 30-32, Video Course and Course Guidebook, The Teaching Company, Chantilly, VA.

Keirsey, D. and Bates, M. (1984), *Please Understand Me: Character and Temperament Types*, Prometheus Book Company, Del Mar, California.

Kessler, R., Peric, M. and Scheuerer, G. (1988), "Solution Error Estimation in the Numerical Predictions of Turbulent Recirculating Flows," in AGARD (1988), pp. P9–1 to 12.

Kipp, K. L., Jr. (1987), "HST3D: A Computer Code for Simulation of Heat and Solute Transport in Three-Dimensional Ground-Water Flow Systems," U. S. Geological Survey, Water Investigations Report 86–4095, USGS, Denver, Colorado.

Kleb, B., and Wood, W. (2004), "CFD: A Castle in the Sand?", AIAA Paper 2004-2627, AIAA 34th Fluid Dynamics Conference, Portland, OR, June 28-July 1, 2004.

Kleb, B., and Wood, W. (2006), "Computational Simulations and the Scientific Method,' *AIAA Journal of Aerospace Computing, Information, and Communication*, Vol. 3, No. 6, June 2006, pp. 244-250.

Kline, S. J. and McClinstock, F. A. (1953), "Describing Uncertainties in Simple-Sample Experiments," *Mechanical Engineering*, Jan. 1953, pp. 3–8.

Kline, S. J., Cantwell, B. J., and Lilley, G. M. (1981a), Eds., *The 1980–81 AFOSR/HTTM–Stanford Conference on Complex Turbulent Flows*, 3 Volumes, ThermoSciences Division, Mechanical Engineering Department, Stanford University, Stanford, California.

Kline, S. J., Cantwell, B. J., and Lilley, G. M. (1981b), eds., *Objectives, Evaluation of Data, Specifications of Test Cases, Discussion, and Position Papers. The 1980–81 AFOSR/HTTM-Stanford Conference on Complex Turbulent Flows*, Vol. 1, ThermoSciences Division, Mechanical Engineering Department, Stanford University, Stanford, California.

Kline, S. J. (1985), "The Purposes of Uncertainty Analysis," *ASME Journal of Fluids Engineering*, Vol. 107, pp. 153–164.

Knapp, C. F. and Roache, P. J. (1968), "A Combined Visual and Hot-Wire Anemometer Investigation of Boundary Layer Transition," *AIAA Journal*, Vol. 6, No. 1, January 1968, pp. 29–36.

Knupp, P.M. and Steinberg, S. L. (1993), *Fundamentals of Grid Generation*, CRC Press, Boca Raton, Florida.

Knupp, P. and Luczak, R. (1995), "Truncation Error in Grid Generation: A Case Study," *Numerical Methods for Partial Differential Equations*, Vol. 11, pp. 561–571, 1995.

Knupp, P. (1996), "A Moving Grid Algorithm for Three-Dimensional Regional Groundwater Flow Calculations with Water Table and Seepage Face," *Advances in Water Resources*, Vol. 19, No. 2, 1996, pp. 83–95.

Knupp, P., Rucker, D. and Corbet, T. F. (1996), "Groundwater Modeling with the Phreatic Boundary Condition: Is It Necessary?," Ecodynamics Research Associates, Albuquerque, New Mexico, December 1996.

Knupp, P. M. and Salari, K. (2003), *Verification of Computer Codes in Computational Science and Engineering*, CRC Press, Boca Raton, FL.

Konikow, L. F. and Bredehoeft, J. D. (1992), "Groundwater Models Cannot be Validated," *Advances in Water Resources*, Vol. 15, 1992, pp. 75–83.

Kordulla, W., Schwamborn, D. and Sobieczky, H. (1988), "The DFVLR-F5 Wing Experiment," in AGARD (1988), Paper No. 13.

Kreiss, H.-O. and Lundqvist, E. (1968), "On Difference Approximations with Wrong Boundary Values," *Mathematics of Computation*, Volume 22, 1968, pp. 1–12.

Kuhn, T. (1962), *The Structure of Scientific Revolutions*, 2nd edition, enlarged, 1970; 3rd edition, 1996. University of Chicago Press, Chicago.

Kuruvila, G., and Anderson, J. D., "A Study of the Effects of Numerical Dissipation on the Calculation of Supersonic Separated Flows," AIAA Paper 85–0301, Jan. 1985.

Laflin, K. R., et al (2005), "Data Summary from the Second AIAA Computational Fluid Dynamics Drag Prediction Workshop," *Journal of Aircraft*, Vol. 42, pp. 1165-1178, 2005.

Lafon, A. and Yee, H. C. (1992), "On the Numerical Treatment of Nonlinear Source Terms in Reaction-Convection Equations," AIAA Paper 92–0419, 30th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 6–9 January 1992.

Lardner, R. W. and Song, Y. (1992), "A Comparison of Spatial Grids for Numerical Modelling of Flows in Near-Coastal Seas," *International Journal for Numerical Methods in Engineering*, Vol. 14, 1992, pp. 109–114.

Larsson, A. (1992), "The International Projects INTRACOIN, HYDROCOIN and INTRAVAL," *Advances in Water Resources*, Vol. 15, 1995, pp. 85–87.

Lax, P. (1957), "Hyperbolic Systems of Conservation Laws II," *Communications on Pure and Applied Mathematics*, Vol. 10, pp. 537–566.

Lee, D. and Yeh, C. L. (1994a), "A Hybrid Adaptive Gridding Procedure for Three-Dimensional Flow Problems," *Computers and Fluids*, Vol. 23, 1994, pp. 39–53.

Lee, D. and Yeh, C. L. (1994b), "Computation of Turbulent Recirculating Flows Using a Hybrid Adaptive Grid," *Numerical Heat Transfer, Part A*, Vol. 26, 1994, pp. 415–430.

Leijnse, A. and Hassanizadeh, S. M. (1994), "Model Definition and Model Validation," *Advances in Water Resources*, Vol. 17, 1994, pp. 197–200.

Leonard, B. P. (1991), "The ULTIMATE Conservative Difference Scheme Applied to Unsteady One-Dimensional Advection," *Computational Methods in Applied Mechanics and Engineering*, Vol. 88, 1991, pp. 17–74.

Leonard, B. P. (1993), "Order of Accuracy of QUICK and Related Convection-Diffusion Schemes," NASA TM 106402, ICOMP–93–47, November 1993.

Leonard, B. P. (1995a), "Comments on the Policy Statement on Numerical Accuracy," *ASME Journal of Fluids Engineering*, Vol. 117, No. 1, March 1995, pp. 5–6.

Leonard, B. P. (1995b), "Uncertainty in Future of CFD," ICOMP News, Ohio Aerospace Institute, Cleveland, Ohio, Vol. 2, No. 1, January 1995, pp. 1–2.

Leonard, B. P. and Drummond, J. E. (1995a), "Why You Should Not Use 'Hybrid', 'Power Law' or Related Exponential Schemes for Convective Modelling - There Are Much Better Alternatives," *International Journal for Numerical Methods in Fluids*, Vol. 20, 1995, pp. 421–442.

Leonard, B. P. and MacVean, M. K. (1995b), "Grid Refinement Study of a Benchmark Problem Using the Flux-Integral Method," in Johnson and Hughes (1995), pp. 45–50.

Leonard, B. P., MacVean, M. K., and Lock, A. P. (1995), "The Flux Integral Method for Multidimensional Convection and Diffusion," NASA TM 106679, ICOMP–94–13. Also, *Applied Mathematical Modeling*, Vol. 19, pp. 333–342.

Le Provost, C. and Vincent, P. (1986), "Some Tests of Precision for a Finite Element Model of Ocean Tides," *Journal of Computational Physics*, Vol. 65, 1986, pp. 273–291.

Lester, D. H., Jansen, G., and Burkholder, H. C. (1981), "Migration of Radionuclide Chains Through an Adsorbing Medium," *AIChE Symposium Series*, Vol. 71, No. 152.

Levins, R. (1966), "The strategy of model building in population biology," *American Scientist*, Vol. 54, pp. 421-431.

Levy, L. L. Jr. (1978), "Experimental and Computational Steady and Unsteady Transonic Flows About a Thick Airfoil," *AIAA Journal*, Vol. 16, No. 6, 1978, pp. 564–572.

Levy, Y., Hesselink, L. and Degani, D. (1995), "Anomalous Asymmetries in Flows Generated by Algorithms that Fail to Conserve Symmetry," *AIAA Journal*, Vol. 33, 1995, pp. 999–1007.

Levy, D. W., Zickuhr, T., Vassberg, J., Agrawal, S., Wahls, R. A., Pirzadeh, S. and Hemsch, M. J. (2002), "Summary of Data from the First AIAA CFD Drag Prediction Workshop," AIAA-2002-0841, *40th AIAA Aerospace Sciences Meeting*, Reno, NV, January 2002.

Levy, D. W., Zickuhr, T., Wahls, R. A., Pirzadeh, S. and Hemsch, M. J. (2003), "Data Summary from the First AIAA Computational Fluid Dynamics Drag Prediction Workshop," *Journal of Aircraft*, Vol. 40, pp. 875-882, 2003.

Lewis, C. H. (1990), "Comments on the Need for CFD Code Validation," *AIAA Journal of Spacecraft and Rockets*, Vol. 27, No. 2, March–April 1990, p. 97.

Li, X. D. and Wiberg, N.-E. (1994), "A posteriori error estimate by element patch post-processing, adaptive analysis in energy and L2 norms", Computers and Structures, Vol. 53, No. 4, 1994, pp. 907-919.

Lin, C. C. (1967), *The Theory of Hydrodynamic Stability*, Cambridge University Press, London.

Lockman, W. K., Lawrence, S. L. and Cleary, J. W. (1992), "Flow Over an All-Body Hypersonic Aircraft: Experiment and Computation," *Journal of Spacecraft and Rockets*, Vol. 29, No. 1, 1992, pp. 7–15.

Logan, R. W. and Nitta, C. K. (2006), "Comparing Ten Methods for Solution Verification, and Linking to Model Validation," *Journal of Aerospace Computing, Information, and Communication*, Vol. 3, pp. 354-373.

Lohner, R. (1989), "Adaptive Remeshing for Transient Problems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 61, 1989, pp. 323–338.

Lorrain, P. and Corson, D. P. (1962), *Introduction to Electromagnetic Fields and Waves*, W. H. Freeman Company, San Francisco, 1962.

Lotz, R. D., Thompson, B. E., Konings, C. A., and Davoudzadeh, F. (1995), "Grid Dependence and Numerical Uncertainty Analysis of a Transonic Supercritical Airfoil Calculations," in Johnson and Hughes (1995), pp. 13–18.

Lotz, R. D., Thompson, B. E., Konings, C. A., and Davoudzadeh, F. (1997), "Numerical Uncertainties in Transonic Flow Calculations for Aerofoils with Blunt Trailing Edges," *International Journal for Numerical Methods in Fluids*, Vol. 24, pp. 355–373.

Luettich, R. A. and Westerink, J. W. (1994), "Continental Shelf Scale Convergence Studies with a Baratropic Tidal Model," *Quantitative Skill Assessment for Coastal Ocean Models*, D. R. Lynch and A. M. Davies, Eds., AGU Press.

Lynch, P. (2006), *The Emergence of Numerical Weather Prediction: Richardson's Dream*, Cambridge University Press, England.

Lyness, J. N. (1974), *Mathematics of Computation*, Vol. 28, p. 81.

MacIntyre, W. G., Boggs, J. M., Antworth, C. P. and Stauffer, T. B. (1992), "Degradation Kinetics of Aromatic Organic Solutes Introduced Into a Heterogeneous Aquifer," *Water Resources Research*, Vol. 29, 1993, pp. 4045–4051.

MacKinnon, R. (1996), personal communication.

Madnia, C. K., Jaberi, F. A. and Givi, P. (2006), "Large Eddy Simulation of Heat and Mass Transport in Turbulent Flows," Chapter 5 of *Handbook of Numerical Heat Transfer*, Second Edition, W J. Minkowycz, E. M. Sparrow, and J. Y. Murthy, eds., Wiley, New York.

Mandelbrot, B. B. (1977), *Fractals: Form, Chance, and Dimension*, W. H. Freeman and Company, San Francisco.

Martin, F. W. Jr., Sickles, W. L. and Stanley, S. A. (1993), "Transonic Wind Tunnel Wall Interference Analysis for the Space Shuttle Vehicle," AIAA Paper 93–0420, January 1993.

Marvin, J. G., Levy, L. L. Jr., and Seegmiller, H. L. (1980), "On Turbulence Modeling for Unsteady Transonic Flows," *AIAA Journal*, Vol. 18, No. 5, May 1980, pp. 489–496.

Marvin, J. G. (1988), "Accuracy Requirements and Benchmark Experiments for CFD Validation," in AGARD (1988), Paper No. 2.

Marvin, J. G. (1995), "Perspective on Computational Fluid Dynamics Validation," *AIAA Journal*, Vol. 33, No. 10, October 1995, pp. 1778–1787.

Mathur, S. R., Minkowycz, W. J., Sparrow, E. M. and Murthy, J. Y. (2006), Overview of Numerical Method and Recommendations," Chapter 28 of *Handbook of Numerical Heat Transfer*, Second Edition, W J. Minkowycz, E. M. Sparrow, and J. Y. Murthy, eds., Wiley, New York.

Mavriplis, D. J., et al (2009), "Grid Quality and Resolution Issues from the Drag Prediction Workshop Series," *Journal of Aircraft*, to appear.

McCombie, C. and McKinley, I. (1993), "Validation-Another Perspective," *Ground Water*, Vol. 31, 1993, pp. 530–531.

McCormick, S. F. (1989), *Multilevel Adaptive Methods for Partial Differential Equations*, SIAM Frontiers in Applied Mathematics, Vol. 6, SIAM, Philadelphia, Pennsylvania.

McCroskey, W. J. (1981), "The Phenomenon of Dynamic Stall," NASA TM 81264.

McDevitt, J. B., Levy, L. L., and Deiwert, G. S. (1976), "Transonic Flow About a Thick Circular-Arc Airfoil," *AIAA Journal*, Vol. 14, No. 5, May 1976, pp. 606–613.

McDonald, M. G. and Harbaugh, W. (1988), "A Modular Three-Dimensional Finite-Difference Ground-Water Flow Model," Techniques of Water-Resources Investigations of the United States Geological Survey, Book 6, Modeling Techniques, Chapter A1. Scientific Software Group, P.O. Box 23401, Washington, D.C. 20026.

McIlroy, H. M. Jr., McEligot, D. M., Schultz, R. R., Christensen, D., Pink, R. J. and Johnson, R. C. (2006), "PIV Experiments to Measure Flow Phenomena in a Scaled Model of a VHTR Lower Plenum," INL/EXT-08-15161, 28 September 2006.

McWilliams, J. C. (2007), "Irreducible imprecision in atmospheric and ocean simulations," *Proc. National Academy of Sciences*, Vol. 104, N0. 21, May 2007, pp. 8709-8713.

Mehta, U. B. (1977), "Dynamic Stall of an Oscillating Airfoil," AGARD Conference Proceedings CP–227, *Symposium on Unsteady Aerodynamics*, Ottowa, Canada, Paper 23.

Mehta, U. (1988), "Flight Performance Estimation Utilizing Computational Fluid Dynamics," *Proceedings of the 5th National Aerospace Plane Technology Symposium*, Volume 1, NASP CP–5028, NASA Langley Research Center, Hampton, VA, October 1988.

Mehta, U. (1989), "Computational Requirements for Hypersonic Flight Performance Estimates," AIAA Paper 89–1670, June 1989.

Mehta, U. B. (1991), "Some Aspects of Uncertainty in Computational Fluid Dynamics Results," *ASME Journal of Fluids Engineering*, Vol. 113, 1991, pp. 538–543. (See also NASA TM 102804, March 1990.)

Mehta, U. B. (1992a), "NASP and SDI Spearhead CFD Developments," *Aerospace America*, February 1992.

Mehta, U. B. (1992b), "Aerospace Plane Design Challenge: Credible Computations," *Journal of Aircraft*, Vol. 30, No. 4, July–August 1993, pp. 519–525.

Mehta, U. (1995), "Guide to Credible Computational Fluid Dynamics Simulations," AIAA Paper 95–2225, June 1995.

Mehta, U. (1996), "Guide to Credible Computational Fluid Dynamics Simulations," *AIAA Journal of Propulsion and Power*, Vol. 12, No. 5, October 1996, pp. 940–948.

Melnik, R. E., Siclari, M. J., Barber, T., and Verhoff, A. (1994), "A Process for Industry Certification of Physical Simulation Codes," AIAA Paper 94–2235, AIAA 25th Fluid Dynamics Conference, Colorado Springs, Colorado, June 1994.

Melnik, R. E., Siclari, M. J., Marconi, F., Barber, T. and Verhoff, A. (1995), "An Overview of a Recent Industry Effort at CFD Code Validation," AIAA Paper 95–2229, 26th AIAA Fluid Dynamics Conference, San Diego, California, June 19–23, 1995.

Menter, F. R. (1992), Improved Two-Equation k-ω Turbulence Models of Aerodynamic Flows," NASA TM 103975, October 1992.

Merton, T. (1965), *The Way of Chuang Tzu*, "Means and Ends," New Directions, New York, 1965, p.154.

Miller, W. (1989), *Software Quality Assurance Requirements, Revision B, Department Operating Procedures*, Yucca Mountain Project Department, Sandia National Laboratories, Albuquerque, New Mexico.

Mills, R. J. (1987), *A Posteriori Error Estimates in Computational Fluid Mechanics*, Ph.D. Dissertation, University of New South Wales, School of Mechanical and Industrial Engineering, Sydney, Australia, March 1987.

Mittal, R. and Balachnadar, S. (1995), "Vortical Structures in Bluff Body Wakes," AIAA Paper 95–0867, 33rd AIAA Aerospace Sciences Meeting, Reno, Nevada, January 1995.

Moffat, R. J. (1981), "Contribution to the Theory of Uncertainty Analysis for Single-Sample Experiments," *The 1980–81 AFOSR/HTTM–Stanford Conference on Complex Turbulent Flows*, Vol. 1, ThermoSciences Division, Mechanical Engineering Department, Stanford University.

Morgan, K., Peraire, J., Peiro, J., Hassan, O., and Probert, E. J. (1991), "Unstructured Grid Methods for Compressible Flows," *Computational Mechanics*, Y. K. Cheung, J. H. W. Lee and A. Y. Y. Leung, Eds., Vol. 2, Balkema, Rotterdam, pp. 1489–1497.

Morrison, J. H. and Hemsch, M. J. (2007), "Statistical Analysis of CFD Solutions from the Third AIAA Drag Prediction Workshop," AIAA Paper 2007-0254, January 2007. See also "Statistical Analysis of the AIAA Drag Prediction Workshop CFD Solutions," Paper 55, NATO RTO-MP-AVT-147 Workshop, December 2007.

Mueller, T. J., Hall, C. R., and Roache, P. J. (1970), "Influence of Initial Flow Direction on the Turbulent Base Pressure in Supersonic Axisymmetric Flow," *AIAA Journal of Spacecraft and Rockets*, Vol. 7, No. 12, December 1970.

Mueller, T. J. and Roache, P. J. (1973), "Base Drag Calculations in Supersonic Turbulent Axisymmetric Flows," *AIAA Journal of Spacecraft and Rockets*, Vol. 10, No. 4, April 1973.

Mueller, T. J. (1974), "On the Fluid Dynamics of Prosthetic Heart Valve Flow - A Preliminary Numerical and Experimental Study," Technical Note 101, Von Karman Institute, Rhode Saint Genese, Belgium, June 1974.

Muller, A. (1994), Ed., *Proceedings of the Basel World CFD User Days 1994, Second World Conference in Applied Computational Fluid Dynamics*, Basel, Switzerland, 1–5 May 1994.

Muller, A. and Loffler, B. (1996), Eds., *Proceedings of the Basel World CFD User Days 1996, Third World Conference in Applied Computational Fluid Dynamics*, Basel, Switzerland, May 1996.

Myers, I. (1962), *Manual: The Myers-Briggs Type Indicator*, Consulting Psychologists Press, Palo Alto, CA.

NAFEMS (2006), "NAFEMS Website," National Agency for Finite Element Methods and Standards, 2006, www.NAFEMS.org.

NASA (2008), *Standard for Models and Simulations*, NASA-STD-7009.

Naylor, T. H. and Finger, J. M. (1967), "Verification of Computer Simulation Models," *Management Science*, Vol. 14, No. 2, 1967, pp. B92–B101.

Newling, B. et al (1997), "Comparisons of Magnetic Resonance Imaging Velocimetry with Computational Fluid Dynamics," *ASME Journal of Fluids Engineering*, Vol. 119, March 1997, pp. 103–109.

Nguyen, T. V. and Maclaine-Cross, I. L. (1988), "Incremental Pressure Drop Number in Parallel-Plate Heat Exchangers," *ASME Journal of Fluids Eng*ineering, Vol. 110, March 1988, pp. 93–96.

Nies, A., Laurens, J.-M., Galson, D. A., and Webster, S. (1990), *PSACOIN Level 1A Intercomparison*, Nuclear Energy Agency, Organisation for Economic Co-Operation and Development, Paris, November 1990.

NSF (2006), *Simulation-Based Engineering Science: Revolutionizing Engineering Science through Simulation*, Report of the NSF Blue Ribbon Panel on Simulation-Based Engineering Science.

Oberkampf, W. L. (1994), "A Proposed Framework for Computational Fluid Dynamics Code Calibration/Validation," AIAA Paper 94–2540, 18th AIAA Aerospace Ground Testing Conference, Colorado Springs, Colorado, 20–23 June 1994.

Oberkampf, W. L., Blottner, F. G., and Aeschliman, D. P. (1995), "Methodology for Computational Fluid Dynamics Code Verification/Validation," AIAA Paper 95–2226, 26th AIAA Fluid Dynamics Conference, San Diego, California, 19–22 June 1995.

Oberkampf, W. L. and Blottner, F. G. (1997), "Issues in Computational Fluid Dynamics Code Verification and Validation," SAND95–1352, Sandia National Laboratories, Albuquerque, New Mexico, September 1997.

Oberkampf, W. L. (1998), personal communication and draft of AIAA Standard Guide.

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

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

Oberkampf, W. L. and Trucano, T. G. (2007), "Verification and Validation Benchmarks," SAND2007-0853, Sandia National Laboratories, Albuquerque, New Mexico, February 2007.

Oberkampf, W. L. and Trucano, T. G. (2008), "Verification and Validation Benchmarks," *Nuclear Science and Engineering*, Vol. 238, No. 3, March 2008, pp. 716-743.

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

Oden, J. T., Kennon, S. R., Tworzydlo, W. W., Bass, J. M., and Berry, C. (1993), "Progress on Adaptive h-p Finite Element Methods for the Incompressible Navier-Stokes Equations," *Computational Mechanics*, Vol. 11, 1993, pp. 421–432.

OECD (1988), *The International HYDROCOIN Project: Groundwater Hydrology Modelling Strategies for Performance Assessment of Nuclear Waste Disposal. Level 1: Code Verification*, Nuclear Energy Agency, Organisation for Economic Co-Operation and Development (OECD), Paris.

O'Hagan, A. (2004), "Dicing with the Unknown," *Significance*, Vol. 1, pp. 132-133.

O'Hagan, A. (2008), personal communication. 8 May 2008.

Oldenburg, C. M. and Pruess, K. (1995), "Dispersive Transport Dynamics in a Strongly Coupled Groundwater-Brine Flow System," *Water Resources Research*, Vol. 31, No. 2, February 1995, pp. 289–302.

Olsen, M. E. and Seegmiller, H. L. (1993), "Low Aspect Ratio Wing Code Validation Experiment," *AIAA Journal*, Vol. 31, No. 10, 1993, pp. 1744–1752.

Oreskes, N., Shrader-Frechette, K., and Belitz, K. (1994), "Verification, Validation, and Confirmation of Numerical Models in the Earth Sciences," *Science*, Vol. 263, 4 February 1994, pp. 641–646.

Oreskes, N. (1999), *The Rejection of Continental Drift; Theory and Method in American Earth Science*, Oxford University Press, New York, Oxford.

Osborne, W. M. (1988), "Developers Must Design In Maintainability," *IEEE Software*, May 1988.

Pace, D. (2003), personal communication, 24 March 2003.

Padra, C. and Larreteguy, A. (1995), "A-Posteriori Error Estimator for the Control-Volume Finite-Element Method as Applied to Convection-Diffusion Problems," *Numerical Heat Transfer, Part B*, Vol. 27, pp. 63–80.

Patankar, S. V. (1980), *Numerical Heat Transfer and Fluid Flow*, McGraw-Hill, New York, 1980.

Patel, R. (1973), "An Experimental Study of a Plane Mixing Layer," *AIAA Journal*, Vol. 11, No. 1, January 1973.

Pelletier, D. and Ilinca, F. (1994), "Adaptive Remeshing for the k-ε Model of Turbulence," AIAA Paper 94–0818, 32nd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 10–13, 1994.

Pelletier, D., Ignat, L. and Ilinca, F. (1995), "An Adaptive Finite Element Method for Conjugate Heat Transfer," AIAA Paper 95–0637, AIAA 33rd Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 9–12, 1995.

Pelletier, D. and Ignat, L. (1995), "On the Accuracy of the Grid Convergence Index and the Zhu-Zienkiewicz Error Estimator," in Johnson and Hughes (1995), pp. 31–36.

Pelletier, D. (1996), personal communication.

Pelletier, D. and Ilinca, F. (1997), "Adaptive Remeshing for the k-ε Model of Turbulence," *AIAA Journal*, Vol. 35, No. 4, April 1997, pp. 640–646.

Pelletier, D. and Trepanier, J.-Y. (1997), "Implementation of Error Analysis and Norms to Computational Fluid Dynamics Applications, Parts 1-5," Project C.D.T. P2223, Ecole Polytechnique de Montreal, July 1997.

Pelletier, D. (1999), personal communication.

Pelletier, D. and Roache, P. J. (2002), "CFD Code Verification and the Method of Manufactured Solutions", CFD 2002, 10th Annual Conference of the CFD Society of Canada, 9-11 June 2002, Windsor, Ontario, Canada.

Pelletier , D. and Roache, P. J. (2006), "Verification and Validation of Computational Heat Transfer", Chapter 13 of *Handbook of Numerical Heat Transfer*, Second Edition, W J. Minkowycz, E. M. Sparrow, and J. Y. Murthy, eds., Wiley, New York.

Pelletier, D. (2008), "Comments on the Third Lisbon Workshop on Uncertainty Assessment in CFD", *Proc. 3rd Workshop on CFD Uncertainty Analysis*, Lisbon, 23-24 October 2008, Instituto Superior Técnico, Lisbon. L. Eça, and M. Hoekstra, eds.

Pepper, D. M. (2006), "Meshless Methods," Chapter 7 of *Handbook of Numerical Heat Transfer*, Second Edition, W J. Minkowycz, E. M. Sparrow, and J. Y. Murthy, eds., Wiley, New York.

Pernice, M. (2007), Seminar at Idaho National Laboratories, Idaho Falls, Idaho, "Requirements and Capabilities for CFD Analysis of Reactors" 6 November 2007.

Piacsek, S. and Allard, R. (1993), "The Importance of Horizontal Resolution in Coupled Ice/Ocean Models of the Arctic," *SIAM Conference on Mathematical and Computational Issues in the Geosciences*, Houston, Texas, April 19–21, 1993.

Pironneau, O. (1990), Ed., "Summary and Conclusions for Test Case T2," in *Numerical Simulation of Unsteady Flows and Transition to Turbulence*, Cambridge University Press.

Pollard, A. (1994), "Validation of CFD Codes," *CFD Bulletin*, CFD Society of Canada, Winter 1994, pp. 2–3.

Pollard, A. (1995), "Validation Exercise of Turbulent CFD Codes," Proceedings CFD Society of Canada Meeting, Banff, July 1995.

Polya, G. (1957), *How to Solve It; A New Aspect of Mathematical Method*, Princeton University Press, Princeton, New Jersey.

Popper, K.. (1980), *The Logic of Scientific Discovery*, Routledge, London.

Porter, J. L. (1996), "A Summary/Overview of Selected Computational Fluid Dynamics (CFD) Code Validation/Calibration Activities," AIAA Paper 96–2053, 27th AIAA Fluid Dynamics Conference, New Orleans, Louisiana, June 17–20, 1996.

Powers, J. M. and Gonthier, K. A. (1992), "Reaction Zone Structure for Strong, Weak Overdriven and Weak Underdriven Oblique Detonations," *Physics of Fluids A*, Vol. 4, No. 9, September 1992, pp. 2082–2089.

Powers, J. M. and Stewart, D. S. (1992), "Approximate Solutions for Oblique Detonations in the Hypersonic Limit," *AIAA Journal*, Vol. 30, No. 3, March 1992, pp. 726–736.

Prabhakar, V. and Reddy, J. N. (2006), "Spectral/hp Penalty Least-Squares Finite Element Formulation for the Steady Incompressible Navier-Stokes Equations," *Journal of Computational Physics*, Vol. 215, No. 1, pp. 274-297.

Pulliam, T. H. (1986), "Efficient Solution Methods for the Navier-Stokes Equations," *Lecture Notes for the Von Karman Institute for Fluid Dynamics, Lecture Series: Numerical Techniques for Viscous Flow Computation in Turbomachinery Bladings*, Von Karman Institute for Fluid Dynamics, Brussels, Belgium, January 20–24, 1986.

Puttre, M. (1994), "Verifying Analytical Models," *Mechanical Engineering*, Vol. 116, No. 11, November 1994, pp. 70–74.

Quigley, W. (2008), "Financial Risk Models Don't Fly," *Albuquerque Journal*, 20 December 2008, pg. 1.

Quirk, J. (2007), Amrita. www.amrita-cfd.org.

Ragusa, J. (2008), "Discretization and Coupling Error Control in Multiphysics Applications: Current Status and the Way Forward," in Anon. (2008).

Rakich, J. V. (1973), "Introduction to the Proceedings of the Symposium on Computational Fluid Dynamics," Proceedings of the AIAA Computational Fluid Dynamics Conference, Palm Springs, California, July 19–20, 1973, p.1.

Raven, H. C., Hoekstra, M. and Eça, L. (2002), "A Discussion of Procedures for CFD Uncertainty Analysis," MARIN Report 17678-1-RD, Maritime Institute of the Netherlands, October 2002. www.marin.nl/publications/pg_resistance.html.

Rechard, R. P., Roache, P. J., Blaine, R, L., Gilkey, A. P., and Rudeen, D. K. (1991), *Quality Assurance Procedures for Computer Software Supporting Performance Assessments of the Waste Isolation Pilot Plant*, SAND90–1240, Sandia National Laboratories, Albuquerque, New Mexico, April 1991.

Rechard, R. P., Rudeen, D. K., and Roache, P. J. (1992), *Quality Assurance Procedures for Analyses and Report Reviews Supporting Performance Assessments of the Waste Isolation Pilot Plant*, SAND91–0428, Sandia National Laboratories, Albuquerque, New Mexico, October 1992.

Reed, H. L. and Saric, W. S. (1989), "Stability of Three-Dimensional Boundary Layers," *Annual Review of Fluid Mechanics*, Vol. 21, 1989, pp. 235–284.

Reed, H. L., Saric, W. S., and Arnal, D. (1996), "Linear Stability Theory Applied to Boundary Layers," *Annual Review of Fluid Mechanics*, Vol. 28, 1996, pp. 389–428.

Reeves, M., Ward, D. S., Johns, N. D., and Cranwell, R. M. (1986), "Theory and Implementation for SWIFT II The Sandia Waste-Isolation Flow and Transport Model for Fractured Media," SAND83–1159, Sandia National Laboratories, Albuquerque, New Mexico, 1986.

Refsgaard, J. C. and Henriksen, H. J. (2004), "Modeling Guidelines - Terminology and Guiding Principles," *Advances in Water Resources*, vol. 27, pp. 71-82.

Rehfeldt, K. R., Boggs, J. M., and Gelhar, L. W. (1992), "Field Study of Dispersion in a Heterogeneous Aquifer. 3. Geostatistical Analysis of Hydraulic Conductivity," *Water Resources Research*, Vol. 28, 1992, pp. 3309–3324.

Rehmeyer, J. (2007), "Cloudy Crystal Balls," *Science News*, 1 August 2007.

Richards, S. A. (1997), "Completed Richardson Extrapolation in Space and Time, *Communications in Numerical Methods in Engineering*, Vol. 13, No. 7, 573–582.

Richardson, L. F. (1908), The Approximate Arithmetical Solution by Finite Differences of Physical Problems Involving Differential Equations, with an Application to the Stresses in a Masonry Dam," *Transactions of the Royal Society of London*, Series A, Vol. 210, pp. 307–357.

Richardson, L. F. (1922), *Weather Prediction by Numerical Process*, Cambridge University Press. (Re-published in 1965.)

Richardson, L. F. (1925), "How to Solve Differential Equations Approximately by Arithmetic", *The Mathematical Gazette*, London, Vol. XII, No. 177, July 1925, pp. 415–421.

Richardson, L. F. (1927), "The Deferred Approach to the Limit," *Transactions of the Royal Society of London, Series A.*, Vol. 226, pp. 299–361. "Part I. Single Lattice." "Part II. Interpenetrating Lattices" by J. A. Gaunt.

Richardson, L. F. (1960a), *Arms and Insecurity: A Mathematical Study of the Causes and Origins of War*, Ed. N. Rashevsky and E. Trucco, Boxwood Press, Pittsburgh (now, Pacific Grove).

Richardson, L. F. (1960b), *Statistics of Deadly Quarrels*, Q. Wright and C. C. Kienau, eds., Boxwood Press, Pittsburgh (now, Pacific Grove).

Richardson, L. F. (1961), "The Problem of Contiguity: An Appendix of Statistics of Deadly Quarrels", *General Systems Yearbook*, Vol. 6, 1961, pp. 139–187.

Richtmyer, R. D. and Morton, K. W. (1967), *Difference Methods for Initial-Value Problems*, Interscience Publishers, New York.

Rizzi, A. and Vos, J. (1996), "Towards Establishing Credibility in CFD Simulations," AIAA Paper 96–2029, 27th AIAA Fluid Dynamics Conference, New Orleans, Louisiana, June 17–20, 1996. See also *AIAA Journal*, 1998, Vol. 36, No. 5, pp. 68-675.

Rizzo, A. R. (1991), "Estimating Errors in FE Analysis," *Mechanical Engineering*, Vol. 113, No. 5, May 1991, pp. 61–63.

Roache, P. J. and Mueller, T. J. (1970), "Numerical Solutions of Laminar Separated Flows," *AIAA Journal*, Vol. 8, No. 3, March 1970, pp. 530–538. See also AIAA Paper 68–741, June 1968.

Roache, P. J. (1972a), "On Artificial Viscosity," *Journal of Computational Physics*, Vol. 10, No. 2, October 1972, pp. 169–184.

Roache, Patrick J. (1972b), *Computational Fluid Dynamics*, First Edition, Hermosa Publishers, New Mexico, USA. Revised Printing 1976. See also Roache (1998b).

Roache, P. J. (1975a), "Computers vs. Wind Tunnels," in Letters, *Astronautics and Aeronautics*, September 1975, pp. 4–5.

Roache, P. J. (1975b), "The LAD, NOS, and Split NOS Methods for the Steady-State Navier-Stokes Equations," *Computers and Fluids*, Vol. 3, 1975, pp. 179–195.

Roache, P. J. and Ellis, M. A. (1975), "The BID Method for the Steady-State Navier-Stokes Equations," *Computers and Fluids*, Vol. 3, 1975, pp. 305–320.

Roache, P. J. (1978), "A Pseudo-Spectral FFT Technique for Non-Periodic Problems," *Journal of Computational Physics*, Vol. 27, No. 2, May 1978.

Roache, P. J. (1982a), "Scaling of High-Reynolds-Number Weakly Separated Channel Flows," in *Numerical and Physical Aspects of Aerodynamic Flows*, T. Cebeci, Ed., Chapter 6, Springer Verlag, NY, pp. 87-98.

Roache, P. J. (1982b), "Semidirect/Marching Methods and Elliptic Grid Generation," *Symposium on the Numerical Generation of Curvilinear Coordinate Systems and Use in the Numerical Solution of Partial Differential Equations*, J. F. Thompson, Ed., Nashville, Tennessee, Elsevier Science Publishing Company, April 1982, pp. 729–737.

Roache, P. J., Steinberg, S., and Moeny, W. M. (1984a), "Interactive Electric Field Calculations for Lasers," AIAA Paper 84–1655, AIAA 17th Fluid Dynamics, Plasma Physics, and Lasers Conference, Snowmass, Colorado, 25–27 June 1984.

Roache, P. J. and Steinberg, S. (1984b), "Symbolic Manipulation and Computational Fluid Dynamics," *AIAA Journal*, Vol. 22, October 1984, pp. 1390–1394. See also AIAA Paper No. 83–1952.

Roache, P. J. and Steinberg, S. (1985), "A New Approach to Grid Generation Using a Variational Formulation," AIAA Paper 85–1527, AIAA 7th Computational Fluid Dynamics Conference, Cincinnati, Ohio, July 15–17, 1985.

Roache, P. J., Ghia, K. and White, F. (1986), "Editorial Policy Statement on the Control of Numerical Accuracy," *ASME Journal of Fluids Engineering*, Vol. 108, No. 1, March 1986, p. 2.

Roache, P. J. and Dietrich, D. E. (1988), "Evaluation of the Filtered Leapfrog-Trapezoidal Time Integration Method," *Numerical Heat Transfer*, Vol. 14, 1988, pp. 149–164.

Roache, P. J. and Salari, K. (1990), "Weakly Compressible Navier-Stokes Solutions with an Implicit Approximate Factorization Code," AIAA Paper 90–0235, 28th AIAA Aerospace Sciences Meeting, Reno, Nevada, January 8–11, 1990.

Roache, P. J. (1990), "Need for Control of Numerical Accuracy," *AIAA Journal of Spacecraft and Rockets*, Vol. 27, No. 2, March–April 1990, pp. 98–102.

Roache, P. J., Knupp, P. M., Steinberg, S., Blaine, R. L. (1990a), "Experience with Benchmark Test Cases for Groundwater Flow." ASME FED Vol. 93, *Benchmark Test Cases for Computational Fluid Dynamics*, I. Celik and C. J. Freitas, Eds., Book No. H00598, June 1990, pp. 49–56.

Roache, P. J., Mikatarian, R. R., Chandler, L. J., Steinberg, S., Salari, K., and Sery, P. (1990b), "Adaptive Grids for RAVEN: Vol. I of IV, Project Summary," WL–TN–90–10, Air Force Systems Command, Kirtland AFB, NM, June 1990.

Roache, P. J. (1991), "Computational Fluid Dynamics Algorithms and Codes Developed for WIPP Site Simulations," in *Computational Mechanics*, Cheung, Y. K., Lee, J. H. W. and Leung, A. Y. T., Eds., *Proceedings of the Asian Pacific Conference on Computational Mechanics*, A. A. Balkema, Rotterdam, Vol. 2, Hong Kong, 11–13 December 1991, pp. 1325–1335.

Roache, P. J. (1992a), "A Flux-Based Modified Method of Characteristics," *International Journal for Numerical Methods in Fluids*, Vol. 15, pp. 1259–1275

Roache, P. J. (1992b), "Computational Fluid Dynamics Algorithms Developed for WIPP Site Simulations," in Russell, T. F. et al, Eds., *Proceedings of the IX International Conference on Computational Methods in Water Resources*, Denver, Colorado, Elsevier Applied Science, New York, June 9–12, 1992, pp. 375–382.

Roache, P. J. (1993a), "The SECO Suite of Codes for Site Performance Assessment," *Proceedings of the 1993 International High-Level Radioactive Waste Management Conference*, Las Vegas, Nevada, April 26–30, 1993.

Roache, P. J. (1993b), "The SECO Code Algorithms for Groundwater Flow and Transport," *Finite Elements in Fluids: New Trends and Applications*, Part II, Proceedings of the VIII International Conference on Finite Elements in Fluids, Univ. Politécnica de Cataluña, Barcelona, Spain, K. Morgan, E. Onate, J. Periaux, J. Peraire, and O. C. Zienkiewicz, Eds., 20–23 Sept. 1993, pp. 939–948.

Roache, P. J. (1993c), "A Method for Uniform Reporting of Grid Refinement Studies," *Quantification of Uncertainty in Computational Fluid Dynamics*, ASME FED–Vol. 158, pp. 109–120.

Roache, P. J. and Knupp, P. M. (1993), "Completed Richardson Extrapolation," *Communications in Applied Numerical Methods*, Volume 9, pp. 365–374.

Roache, P. J. (1994a), "Response: To the Comments by Drs. W. Shyy and M. Sindir," *ASME Journal of Fluids Engineering*, Vol. 116, No. 2, June 1994, pp. 198–199.

Roache P. J. (1994b), "Perspective: A Method for Uniform Reporting of Grid Refinement Studies," *ASME Journal of Fluids Engineering*, Sept. 1994, Vol. 116, pp. 405–413. See also Roache, P. J. (1993), "A Method for Uniform Reporting of Grid Refinement Studies," in Celik et al (1993), pp. 109–120. See also Roache, P. J. (1993), *Proceedings of the AIAA 11th Computational Fluid Dynamics Conference*, Part 2, Orlando, Florida, 6–9 July 1993, pp. 1057–1058.

Roache, P. J. (1995a), "Verification of Codes and Calculations," AIAA Paper 95–2224, 26th AIAA Fluid Dynamics Conference, San Diego, California, June 19–22, 1995. Also, *AIAA Journal*, May 1998.

Roache, P. J. (1995b), *Elliptic Marching Methods and Domain Decomposition*, CRC Press, Boca Raton, Florida.

Roache, P. J., Blaine, R. L., and Baker, B. L. (1995), "SECOFL2D User's Manual," SAND90–7096, Sandia National Laboratories, Albuquerque, New Mexico, October 1995.

Roache, P. J., Blaine, R. L., and Baker, B. (1996a), "SECOFL2D User's Manual," Appendix I of "WIPP PA User's Manual for SECOFL2D, Version 3.03," WPO # 37271, Sandia National Laboratories, Albuquerque, New Mexico, May 7, 1996.

Roache, P. J., Knupp, P., Salari, K., et al (1996b), "High Accuracy Robust Groundwater Flow and Transport Codes for Environmental Engineering: Phase II," Final Report, F08637–94–C6049 for Armstrong Laboratory, Tyndall AFB, Florida, Ecodynamics Research Associates, Albuquerque, New Mexico.

Roache, P. J. and Rucker, D. (1996), "SECO Code Development for MADE-2 Simulations," Proceedings of the 3rd Army Groundwater Model Users Workshop, Albuquerque, New Mexico, 31 July 1996.

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

Roache, P. J. (1998a), *Verification and Validation in Computational Science and Engineering*, Hermosa Publishers, Albuquerque, August 1998.

Roache, P. J. (1998b), *Fundamentals of Computational Fluid Dynamics*, Hermosa Publishers, New Mexico 87119, USA. http://www.hermosa-pub.com/hermosa.

Roache, P. J. (1998c), "Uncertainties and CFD Code Validation," *ASME Journal of Fluids Engineering*, Vol. 120, September 1998, pp. 635-636.

Roache, P. J. (2000), "Educational Cornerstones for CFD", *Computational Fluid Dynamics Journal (Japan Society of Computational Fluid Dynamics)*, Vol. 9, No. 3, October 2000, pp. 200-207.

Roache, P. J. (2002), "Code Verification by the Method of Manufactured Solutions", *ASME Journal of Fluids Engineering*, Vol. 114, No. 1, March 2002, pp. 4-10.

Roache, P. J. (2003a), "Error Bars for CFD," AIAA Paper 2003-0408, AIAA 41st Aerospace Sciences Meeting, January 2003, Reno, Nevada.

Roache, P. J. (2003b), "Conservatism of the GCI in Finite Volume Computations on Steady State Fluid Flow and Heat Transfer", *ASME Journal of Fluids Engineering*, Vol. 125, No. 4, July 2003, pp 731-732.

Roache, P. J. (2003c), "Criticisms of the 'Correction Factor' Verification Method", *ASME Journal of Fluids Engineering*, Vol. 125, No. 4, July 2003, pp 732-733.

Roache, P. J. (2004a), "Building PDE Codes to be Verifiable and Validatable," *Computing in Science and Engineering*, Special Issue on Verification and Validation, September/October 2004, pp 30-38.

Roache, P. J. (2004b), "Calculation Verification: an Overview", *Proc. Workshop on CFD Uncertainty Analysis*, 21-22 October 2004. Instituto Superior Técnico, Lisbon, Portugal. L. Eça, and M. Hoekstra, eds.

Roache, P. J. (2006), "Recent Developments in Verification and Validation," *Proc. Second Workshop on CFD Uncertainty Analysis*, 19-20 October 2006. Instituto Superior Técnico, Lisbon, Portugal. L. Eça, and M. Hoekstra, eds.

Roache, P. J. (2008a), "Validation: Definitions or Descriptions?", *Proc. 3rd Workshop on CFD Uncertainty Analysis*, Lisbon, 23-24 October 2008, Instituto Superior Técnico, Lisbon. L. Eça, and M. Hoekstra, eds.

Roache, P. J. (2008b), "Comments on the Third Lisbon V&V Workshop", *Proc. 3rd Workshop on CFD Uncertainty Analysis*, Lisbon, 23-24 October 2008, Instituto Superior Técnico, Lisbon. L. Eça, and M. Hoekstra, eds.

Roache, P. J. (2009), "Perspective: Validation - What Does it Mean?", *ASME Journal of Fluids Engineering*, Vol. 131, Issue 3, March 2009, pg./CID 034503.

Roberts, L. (1988), in Round Table Discussion in AGARD (1988).

Romero, V. J. (2008), "Type X and Y Errors and Data and Model Conditioning for Systematic Uncertainty in Model Calibration, Validation, and Extrapolation," SAE Paper 2008-01-1368, *Society of Automotive Engineers 2008 World Congress*, April 14-17, 2008, Detroit, MI.

Rosenfeld, M. (1994), "Grid Refinement Test of Time-Periodic Flows over Bluff Bodies," *Computers and Fluids*, Vol. 23, No. 5, 1994, pp. 693–709.

Ross, S .M. (2003), "Peirce's Criterion for the Elimination of Suspect Experimental Data," *ASEE Journal of Engineering Technology*.

Roy, C. J. and Blottner, F. G. (2000). "Assessment of One- and Two-Equation Turbulence Models for Hypersonic Transitional Flows," AIAA Paper 2000-0132, Reno, NV.

Roy, C. J. and Blottner, F. G. (2001). "Assessment of One- and Two-Equation Turbulence Models for Hypersonic Transitional Flows," *Journal of Spacecraft and Rockets*, Vol. 38, No. 5, pp. 699-710.

Roy, C. J. (2003), "Grid Convergence Error Analysis for Mixed-Order Numerical Schemes," *AIAA Journal*, Vol. 41, No. 4, April 2003, pp. 595-604. See also AIAA Paper 2001-2606, Anaheim, June 2001.

Roy, C. J. and Blottner, F. G. (2003), "Methodology for Turbulence Model Validation: Application to Hypersonic Flows," *Journal of Spacecraft and Rockets*, Vol. 40, No. 3, 2003, pp. 313-325.

Roy, C. J. (2004), "Verification of Codes and Solutions in Computational Simulation," Proceedings of CHT-04-V2, ICHMT International Symposium on Advances in Computational Heat Transfer, April 19-24, 2004, Norway.

Roy, C. J., Nelson, C. C., Smith, T. M., and Ober, C. C. (2004), "Verification of Euler / Navier-Stokes Codes using the Method of Manufactured Solutions," *International Journal for Numerical Methods in Fluids*, Vol. 44, No. 6, 2004, pp. 599-620.

Rudy, D. H. et al (1991), "Computation of Laminar Hypersonic Compression-Corner Flows," *AIAA Journal*, Vol. 29, July 1991, pp. 1108–1113.

Ruge, J. W. (1995), "An Evaluation of the Grid Convergence for 1-D Two-Phase Flow in Porous Media", in Johnson and Hughes (1995), pp. 51–56.

Rumsey, C. L. and Anderson, W. K. (1988), "Parametric Study of Grid Size, Time Step, and Turbulence Modeling on Navier-Stokes Computations Over Airfoils," Paper 5 in AGARD (1988).

Rumsey, C. L., Sanetrik, M. D., Biedron, R. T., Melson, N. D. and Parlette, E. B. (1996), "Efficiency and Accuracy of Time-Accurate Turbulent Navier-Stokes Computations," *Computers and Fluids*, Vol. 25, No. 2, 1996, pp. 217–236.

Rumsey, C. L., Rivers, S. M. and Morrison, J. H. (2005), "Study of CFD Variation on Transport Configurations for the Second Drag-Prediction Workshop," *Computers & Fluids*, Vol. 34, pp. 785-816, 2005.

Rumsey, C. L., and Thomas, J. L. (2008), "Application of FUN3D and CFL3D to the Third Workshop on CFD Uncertainty Analysis," NASA/TM-2008-215537.

Russell. T. (1996), personal communication.

Rykiel, E. J. Jr. (1994), in "The Meaning of Models," Letters, *Science*, Vol. 264, 15 April 1994, pp. 330–331.

Rykiel, E. J. Jr. (1996), "The Meaning of Validation," *Ecological Modelling*, Vol. 90, No. 3, pp. 229-244.

Salari, K. and Roache, P. J. (1990), "The Influence of Sweep on Dynamic Stall Produced by a Rapidly Pitching Wing," AIAA Paper 90–0581.

Salari, K., Knupp, P., Roache, P. J., and Steinberg, S. (1992), "TVD Applied to Radionuclide Transport in Fractured Porous Media," *Numerical Methods in Water Resources*, Denver, Colorado.

Salari, K. and Steinberg, S. (1994), "Flux-Corrected Transport in a Moving Grid," *Journal of Computational Physics*, Vol. 111, No. 1, March 1994, pp. 24–32.

Salari, K., Roache, P. J., and Wilcox, D. C. (1994), "Numerical Simulation of Dynamic Stall Using the Two-Equation K-ω Turbulence Model," ASME Fluids Engineering Division Summer Meeting, Lake Tahoe, Nevada, 20–23 June 1994, ASME FED Vol. 196, K. Ghia and M. Hafez, Eds., pp. 1–7.

Salari, K., Blaine, R. L., Economy, K., and Roache, P. J. (1995), "Grid Resolution Studies of Radionuclide Transport in Fractured Porous Media," in R. W. Johnson and E. D. Hughes (1995), eds., *Quantification of Uncertainty in Computational Fluids Dynamics–1995*, Joint JSME–ASME Fluid Mechanics Meeting, ASME FED Vol. 213, Hilton Head, South Carolina, August 14–18, 1995, pp. 25–30.

Salari, K. (1995), personal communication.

Salari, K. and Blaine, R. L. (1996), "The SECOTP2D User's Manual," Sandia National Laboratories, Albuquerque, NM.

Salari, K. (1997), personal communication.

Salas, M. D. (2006), "Some Observations on Grid Convergence," *Computers and Fluids*, Jan. 2006.

Salas, M. D. (2008), "The Trouble with Drag," *Computers and Fluids*, 2008.

Sandia (1986), *Sandia Software Guidelines*, Vol. 3, SAND85–2346, Sandia National Laboratories, Albuquerque, New Mexico.

Sani, R. L. and Gresho, P. M. (1994), "Resume' and Remarks on the Open Boundary Condition Minisymposium," *International Journal for Numerical Methods in Fluids*, Vol. 18, 1994, pp. 983–1008.

Saric, W. S. (1994), "Goertler Vortices," in *Annual Review of Fluid Mechanics*, Vol. 26, 1994, pp. 379–409.

Schlesinger, S. (1979), Terminology for Model Credibility," *Simulation*, Vol. 32, No. 3, pp. 103–104.

Schonauer, W. (1981), personal communication; see also, Schonauer, W., Raith, K., and Glotz, G. (1981).

Schonauer, W., Raith, K., and Glotz, G. (1981), "The Principle of the Difference of Difference Quotients as a Key to the Self-Adaptive Solution of Nonlinear Partial Differential Equations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 28, No. 3, pp. 327–359.

Sclafani et al (2008), "CFL3D/OVERFLOW Results for DLR-F6 Wing/Body and Drag Prediction Workshop Wing," *Journal of Aircraft*, Vol. 45, No. 3, May-June 2008.

Sengupta, R., Orkwis, P. D., and Davis, S. M. (1995), "Unsteady Solutions, Solution Transients and Local Grid Resolution Requirements for Asymmetric Vortex Formation," AIAA Paper 95–0364, AIAA 33rd Aerospace Sciences Meeting, Reno, Nevada, January 9–12, 1995.

Serrano, S. E. (1995), "Analytical Solutions of the Nonlinear Groundwater Flow Equation in Unconfined Aquifers and the Effect of Heterogeneity," *Water Resources Research*, Vol. 31, No. 11, November 1995, pp. 2733–2742.

Settles, G. S. and Dodson, L. J. (1991), "Hypersonic Shock/Boundary-Layer Interaction Database," NASA CR–177577, April 1991.

Shampine, L. F., Watts, W. A., and Davenport, S. (1976), "Solving Non-Stiff Ordinary Differential Equations–The State of the Art," Sandia Laboratories Report SAND 75–0182, 1975, Sandia National Laboratories, Albuquerque, New Mexico. Also, *SIAM Review*, Vol. 18, No. 3, July 1976, pp. 376–411.

Shashkov, M. (1995), *Conservative Finite-Difference Methods on General Grids*, CRC Press, Boca Raton, Florida.

Shashkov, M. and Steinberg, S. (1995), "Support-Operator Finite-Difference Algorithms for General Elliptic Problems," *Journal of Computational Physics*, Vol. 118, p. 131.

Shashkov, M. and Steinberg, S. (1996), "Solving Diffusion Equations with Rough Coefficients in Rough Grids," *Journal of Computational Physics*, Vol. 129, p. 383–405.

Shih, T. M., Tan, C. H., and Hwang, B. C., "Effects of Grid Staggering on Numerical Schemes," *International Journal for Numerical Methods in Fluids*, Vol. 9, No. 2, 1989, pp. 193–212.

Shirazi, S. A., and Truman, C. R., "Evaluation of Algebraic Turbulence Models for PNS Predictions of Supersonic Flow Past a Sphere-Cone," *AIAA Journal*, Vol. 27, No. 5, 1989, pp. 560–568.

Sickles, W. L. and Erickson, J. C. Jr. (1990), "Wall Interference Corrections for Three-Dimensional Flows," AIAA Paper 90–1408, June 1990.

Sinclair, G. B., Anaya-Dufresne, M., Meda, G, and Okajima, M. (1997), "Tuned Test Problems for Numerical Methods in Engineering," *International Journal for Numerical Methods in Engineering*, Vol. 40, pp. 4183-4209.

Sinclair, G. B., Beisheim, J. R., and Sezer, S. (2006), "Practical Convergence-Divergence Checks for Stresses from FEA," Proc. 2006 International ANSYS Users Conference and Exposition, 2-4 May 2006, Pittsburgh, PA, U.S.A.

Sinclair, G. B. (2006), personal communication.

Sindir, M. M., Barson, S. L., Chan, D. C., and Lin, W. H. (1996), "On the Development and Demonstration of a Code Validation Process for Industrial Applications," AIAA Paper 96–2032, 27 AIAA Fluid Dynamics Conference, New Orleans, Louisiana, June 17–20, 1996.

Somerville, R. C. J. (1996), *The Forgiving Air: Understanding Environmental Change*, University of California Press, Berkeley.

Stauffer, T. B., Antworth, C. P., Young, R. G., MacIntyre, W. G., Boggs, J. M., and Beard, L. M. (1994), "Degradation of Aromatic Hydrocarbons in an Aquifer During a Field Experiment Demonstrating the Feasibility of Remediation by Natural Attenuation," U. S. Air Force, Armstrong Laboratory, AL/EQ–TR–1993–0007, 1994.

Steele, W. G., Maciejewski, P. K., James, C. A., Taylor, R. P., and Coleman, H. W. (1996), "Asymmetric Systematic Uncertainties in the Determination of Experimental Uncertainty," *AIAA Journal*, Vol. 34, No. 7, July 1996, pp. 1458–1463.

Steffen, C. J., Reddy, D. R., and Zaman, K. B. M. Q. (1995), "Analysis of Flowfield from a Rectangular Nozzle with Delta Tabs," AIAA Paper 95–2146, 26th AIAA Fluid Dynamics Conference, San Diego, California, June 19–22, 1995.

Steger, J. L. and Sorenson, R. L. (1979), "Automatic Mesh–Point Clustering Near a Boundary in Grid Generation with Elliptic Partial Differential Equations," *Journal of Computational Physics*, Vol. 33, No. 3, December 1979, pp. 405–410.

Steinberg, S. and Roache, P. J. (1985), "Symbolic Manipulation and Computational Fluid Dynamics," *Journal of Computational Physics*, Vol. 57, No. 2, January 1985, pp. 251–284.

Steinberg, S. and Roache, P. J. (1986a), "A Tool Kit of Symbolic Manipulation Programs for Variational Grid Generation," AIAA Paper No. 86–0241, AIAA 24th Aerospace Sciences Meeting, Reno, Nevada, 6–9 January 1986.

Steinberg, S. and Roache, P. J. (1986b), "Using MACSYMA to Write FORTRAN Subroutines," *Journal of Symbolic Computation*, Vol. 2, pp. 213–216.

Steinberg, S. and Roache, P. J. (1990), "Symmetric Operators in General Coordinates," Ecodynamics Research Associates, Albuquerque, New Mexico.

Steinberg, S. (1997), personal communication.

Steinbrenner, J. P. and Chawner, J. R. (1992), "The GRIDGEN Version 8 Multiple Block Grid Generation Software," MDA Engineering Report 92–01, 1992.

Stephens, A. B and Shubin, G. R. (1981), "Multiple Solutions and Bifurcation of Finite Difference Approximations to Some Steady Problems in Fluid Dynamics," *SIAM Journal of Scientific and Statistical Computing*, Vol. 2, No. 4, 1981, pp. 404–415.

Stern, F., Wilson R. V., Coleman, H. W., and Paterson, E. G. (1999), "Verification and Validation of CFD Simulations," IIHR Report No. 407, September 1999.

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

Stern, F., Wilson, R. and Shao, J. (2004), "Quantitative V&V of CFD Simulations and Certification of CFD Codes with Examples," *Proceedings of CHT-04, ICHMT International Symposium on Advances in Computational Heat Transfer*, April 19-24, 2004, Norway, CHT-04-V1.

Straka, J. M., Wilhelmson, R. B., Wicker, L. J., Anderson, J. R. and Droegemeier, K. K. (1993), "Numerical Solutions of a Non-Linear Density Current: A Benchmark Solution and Comparisons," *International Journal for Numerical Methods in Engineering*, Vol. 17, 1993, pp. 1–22.

Strazisar, A. J. and Denton, J. D. (1995), "CFD Code Assessment in Turbomachinery–A Progress Report," *ASME Global Gas Turbine News*, May/June 1995, pp. 12–14.

Strouboulis, T. and Oden, J. T. (1990), "A Posteriori Error Estimation of Finite Element Approximations in Fluid Mechanics," *Computer Methods in Applied Mechanics and Engineering*, Vol. 78, pp. 201–242.

Sudicky, E. A. and Frind, E. O. (1982), "Contaminant Transport in Fractured Porous Media: Analytical Solutions for a System of Parallel Fractures," *Water Resources Research*, Vol. 18, No. 6, December 1982, pp. 1634–1642.

Tang, D. H., Frind, E. O., and Sudicky, E. A. (1981), "Contaminant Transport in Fractured Porous Media: Analytical Solution for a Single Fracture," *Water Resources Research*, Vol. 17, No. 3, 1981, pp. 555–564.

Taylor, B. N. and Kuyatt, C. E. (1994), *Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results*, NIST TN 1297, website of the National Institute for Standards and Testing. http://physics.nist.gov/Pubs/guidelines/TN1297/tn1297s.pdf

Telionis, D. P. (1995), "Editorial," *ASME Journal of Fluids Engineering*, Vol. 117, September 1995, pp. 329–330.

Thareja, R. R., Stewart, J. R., Hassan, O., Morgan, K., and Peraire, J. (1989), "A Point Implicit Unstructured Grid Solver for the Euler and Navier-Stokes Equations," *International Journal for Numerical Methods in Fluids*, Vol. 9, 1989, pp. 405–425.

Thibault, P. (1994), "Code Validation: A First Step Towards Quality Assurance for CFD Software," *CFD Bulletin*, CFD Society of Canada, Winter 1994, p. 6.

Thibault, P. A. and Bergeron, D. M. (1995), Eds., *Proceedings of the Third Annual Conference of the CFD Society of Canada*, Volumes I and II, June 25–27, 1995, Banff, Alberta.

Thomas, P. D. and Middlecoff, J. F. (1980), "Direct Control of the Grid Point Distribution in Meshes Generated by Elliptic Equations," *AIAA Journal*, Vol. 18, No. 6, June 1980, pp. 652–656.

Thomas, J. and Salas, M. (1985), "Far-Field Boundary Conditions for Transonic Lifting Solutions to the Euler Equations," AIAA Paper 85–0020.

Thompson, J. F., Thames, F. C., and Mastin, C. W. (1974), "Automatic Numerical Generation of Body–Fitted Curvilinear Coordinate System for Field Containing Any Number of Arbitrary Two–Dimensional Bodies," *Journal of Computational Physics*, Vol. 15, No. 3, July 1974, pp. 299–319.

Thompson, J. F., Warsi, Z. U. A., and Mastin, C. W. (1985), *Numerical Grid Generation Foundations and Applications*, North-Holland, New York.

Tjonneland, T. (1988), in Round Table Discussion in AGARD (1988).

Truman, C. R. and Lee, M. J. (1990), "Effects of Organized Turbulence Structures on the Phase Distortion in a Coherent Optical Beam Propagating through a Turbulent Shear Flow," *Physics of Fluids A*, Vol. 2, No. 5, 1990, pp. 851–857.

Tsang, C.-F. (1991), "The Modeling Process and Model Validation," *Ground Water*, Vol. 29, No. 6, November–December 1991, pp. 825–831.

Turgeon, E., Pelletier, D. and Ignat, L. (2000), "Effects of Adaptivity on Finite Element Schemes for Turbulent Heat Transfer and Flow Predictions," *Numerical Heat Transfer, Part A*, Vol. 38, pp. 847-868, 2000.

Turgeon, E. and Pelletier, D. (2002), "A General Continuous Sensitivity Equation Formulation for Complex Flows," *Numerical Heat Transfer, Part B*, Vol. 42, pp. 485-498, 2002.

Underwood, F. N. and Mueller, T. J. (1977), "Numerical Study of the Steady Axisymmetric Flow Through a Disk-Type Prosthetic Heart Valve in a Constant Diameter Chamber," *ASME Journal of Biomechanical Engineering*, Vol. 99, May 1977, pp. 91–97.

V&V1, see Roache (1998a)

V&V10, see ASME Committee PTC-60 (2006)

V&V20, see ASME Committee PTC-61 (2009)

Van Dyke, M. (1958), "The Supersonic Blunt Body Problem–Review and Extension," *Journal of the Aeronautical Sciences*, Vol. 25, p. 485.

Van Dyke, M. (1975), *Perturbation Methods in Fluid Dynamics*, Parabolic Press, Stanford, California.

Van Gulick, P. (1994), personal communication.

Van Leer, B. (1979), "Towards the Ultimate Conservative Difference Scheme. V. A Second-Order Sequel to Godunov's Method," *Journal of Computational Physics*, Vol. 32, 1979, pp. 101–136.

Van Straalen, B. P., Simpson, R. B., and Stubley, G. D. (1995), "A Posteriori Error Estimation for Finite Volume Simulations of Fluid Flow Transport," *Proceedings of the Third Annual Conference of the CFD Society of Canada*, Banff, Alberta, P. A. Thibault and D. M. Bergeron, Eds., Vol. I, 25–27 June 1995.

Van Wie, D. M. and Rice, T. (1996), "Quantification of Data Uncertainties and Validation of CFD Results in the Development of Hypersonic Airbreathing Engines," AIAA Paper 96–2028, 27th AIAA Fluid Dynamics Conference, New Orleans, Louisiana, June 17–20, 1996.

Vanka, P. (1995), "Comment on 'Toward a Numerical Error Bar in CFD'," *ASME Journal of Fluids Engineering*, Vol. 117, No. 1, 1995, pp. 9–10.

Vasilyev, O. V., Paolucci, S. and Sen, M. (1995), "A Multilevel Wavelet Collocation Method for Solving Partial Differential Equations in a Finite Domain," *Journal of Computational Physics*, Vol. 120, 1995, pp. 33–47.

Vasilyev, O. V. and Paolucci, S. (1996), "A Dynamically Adaptive Multilevel Wavelet Collocation Method for Solving Partial Differential Equations in a Finite Domain," *Journal of Computational Physics*, Vol. 125, 1996, pp. 498–512.

Vassberg, J.C., et al (2007), "Summary of DLR-F6 Wing-Body Data from the Third AIAA CFD Drag Prediction Workshop," Paper 57, NATO RTO-MP-AVT-147 Workshop, December 2007.

Vassberg, J. C., et al (2008a), "Abridged Summary of the Third AIAA Computational Fluid Dynamics Drag Prediction Workshop," *Journal of Aircraft*, Vol. 45, No. 3, May-June 2008.

Vassberg, J. C., et al (2008b), "Comparison of NTF Experimental Data with CFD Predictions from the Third AIAA CFD Drag Prediction Workshop," AIAA-2008-6918, 26th AIAA Applied Aerodynamics Conference, Honolulu, Hawaii, Aug. 18-21, 2008.

Viegas, J. R., Rubesin, M. W., and MacCormack, R. W (1988), "On the Validation of a Code and a Turbulence Model Appropriate to Circulation Controlled Airfoils," in AGARD (1988), Paper No. 6.

von Mises, R. (1957), *Probability, Statistics and Truth*, 2nd Revised English Edition, Dover, NY.

Vos, J. B., Rizzi, A., Darracq, D., and Hirschel, E. H. (2002), "Navier-Stokes Solvers in European Aircraft Design," *Progress in Aerospace Sciences*, Vol. 38, No. 8, Pp. 601-697.

Wagenmakers, E.-J., Lee, M., Lodewyckx, T., and Iverson, G., *Bayesian Versus Frequentist Inference*, Ch. 9 in *Bayesian Evaluation of Informative Hypotheses*, H. Hoijtink, I. Klugkist and P. A. Boelen, eds., Springer, New York.

Walker, M. A. and Oberkampf, W. L. (1992), "Joint Computational/Experimental Aerodynamics Research on a Hypersonic Vehicle: Part 2, Computational Results," *AIAA Journal*, Vol. 30, No. 8, August 1992, pp. 2010–2116.

Wang, C. Y. (1991), "Exact Solutions of the Steady-State Navier-Stokes Equations," *Annual Review of Fluid Mechanics*, Vol. 23, 1991, pp. 159–177.

Wang, S. S. Y. and Jia, Y. (2009), "Mathematical Verification Using Prescribed or Manufactured Solutions," Ch. 4 of ASCE/EWRI (2009).

Wang, S. S. Y., Jia, Y., Roache, P. J., Smith, P. E., and Schmalz, R. A. Jr., eds. (2009), *Verification and Validation of 3D Free-Surface Flow Model*, ASCE/EWRI Task Committee.

Weiss, E. H. (1983), "Why We Still Have So Little Technical Documentation," *Infosystems*, June 1983, pp. 88–89.

Westerink, J. J., K. D. Stolzenbach and J. J. Connor (1989), "General Spectral Computations of the Non-Linear Shallow Water Tidal Interactions Within the Bight of Abaco," *Journal of Physical Oceanography*. Vol. 19, 1989, pp. 1350–1373.

Westerink, J. J., R. A. Luettich, and J. C. Muccino (1994), "Modeling Tides in the Western North Atlantic Using Unstructured Graded Grids," *Tellus*, Vol. 46A, 1994, pp. 187–199.

Westerink J. J. and Roache P. J. (1995), "Issues in Convergence Studies in Geophysical Flow Computations," in Johnson and Hughes (1995), pp. 1–6.

Westerink, J. J., R. A. Luettich, C. A. Blain, and S. C. Hagen (1995), "The Utility of the Finite Element Method in Computing Surface Elevation and Circulation in Continental Margin Waters," in *Finite Element Modeling of Environmental Problems*, G. F. Carey, Ed., John Wiley and Sons, Chichester, pp. 39–59.

Wilcox, D. C. (1993), *Turbulence Modeling for CFD*, DCW Industries, La Cañada, California.

Wilcox, D. C. (1994), personal communication.

Wilcox, D. C. (1995), *Perturbation Methods in the Computer Age*, DCW Industries, La Cañada, California.

Wilcox, D. C. (1995), personal communication.

Wilcox, D. C. (2006), *Turbulence Modeling for CFD*, Third Edition, DCW Industries, La Canada, CA.

Willhite, P., Jasper, D., and Romer, W. (1995), "A Critical Evaluation of CFD Predictions of Full Aircraft Drag Increments," AIAA Paper 95–2289, June 1995.

Williams, M. (1989), "A Truncation Error Elimination Method for Steady-State Navier-Stokes Calculations," AIAA Paper 89–1934, 1989.

Williams, P. T. and Baker, A. J. (1996), "Incompressible Computational Fluid Dynamics and the Continuity Constraint Method for the Three-Dimensional Navier-Stokes Equations," *Numerical Heat Transfer, Part B*, Vol. 29, 1996, pp. 137–273.

Wilson, R. V., Stern, F., Coleman, H. W., and Paterson, E. G. (2001), "Comprehensive Approach to Verification and Validation of CFD Simulations - Part 2: Application for RANS Simulation of a Cargo/Container Ship", *ASME Journal of Fluids Engineering*, Vol. 123, pp. 803-810. December 2001.

Wilson, R. and Stern, F. (2002), "Verification and Validation for RANS Simulation of a Naval Surface Combatant," AIAA Paper 2002-0904, AIAA 40th Aerospace Sciences Meeting, 14-17 January 2002, Reno, Nevada.

Wilson, R., Shao, J., and Stern, F. (2004), "Discussion: Criticisms of the 'Correction Factor' Verification Method," *ASME Journal of Fluids Engineering*, Vol. 126, pp. 704-706.

Winslow, A. (1967), "Numerical Solution of the Quasilinear Poisson Equations in a Nonuniform Triangle Mesh," *Journal of Computational Physics*, Vol. 2, pp. 149–172.

WIPP PA Dept. (1992), *Annual Performance Assessment for the Waste Isolation Pilot Plant, Dec. 1992, Vol. 1, Preliminary Comparison with 40 CFR 191, Subpart B, Vol. 2: Technical Basis*, Performance Assessment Department, Sandia National Laboratories, Albuquerque, New Mexico.

Wood, W. A. and Kleb, W. L. (2002), "Extreme Programming in a Research Environment," in *Extreme Programming and Agile Methods* pp. 89-99, D. Wells and L. Williams, eds., Vol. 2418,

Lecture Notes in Computer Science, Springer-Verlag, New York, 2002.

Wu, J., Zhu, J. Z., Szmelter, J. and Zienkiewicz, O. C. (1990), "Error Estimation and Adaptivity in Navier-Stokes Incompressible Flows," *Computational Mechanics*, Vol. 6, 1990, pp. 259–270.

Xing, T., Carrica, P., Stern, F. (2008), "Computational Towing Tank Procedures for Single Run Curves of Resistance and Propulsion," *ASME Journal of Fluids Engineering*, Vol. 130, 101102, pp. 1-14.

Xing, T. and Stern, F. (2008), "Factors of Safety for Richardson Extrapolation for Industrial Applications," IIHR Report No. 466, September 2008.

Xing, T. and Stern, F. (2009), "Factors of Safety for Richardson Extrapolation," IIHR Technical Report No. 469, IIHR - Hydroscience & Engineering, University of Iowa, March 2009.

Yee, H. C., Sweby, P. K., and Griffiths, D. F. (1991), "Dynamical Approach Study of Spurious Steady-State Numerical Solutions of Nonlinear Differential Equations. I. The Dynamics of Time Discretization and its Implications for Algorithm Development in Computational Fluid Dynamics," *Journal of Computational Physics*, Vol. 97, 1991, pp. 249–310.

Yee, H. C. (1992), personal communication.

Yee, H. C. and Sweby, P. K. (1996), "Some Aspects of Numerical Uncertainties in Time-Marching to Steady-State Numerical Solutions," AIAA Paper 96–2052, 27th AIAA Fluid Dynamics Conference, New Orleans, Louisiana, June 17–20, 1996.

Yee, H. C. and Sjogreen, B. (2006), "Simulation of Richtmyer-Meshkov Instability by Sixth-Order Filter Methods," ISIS17 Symposium, 4-8 September 2006, University of Rome, Italy.

Yee, H. C. (2007), personal communication. 12 February 2007.

Yee, H. C. and Sjogreen, B. (2007), "Simulation of Richtmyer-Meshkov Instability by Sixth-Order Filter Methods," *Shock Waves*, 17:185-193.

Zha, G-C. and Knight, D. (1996), "Three-Dimensional Shock/Boundary-Layer Interaction Using Reynolds Stress Equation Turbulence Model," *AIAA Journal*, Vol. 34, No. 7, July 1996, pp. 1313–1321.

Zhu, J. Z. and Zienkiewicz, O. C. (1990), "Superconvergence Recovery Technique and *A Posteriori* Error Estimates," *International Journal for Numerical Methods in Engineering*, Vol. 30, 1990, pp. 1321–1339.

Zienkiewicz, O. C. and Zhu, J. Z. (1987), "A Simple Error Estimator and Adaptive Procedure for Practical Engineering Analysis," *International Journal for Numerical Methods in Engineering*, Vol. 24, 1987, pp. 337–357.

Zienkiewicz, O. C. and Zhu, J. Z. (1992), "The Superconvergent Patch Recovery and *A Posteriori* Error Estimates. Part I: The Recovery Technique," *International Journal for Numerical Methods in Engineering*, Vol. 33, 1992, pp. 1331–1364.

Zingg, D. W. (1991), "Viscous Airfoil Computations Using Richardson Extrapolation," AIAA Paper 91–1559–CP, AIAA 30th Aerospace Sciences Meeting, Reno, Nevada, January 6–9, 1992.

Zingg, D. W. (1992), "Grid Studies for Thin-Layer Navier-Stokes Computations of Airfoil Flowfield," *AIAA Journal*, Vol. 30, No. 10, October 1992, pp. 2561–2564.

APPENDIX A.

NEED FOR CONTROL OF NUMERICAL ACCURACY

The following is a reproduction of Roache, P. J. (1990), "Need for Control of Numerical Accuracy," AIAA *Journal of Spacecraft and Rockets*, Vol. 27, No. 2, March–April 1990, pp. 98–102. See also AIAA Paper No. 89–1669.

Need for Control of Numerical Accuracy[*]

Patrick J. Roache
Ecodynamics Research Associates, Albuquerque, New Mexico 87198

Abstract

The need for the control of numerical accuracy in computational fluid dynamics (CFD) code solutions is reviewed in the light of current journal practice and experience with implementation of an editorial policy on the same subject published by the *Journal of Fluids Engineering*. Various actual objections to that

[*] Presented as Paper 89-1669 at the 24th AIAA Thermophysics Conference, Buffalo, NY, June 12-14, 1989; Received July 8, 1989.
[*] President. Senior Member AIAA.

policy are listed and responses are given. The general successes and particular difficulties experienced in the implementation of the policy are noted. The broader question of code verification, validation, and certification is considered. It is suggested that professional societies such as the AIAA and American Society of Mechanical Engineers may ultimately become involved in the task of certification of available CFD commercial codes.

I. Introduction

This paper, on the general philosophy and need for numerical accuracy control in computational fluid dynamics (CFD) codes, is based on experience with the implementation of an editorial policy statement by the American Society of Mechanical Engineers (ASME), published in the *Journal of Fluids Engineering* (*JFE*) [1]. The policy statement was conceived following the creation of the position of Associate Editor for Numerical Methods in the *JFE,* which formally recognized the special needs of this discipline. The ASME policy and supporting statements are reproduced in the Appendix. The rationale and needs for the policy statement are explained in the announcement. The *JFE* had previously published and had many years of experience with a similar requirement for uncertainty analysis in experimental papers. Following our early experience with this new policy, a similar policy was also adopted by the ASME's *Journal of Heat Transfer*. The general subject of control of numerical accuracy has become something of a "hot topic," with special reference to the National Aero-Space Plane, a session at the AIAA 1989 Thermophysics Conference, ASME sessions at the 1988 and 1989 Winter Annual Meetings, and by the Texas Institute for Computational Mechanics workshop on the slightly broader topic of reliability in computational mechanics in October 1989.

II. Resistance and Objections

Although Roache et al [1] thought that the policy statement and the editorial requirement were quite mild, it was not universally welcomed. Objections were offered by some of the other editorial board members and by other members of the professional community from whom I solicited comments in the months following the publication of the statement. Some of these objections, all of which are actual (i.e., not straw-man objections), are listed below, together with my responses.

Objection 1

It is too expensive of computer time to do mesh doubling calculations in order to ascertain grid convergence.

There are, of course, other ways to ascertain grid convergence than the straightforward method of grid doubling. In fact, this is probably the most reliable method available, but there are other approaches, as briefly touched upon in the original policy statement. If the cost of computer resources is not a problem to the researcher, this is certainly the easiest approach to take. However, if computer resources are a problem, there are other methods that are not intensive users of computer time. It seems that the greater objection for doing grid convergence studies is the fact that it requires a bit of conscientious work on the part of researchers.

Also, if it is argued that it is simply too expensive to do any kind of control of numerical accuracy, then I would argue that the author simply cannot be in this CFD business. After all, if you do not have a wind tunnel you cannot do experimental testing. My impression of the situation is actually worse than this. Journal articles in the late 1960s and early 1970s commonly predicted high resolution accuracy runs when the next generation of computers became available. But for the most part they have not been used that way.

Generally the tremendous advance of computing power has been used to produce more mediocre papers rather than fewer reliable ones.

Objection 2

Some exception to the policy should be made for expensive calculations, particularly three-dimensional turbulent studies.

I do not think that the overall cost of the computations should be a consideration. It seems clear that the incremental cost of performing a grid convergence test should be normalized by the cost of the base case. In this sense it is cheaper to validate the grid convergence on a three-dimensional problem than on a two-dimensional problem, presuming that the incremental work involves an extra coarse grid computation. This again relates to the first point, which states that it is not necessary to do a grid doubling in order to ascertain some index of numerical accuracy. A grid halving is also appropriate. Of course, the advantage lies in doing a grid doubling test because the error bounds will be sharper.

Objection 3

Turbulence modeling, rather than the numerical solution of the partial differential equations, is the real determinator of accuracy.

My response is, yes and no. Accuracy is a question to be addressed even for laminar flow calculations, wherein the constitutive equations are not in doubt. The discretization error does not disappear just because one uses a turbulence model! Our point in the *JFE* policy statement, and the first criticism which our evaluation committee made at the 1980/81 Stanford meeting on complex turbulence flows [2], is that one cannot evaluate different turbulence models unless one first satisfies grid convergence. There are yet more considerations in the overall accuracy question; including, for example, the attainment of a true steady state in the computations (yet another "convergence"), inner-loop convergence for incompressible flows, equation of state accuracy, low Mach number approximation, geometry representation, accuracy of viscosity and conductivity coefficients, constant Prandtl number assumptions, chemical reaction rates, and so forth. And, of course, coding errors! These all affect accuracy but do not remove or over-ride the requirement for grid convergence testing. I am of the opinion that these modeling questions, including turbulence modeling, should be kept separate from the question of the numerical accuracy of the solution to those mathematical models.

However, there is in fact another special problem with turbulence modeling in that both the fine and coarse grids have to get some points into the viscous sublayer. I do not know any easy way around this or any other problem that generates such a range of significant length scales which must be adequately resolved. However, thorough work can be done. See, for example, Ref. 3.

Objection 4

The policy statement does not go far enough.

This is certainly true. We consider the *JFE* policy statement to be a minimalist statement. The idea is to make it clear that the authors have to address the topic of numerical accuracy control, and to give the editors and the referees the support they need in demanding some effort in this regard. For example, in the policy statement, no mention is made of the control of errors due to far-field boundary conditions. Cheng [4] pointed out two decades ago that this computational modeling error does not improve as the grid is refined. The only way to test for its effect is to move the position of the far-field boundary. Likewise, the simple reporting of, for example, a 5 or 10% difference in some function of the solution between two grids is not really a rigorous indicator that the solution has been obtained to 5 or 10% accuracy. Although claims

have been made that the finite-element methodology offers a more rigorous approach to error estimation, I have not been convinced that anything useful and dependable has been produced for nonlinear systems of partial differential equations. (The theoretical work that has been done is interesting to some extent, but it is usually limited to a very simple model equation.) Schonauer [5] has devoted much time and effort to producing accurate error estimators for Navier-Stokes codes. In fact, his goal is not merely to produce accurate solutions but to accurately predict the error bounds on those solutions. I suppose a really tough journal policy or certification policy would require such effort before accepting a solution, but the journal policy that we adopted was much more lax than this. The only intent was to avoid the syndrome of producing a single calculation on a single grid (the all too common "take it or leave it" attitude).

It is important to note here that grid convergence is really only one aspect of numerical accuracy control; it is, however, a necessary component. For example, a statement of results on grid convergence studies really has nothing to do with coding errors (unless an exact solution is known, as will be discussed subsequently).

Mehta [6] is concerned with code validation, verification, and certification. Note that a *code* is certified, but a particular calculation still needs to be examined for the control of numerical accuracy. A wrong code can "converge" and, conversely, a correct (i.e., certified) code can be applied to a particular problem with an inadequate resolution. Therefore the question of grid convergence testing really presumes ahead of time that we have a correct code. The only question then remaining is whether or not the code has been applied with adequate resolution to obtain an accurate answer to the problem at hand.

Common sense and experience certainly indicate that this presumption of code correctness is not always justified. There are now tools available to perform this kind of verification quite convincingly, in my opinion. In Ref. 7, we showed how to verify a FORTRAN code that was produced entirely by a symbolic manipulation code. The idea is to generate a selected analytic solution to a problem by introducing forcing terms and then to monitor the convergence of the CFD code to that solution as the grid size is reduced. This procedure verifies the coding accuracy, the grid transformation equations, and even the order of accuracy of numerical method. In Ref. 7, it was performed only for the Poisson equation in a transformed three-dimensional coordinate system and for the most common elliptic grid generation equations. Shih et al [8] have applied this concept to the full incompressible Navier-Stokes equations in two-dimensions. I consider this paper to be one of fundamental importance in the area of code verification. It would seem that the principle can readily be extended to compressible and three-dimensional flows using symbolic manipulation programs such as MACSYMA. Turbulence models also appear to be amenable to this approach of accuracy checking-noting of course that we are not talking about the adequacy of the turbulent representation of the real physics but rather the correct coding and the numerical solution of the turbulence model. Shocks, however, seem to be a more difficult topic to approach in this manner.

Note that there is an earlier history of using very simple solutions to verify models, but false indicators of accuracy sometimes have been obtained. The key in such an exercise is to choose a solution with enough structure in it to exercise all of the terms in the equations and all of the leading error terms in the discretization. If, for example, one selects a Couette flow solution, one will be lead to the erroneous conclusion that a first-order accurate method is perfectly adequate even with a coarse-grid resolution. The power of the Shih et al method [8] is that the particular solution they have constructed behaves like a real fluid dynamics problem with boundary layers and significant solution structure.

Objection 5

First-order methods and hybrid methods are not more difficult to judge for convergence than second-order methods, contrary to the discussion in the policy statement.

I do not understand this objection. It is an elementary behavior of numerical methods that higher order accurate methods, as they approach convergence, do so more quickly. For example, a fourth-order Runge-Kutta method applied to an ordinary differential equation is easier to judge as being converged than is a second-order method. If a paper reports that a successive grid doubling of a solution of a linearized advection-diffusion equation produces only a 5% change in the answers, one could expect with some confidence that the fine grid solution is indeed within 5% of the true solution, if the method used is fourth-order accurate. If the method used is second-order accurate, it is more problematical; if the method used is only first-order accurate, I would be skeptical of the 5% limit on the "accuracy."

{*Addition to the original JSR paper*. In view of my later analysis in the paper on the Grid Convergence Index (Roache, 1994 cited in Chapter 5) I am now more optimistic, especially for a grid doubling with a second order method. However, the basic point, that reliable grid convergence is more difficult to ascertain with lower order methods, is still valid.}

Objection 6

Agreement of the calculations with experiment is enough justification for the solution accuracy, without any need for doing systematic grid convergence testing.

This objection is quite attractive at first glance. In practice, it does not seem to work very often because the agreement with the experiment is usually not universal. This is especially obvious (as stated in the original policy statement) with turbulence modeling. Whether or not the turbulence parameters have been "tuned" to a particular problem, we still have a "package deal" of discretization errors and turbulence modeling errors. The discretization errors should be separated from the turbulence modeling errors. If constitutive equations are not an issue, and if the agreement with a very good experiment is complete, I suppose I would have to relent. But I would still maintain that a *better* paper would result if two grids or some other grid convergence testing were used. For one thing, it would give some idea of how difficult the problem is numerically. (For example, perhaps the results presented were on 100×100 and a 30×30 grids might have been sufficient.) Also, such an exercise may help to verify the code, and would also verify the order of convergence of the code. As Blottner [9] demonstrated two decades ago, plausible second-order, boundary-layer approximations in fact do not always behave in a second-order manner, and this rate of convergence can be established by a systematic grid refinement testing.

Objection 7

I do not have any "truncation error" in my solution since I use finite-element methods (FEM).

Sure you do! (Whether or not the Taylor series is used in the *derivation* of a discrete method, it can still be used in an *analysis* of the method.) But it would have been better in the *JFE* policy statement if we had used the more general term of "discretization error." (Also, "truncation error" strictly speaking is not applicable in the presence of discontinuities.) I would also note that the commonly referred to FEM "error evaluation" practice of substituting the basis function and the solution values into the original partial differential equation is perhaps a valid index of discretization error, but it is not absolutely the error evaluation that we ultimately want. That is, this procedure does not tell us the difference between our discrete solution and the exact solution.

Objection 8

It should not be necessary to legislate such a requirement for doing conscientious work.

It should not have been necessary, but it was. The need is indicative of the problem; that is always the case for legislation, rules, formal ethics committees, and so forth. If a 25th century historian were to read

through Albuquerque newspapers from 1989 he would find Sunwest Bank advertising that it has been "Safe, Strong, and Sound for 65 Years." The historian would rightly conclude that the U. S. banking industry in 1989 was in trouble. He would also read a lot about congressional ethics committees and rightly conclude that political ethics were somewhat lacking. And if he read our JFE policy statement on the control of numerical accuracy, he would rightly conclude that there was a bit of sloppy work going on. As Professor George Raithby wrote to me after Ref. 1 was published, "The need to legislate that authors check accuracy is sad, but the enforced discipline will benefit the field."

III. Difficulties in Applications

Generally speaking, the response from the reviewers has been positive. Also, the authors have been willing to perform the extra work required. The existence of the policy statement has helped since both the reviewers and the editors can refer to the policy statement and thus avoid rehashing the arguments with the authors. However, there were some particular difficulties.

1. The policy statement was phrased in such a way as to allow the editor to reject papers outright (without bothering reviewers) for failure to address the numerical accuracy issue. In practice, this seemed too severe and would have led to excessive publication lag since it would have put the numerical accuracy review/response in series with the other communications. Consequently, I often did initiate the review process even on papers that clearly did not abide by the policy. This also made it easier to be firm with the authors later, since I had the additional moral support of the reviewer's comments.
2. Some authors chose to abide by the letter of the law in the policy statement with generic statements that could be inserted into any paper, such as "the results from different grid resolutions were compared, and a 13×13 grid was shown to be adequate." Although our policy statement was not very demanding, I did reject such "trust me" statements and insisted on some quantifiable grid convergence results. As someone put it, "Convergence lies in the eyes of the beholder." I agree, but that is exactly the point of requiring some quantifiable criterion; it allows the reader to decide for himself. It would be naive in the extreme to think that there is any consensus agreement on what constitutes adequate convergence testing. We were not requiring a priori any particular method or any particular quantitative measure of convergence, but it is necessary to supply the reader with some *numbers*. Note that compliance with this requirement does not necessarily guarantee acceptance of the paper; it is still up to the reviewers and editors to exercise their own professional judgment on the adequacy of the quantified convergence criteria.
3. Some authors would prefer to refer to other papers on similar problems for the grid convergence test. This approach is legitimate in theory, and sometimes in practice, but for reasons noted in the previous paragraph, it is difficult and problematical. It is certainly preferable to have every paper self-contained in this regard.

Some of the papers submitted and accepted did *not* display very convincing convergence results. This again alludes to the difference of opinions mentioned previously, and it is a difficult subject. During my tenure as associate editor for *JFE, I* was quite liberal in accepting papers as long as the grid convergence numbers were provided so that the reader could judge for him- or herself. There are good and bad, sensitive and insensitive indicators of convergence. As noted earlier, it is possible to be fooled by only looking at a neighboring problem. An obvious example is that of Poiseuille flow. If one tries to answer the question of "how many points are required to be in a boundary layer on an aerodynamics problem" by considering the neighboring problem of two-dimensional Poiseuille flow, one quickly comes to the conclusion that a single point in the boundary layer gives perfect accuracy! The obvious reason is that the solution to two-

dimensional Poiseuille flow is a parabola, and therefore any second-order finite-difference method will give the exact answer.

Likewise, one can erroneously conclude that a uniform mesh is optimal [10]. A not-so-obvious bad indicator of grid convergence is the vortex wake solution. Many authors have "pointed with pride" to their quite accurate predictions (compared to experiment) of the Strouhal number for the near wake of a circular cylinder. It is apparently not very widely recognized that this is an easy calculation, because the frequency of the von Kármán vortex wake is essentially an inviscid phenomena, and therefore the accuracy of its prediction is not a good indicator of the accuracy of predicted boundary layer heat transfer, friction drag, or wake decay.

IV. Examples of What Can Be Done

I would like to point out a few exemplary publications addressing the question of the control of numerical accuracy. I have already mentioned the turbulence calculations of Shirazi and Truman [3], Schonauer's ambitious approach [5], and the work of Shih et al [8] in which both the accuracy of the code (i.e., freedom from coding errors) and its second-order convergence for a particular problem are convincingly demonstrated. Another paper, by Kuruvila and Anderson," is nicely illustrative of the difficulties and pitfalls of doing convergence studies with artificial dissipation terms in the equations. The paper by Fujii et al [12] was illuminating in demonstrating the importance of grid resolution for Euler equation solutions to the leading edge separation problem. Thareja et al [13] present solution-adaptive, nonstructured finite-element solutions of the supersonic blunt body problem, which include, in the methodology, the control of numerical accuracy. Dietrich et al [14] presented systematic grid truncation error testing of ocean circulation codes with four different methods. This procedure is so rare in the geophysical community as to be virtually nonexistent. The paper by Durst and Pereira [15] demonstrated that the entire procedure is readily applicable to time dependent problems as well as steady state problems. Blottner's paper [16] demonstrated his thorough tests for hypersonic nose tips. Nguyen and Maclaine-Cross [17] use Richardson extrapolation to zero mesh size to produce reliable curve fits to incremental pressure drop number in heat exchangers from full Navier-Stokes solutions. Finally, any such list, no matter how fragmentary, must include the classic study of the natural convection benchmark problem by de Vahl Davis [18].

V. Conclusions and Recommendations

The control of numerical accuracy as addressed in the *Journal of Fluids Engineering* policy statement is a necessary but not sufficient component of the broader problem of code validation and verification. It is important to realize, however, that a code can be validated and verified and may be indeed certifiable as "error free," but this does not obviate the need for systematic grid convergence testing (or other systematic, quantitative error estimation) in any new application.

How far should we go? In research journals, I think a broad application of some numerical accuracy control philosophy would do much to improve the quality of published work. (Also, by putting a burden of more work on researchers, it would probably cut down on the number of papers. I, for one, would shed no tears.) In the long run, I would also be in favor of certification of CFD commercial codes. The American National Standards Institute and the American Society of Testing Materials are already involved in such certification activities in many areas, including FORTRAN compilers. I have some doubts, however, regarding the ability of such organizations to handle certification for complex CFD codes. It is probably worthwhile to consider now that AIAA and ASME may be active in the area of CFD code certification (and perhaps *de*certification) in the future. I hasten to add that this is *not* the kind of work that I would

personally enjoy or am qualified to do. But, like it or not, I can envision seeing, in my lifetime, CFD code certification by professional societies or joint committees thereof. Note that CFD code certification was first suggested by Mehta [19].

I do not particularly like the idea of CFD code certification, and perhaps many readers do not either. But *users* do, and if we do not involve ourselves, I think we will find ourselves frozen out. Clearly, users of CFD codes (not fellow CFD researchers, but real engineers interested in building something) want something that they can use with ease and confidence. Without a certification program, I am confident that simple "market forces" will not enforce high standards, but in fact will strongly favor pseudo-robust codes. By "pseudo-robust" I mean a code that appears to give a reasonable answer to most any problem. A truly robust code, which really does give a reasonably accurate answer to most any problem, is most difficult to achieve as we all know. From the user's point of view, certification is extremely important. In my own experience, it is a major consideration in codes for ground water hydrology studies. Older codes, with perhaps a 20-year history and correspondingly archaic FORTRAN styles, are inefficient, clumsy to use, unreadable, and therefore virtually un-modifiable; nevertheless, they are certified and they are the standards of performance for new codes.

Available commercial codes certified by some agency or professional society must include attempts to guard against misuse. It is of course impossible to fulfill this goal completely. In fact, the experience of many of our professional colleagues suggest that it is well nigh hopeless. (I asked a professional colleague at a national laboratory if his code was publicly available, and he said no, but he would give it to me, because he knew or had a reasonable expectation that I would not misuse it.) CFD codes are much more difficult than linear algebra packages or statistical packages, but there are plenty of horror stories regarding the misuse of even these more straightforward codes. The reader may be of the opinion that we should not even turn loose on the world a general-purpose CFD code, but rather leave it in the hands of us "experts," at least until we can agree amongst ourselves. Unfortunately this agreement is not forthcoming, and in the meantime, commercial codes are already available (none of which is very impressive, in my opinion).

One thing that we (professionally) could require is that any code, in order to be certified, have built into it automatic error estimators, *not* as user options, but as hard-wired additional output. Like the experimental error bars on data, the error estimates can be ignored or incorporated by the practicing engineer according to his judgment, but he must be apprised of the estimates. There is a natural human desire for easy and unambiguous answers; it is our professional obligation to refuse to give them, in spite of "market forces," which I do not hold to be sacrosanct.

Appendix: Editorial Policy Statement on the Control of Numerical Accuracy[†]

A professional problem exists in the computational fluid dynamics community and also in the broader area of computational physics. Namely, there is a need for higher standards on the control of numerical accuracy.

The numerical fluid dynamics community is aware of this problem but, although individual researchers strive to control accuracy, the issue has not to our knowledge been addressed collectively and formally by any professional society or journal editorial board. The problem is certainly not unique to the *JFE* and came into even sharper focus at the 1980–81 AFOSR-HTTM-Stanford Conference on Complex Turbulent

[†] Originally published by the American Society of Mechanical Engineers as "Editorial Policy Statement on the Control of Numerical Accuracy," by P. J. Roache, K. N. Ghia, and F. M. White, in the *ASME Journal of. Fluids Engineering,* Vol. 108, No. 1, 1986; reprinted with permission.

Flows. It was a conclusion of that conference's Evaluation Committee[‡] that, in most of the submissions to the conference, it was impossible to evaluate and compare the accuracy of different turbulence models, since one could not distinguish physical modeling errors from numerical errors related to the algorithm and grid. This is especially the case for first order accurate methods and hybrid methods.

The practice of publishing comparisons based on coarse grid solutions, without systematic truncation error testing, may have been acceptable in the past. Certainly 10–15 years ago any calculation was of interest, and much of the exploratory work deserved publication, as many researchers lacked the computational power or funds to do a thorough and systematic error estimation. We are of the opinion that this practice, however understandable in the past, is outmoded and that, with powerful computers becoming more common, standards should be raised. Consequently, this journal hereby announces the following policy:

The Journal of Fluids Engineering will not accept for publication any paper reporting the numerical solution of a fluids engineering problem that fails to address the task of systematic truncation error testing and accuracy estimation.

Although the formal announcement of this journal policy is new, it has been the practice of many of our conscientious reviewers. Thus the present announcement is not a change in policy so much as a clarification and standardization.

Methods are available to accomplish this task, such as Richardson Extrapolation (when applicable), calculations with a high- and low-order method on the same grid, and straightforward repeat calculations with finer or coarser grids. As in the case of experimental uncertainty analysis, "…any appropriate analysis is far better than none as long as the procedure is explained."[§] Whatever the authors use will be considered in the review process, but we must make it clear that *a single calculation in a fixed grid will not be acceptable*, since it is impossible to infer an accuracy estimate from such a calculation. Also, the editors will not consider a reasonable agreement with experimental data to be sufficient proof of accuracy, especially if any adjustable parameters are involved, as in turbulence modeling.

We recognize that it can be costly to do a thorough study, and that many practical engineering calculations will continue to be performed on a single fixed grid. However, this practice is insufficient for publication in an archival journal.

Acknowledgments

This work was partially supported by contracts from the Air Force Weapons Laboratory (Dr. N. L. Rapagnani, contract monitor), the Naval Underwater Systems Center (S. Jordan, contract monitor), Sandia National Laboratories in Albuquerque (Dr. M. Marietta, contract monitor), and by a grant from the National Science Foundation.

References for Appendix A

1. Roache, P. J., Ghia, K. N., and White, F. M., "Editorial Policy Statement on the Control of Numerical Accuracy," *ASME Journal of Fluids Engineering*, Vol. 108, No. 1, 1986, p. 2.

[‡] Emmons, H. W. (Chairman), "Evaluation Committee Report," pp. 979-986 in *Proc. 1980-81 AFOSR-HTTM-Stanford Conference on Complex Turbulent Flows*, Vol. 11, "Taxonomies, Reporter's Summaries, Evaluation, and Conclusions," Thermosciences Division, Mechanical Engineering Department, Stanford University.

[§] Kline, S. J., "The Purposes of Uncertainty Analysis," *Journal of Fluids Engineering*, Vol. 107, June 1985, pp. 153-164.

2. Emmons, H. W. (Chairman), "Evaluation Committee Report," *Proceedings of the 1980–81 AFOSR – HTTM–Stanford Conference on Complex Turbulent Flows, Taxonomies, Reporter's Summaries, Evaluation, and Conclusions, Vol.* 11, Thermosciences Division, Mechanical Engineering Department, Stanford University, Stanford, CA, 1981, pp. 979–986.
3. Shirazi, S. A., and Truman, C. R., "Evaluation of Algebraic Turbulence Models for PNS Predictions of Supersonic Flow Past a Sphere-Cone," *AIAA Journal,* Vol. 27, No. 5, 1989, pp. 560–568.
4. Cheng, S. I., "Numerical Integration of Navier-Stokes Equations," *AIAA Journal,* Vol. 8, Dec. 1970, pp. 2115–2122.
5. Schonauer, W., private communication; see also, Schonauer, W., Raith, K., and Glotz, G., "The Principle of the Difference of Difference Quotients as a Key to the Self-Adaptive Solution of Nonlinear Partial Differential Equations," *Computer Methods in Applied Mechanics and Engineering,* Vol. 28, No. 3, 1981, pp. 327–359.
6. Mehta, U., "Computational Requirements for Hypersonic-Flight Performance Estimates," AIAA Paper 89–1670, June 1989.
7. Steinberg, S., and Roache, P. J., "Symbolic Manipulation and Computational Fluid Dynamics," *Journal of Computational Physics,* Vol. 57, No. 30, 1985, pp. 251–283.
8. Shih, T. M., Tan, C. H., and Hwang, B. C., "Effects of Grid Staggering on Numerical Schemes," *International Journal for Numerical Methods in Fluids,* Vol. 9, No. 2, 1989, pp. 193–212.
9. BIottner, F. G., "Finite Difference Methods of Solution of the Boundary Layer Equations," *AIAA Journal,* Vol. 8, No. 2, 1970, pp. 193–199; see also, "Finite-Difference Methods for Solving the Boundary Layer Equations with Second-Order Accuracy," *Proceedings of the Second International Conference on Numerical Methods in Fluid Dynamics,* Lecture Notes in Physics, Vol. 8, edited by M. Holt, Springer-Verlag, New York, 1971.
10. Blottner, F. G., and Roache, P. J., "Nonuniform Mesh Systems," *Journal of Computational Physics,* Vol. 8, No. 3, 1971, pp. 498–499.
11. Kuruvila, G., and Anderson, J. D., "A Study of the Effects of Numerical Dissipation on the Calculation of Supersonic Separated Flows," AIAA Paper 85–0301, Jan. 1985.
12. Fujii, K., Gavali, S., and Holst, T. L., "Evaluation of Navier-Stokes and Euler Solutions for Leading-Edge Separation Vortices," *International Journal for Numerical Methods in Fluids,* Vol. 8, 1988, pp. 1319–1329.
13. Thareja, R. R., Stewart, J. R., Hassan, O., Morgan, K., and Peraire, J., "A Point Implicit Unstructured Grid Solver for the Euler and Navier-Stokes Equations," *International Journal for Numerical Methods in Fluids,* Vol. 9, 1989, pp. 405–425.
14. Dietrich, D. E., Roache, P. J., and Marietta, M. G., "Convergence Studies with the Sandia Ocean Modeling System," *International Journal for Numerical Methods in Fluids* (to be published).
15. Durst, F., and Pereira, J. C. F., "Time-Dependent, Backward Facing Step Flow," *ASME Journal of Fluid Engineering,* Vol. 110, No. 3, 1988, pp. 289–296.
16. Blottner, F. G., "Accurate Navier-Stokes Results for the Hypersonic Flow over a Spherical Nosetip," AIAA Paper 89–071, June 1989.
17. Nguyen, T. V., and Maclaine-Cross, 1. L., "Incremental Pressure Drop Number in Parallel-Plate Heat Exchangers," *ASME Journal of Fluids Engineering,* Vol. I 10, No. 1, 1988, pp. 93–96.
18. de Vahl Davis, G., "Natural Convection of Air in a Square Cavity: A Bench Mark Numerical Solution," *International Journal for Numerical Methods in Fluids,* Vol. 3, No. 3, 1983, pp. 249–264.
19. Mehta, U., "Flight Performance Estimation Utilizing Computational Fluid Dynamics," *Proceedings of the Fifth National Aero-Space Plane Technology Symposium,* NASA Langley Research Center, Hampton, VA, Oct. 1988.

APPENDIX B.

§ VALIDATION - WHAT DOES IT MEAN ?

The following is a reproduction of Roache, P. J. (2009), "Perspective: Validation - What Does it Mean?", *ASME Journal of Fluids* Engineering.

Perspective: Validation - What Does it Mean?

Patrick J. Roache
Consultant

Abstract

Ambiguities, inconsistencies and recommended interpretations of the commonly cited definition of validation for CFD codes/models are examined. It is shown that the definition-deductive approach is prone to misinterpretation, and that bottom-up descriptions rather than top-down legalistic definitions are to be preferred for science-based engineering and journal policies, though legalistic definitions are necessary for contracts.

Keywords: validation, calibration

Introduction

Validation: The process of determining the degree to which a model {and its associated data} is an accurate representation of the real world from the perspective of the intended uses of the model.

Unfortunately, considerable disagreement exists on what this definition *means*, or should mean.

This definition of validation has been cited extensively in CFD (Computational Fluid Dynamics) and other computational modeling fields, and is widely accepted. Despite the apparent clarity of this concise one-sentence definition using common terms, there is disagreement on its interpretation among scientists and engineers, who are habitually careful readers. There are at least three contested issues: whether *degree* implies acceptability criteria (pass/fail); whether *real world* implies experimental data; and whether *intended use* is specific or general (even by those who think it is needed at all). This gives $2^3 = 8$ possible interpretations of the same definition, without even getting into arguments about what is meant by *model*, i.e. computational, conceptual, mathematical, strong, weak. The job of sorting out claims and arguments is further complicated by the fact that principals in the debates have sometimes switched sides on one or more of these three issues (myself included).

Before examining the definition of validation, we need to make a small distinction on what it is we are claiming to validate, i.e. between *code* and *model*. A model is incorporated into a code, and the same model (e.g. some RANS model) can exist in many codes. Strictly speaking, it is the model that is to be validated, whereas the codes need to be verified. But for a model to be validated, it must be embodied in a code before it can be run. It is thus common to speak loosely of "validating a code" when one means "validating the model in the code," and vendors like to claim they are providing a "validated code," and legal and regulatory requirements may specify use of "verified and validated codes". In theory, the same model would only have to be validated in one (verified) code to be accepted as validated in another (verified) code; in practice for RANS codes, this is unrealistic, so "validating a code" is usually meaningful in context.

History of the Definition

The definition was precisely stated in a 1996 (re-issued in 2003) U.S. DoD Instruction [1; see also 2,3], which referred to an earlier mini-symposium that used almost the same wording. The DoD re-issue in 2003 [1] added the bracketed additional phrase {*and its associated data*} after the word *model*, which would suggest a strong-sense concept of *model*. The definition was adopted (without the bracketed term) in the AIAA Guide for V&V in CFD [4] and in the ASME V&V 10 [5] which was based in many aspects on [4].[155] The definition is widely used beyond these documents and the observations herein should not be construed simply as criticisms of these sources, but rather as cautions that there are inherent problems with interpretation. The documents cited [1-5] are uneven in their stated interpretations on these issues, with V&V 10 [5] being specific and clear on all three issues. Unfortunately, while acknowledging that a range of definitions exist for validation and other V&V terms, [5] does not acknowledge that a range of interpretations exist for the same definition. Also, while citing [1-4] for its definition of validation, it does not acknowledge the fact that it differs from [1-4] in its interpretation notably on the issue of inclusion of pass/fail criteria. It is also a fact that, for each publication, opinions on what the definition means differ even among members of the same committee that wrote the document. ASME V&V 20 [6] noted the definition but also its range of interpretations, adopting a more general descriptive approach. Likewise, neither the ASCE monograph on V&V for free surface flows [7] nor my 1998 book [8] required the deductive top-down approach implied by legalistic definitions, using instead a descriptive bottom-up approach. It is noteworthy that, in spite of all the agonizing over interpretations, none of the specifics of the complete V&V methodology presented in V&V 20 [6] is affected by any of these choices. Furthermore, while consistent use throughout the computational communities is desirable, there is no necessity for this journal or others to accept a DoD or other definition as canonical, especially when it is easily shown that

[155] Although [5] cited the 2003 version of [1] it did not include the bracketed term added to the original 1996 issue of [1].

there are inherent problems with the definition and a wide range of interpretations. (Contracts present a different consideration; see below.)

Based on my contacts in the V&V community, including committee participation in the writing of [5-7], professional contacts with the principals of [4], and experience teaching twelve short courses on V&V, I believe that most professionals make the following interpretations of the definition upon first reading.

Issue #1. Acceptability Criteria (Pass/Fail)

Regarding the issue of whether acceptability criteria (or adequacy, or pass/fail criteria) are included in this definition of validation, initially people generally say "yes" without hesitation. This is due mostly to a correct recognition that pass/fail decisions must be made in any engineering project, and reinforced by the later phrase "from the perspective of the intended uses of the model" which understandably seems to imply such project-specific criteria (see discussion below). However, people quickly see the value of the alternative view. Although pass/fail criteria are certainly project requirements, the requirements do not necessarily need to be included in the term "validation." In fact, in the original DoD documents [1-3] the term "acceptability" was not used in regard to validation, but in regard to "accreditation" (and which has elsewhere been described as "certification"). From [1]:

Acceptability Criteria (Accreditation Criteria). A set of standards that a particular model, simulation, or federation [system of interacting models] must meet to be accredited for a specific purpose.

Accreditation. The official certification that a model, simulation or federation of models and simulations and its associated data are acceptable for use for a specific purpose.

However, acceptability for accreditation as stated in [1] involved additional criteria besides validation accuracy, which supposedly was intended to be included in validation [9]. But close reading of the documents themselves [1-3] give no indication of this, and strongly suggest to me that the acceptability criteria reside under accreditation (or certification, or perhaps another project-related term) rather than validation. The AIAA Guide [4] is somewhat vague (and committee members disagree), and there is widespread misunderstanding of [4] on this point (see discussion below under Issue #3). But V&V 10 [5], even though inspired by the AIAA Guide [4], strongly includes pass/fail criteria, even to the point of insisting that the pass/fail criteria (validation requirements) be set firmly before the comparison to experiment, in the description of intended use. No acknowledgement of this departure from [4] is given in [5], to the likely confusion of any user-engineer who happens to read both documents and who has other things on his mind. It would have been less confusing if the sources each had used different wording for the definition, which might alert the user-engineer, rather than use the same "definition" with different interpretations of the terms.

More important than what the documents state is the fact that people quickly see the advantage of not including a pass/fail tolerance while performing validation. Rather, one simply evaluates the agreement between computational results and experimental results (with their respective uncertainties - see below), and presents the difference as the **level** of validation. This recognizes the fact that the same validation level (e.g., 10% agreement for skin friction coefficient) may be adequate for one application and not for another. This is just the kind of validation exercises performed for many years for RANS turbulence models, for example.

There are two very distinct processes: first, comparison of model predictions with experimental values, leading to an assessment of model accuracy, and second, determination of acceptability or pass/fail of that accuracy level for a particular application. The methodologies employed in each process have virtually

nothing in common. In some usage, a model whose results have been compared to experiments is labeled *validated* regardless of the agreement achieved. In this loosest use of the term, *validated* then is not a quality of the code/model *per se*, but just refers to the QA (Quality Assurance) process. Carried to an extreme, this viewpoint gives the designation *validated* even to very poor models. Celik [10] has pointed out that it would be misleading to assign the inevitably value-laden term "validated" for a code that produces unarguably poor results (say wrong qualitative trends, e.g., lift coefficient decreasing with angle of attack) just because it has gone through the validation QA *process*. I agree, and do not recommend this usage. A more moderate usage is to call the model *validated*, regardless of the agreement achieved, but to state explicitly that the model is validated to a specified level and within the validation uncertainties determined from following the procedures in [6] or other. This way, the validation statement provides a quantitative assessment, but stops short of a rigid pass/fail statement, since that requires consideration of the design, cost, risk, etc. This usage is well presented by Oberkampf et al. [11], pg. 348. "Stating our view succinctly: validation deals with quantified comparisons between experimental data and computational data; not the adequacy of the comparisons." The other extreme makes validation project-specific by specifying the error tolerance *a priori*, e.g. see [5]. This ties a code/model validation rigidly to a particular engineering project rather than to less specific science-based engineering (or worse, it neglects the fact that agreement may be acceptable for one application and not for another).

Since not all comparisons should result in a code being given the value-laden designation of *validated*, some minimal agreement should be required. As a reviewer has noted, since it is impossible to avoid attaching a value to *validated*, it can be argued that it is preferable to attach a well defined criterion from the start. But on balance, I think this is outweighed by the disadvantages, as discussed (ephemeral pass/fail criteria, applicability of validation results to more than one project, disparate methods for assessing fidelity and adequacy). The general (and necessarily vague) level of acceptable agreement must be determined by common practice in the discipline. The simulation results with their uncertainties are compared to experiments with their uncertainties, and if reasonable agreement as determined by the state-of-the-art standards (including at least correct qualitative trends) is achieved, then the code/model can be termed *validated*. This does not necessarily mean that the model will be adequate for all applications. Such a project-specific pass/fail tolerance should be relegated to accreditation or certification [8]. The value of this pass/fail tolerance tends to vary over time with design decisions, product requirements, and economics, even though the objective results of the validation comparison itself have more permanent value.

Many discourage the use of the term "validated code" no matter how good the agreement with experiment, because it might be misleading or even deliberately misused, e.g. in commercial code marketing. But it does not seem realistic to try to outlaw the past participle, and codes that have gone through validation will inevitably be referred to as "validated codes." Nevertheless, as Tsang [12, cited in 8, page 26] noted, "almost by definition, one can never have a Validated computer model without further qualifying phrases." The qualifications include knowledge of the experimental validation set points, the specific validation variables or metrics, what is included in *model*, and of course the degree of validation achieved, which requires stated uncertainties of both computations and experiments.

Issue #2. Necessity for Experimental Data

In the validation definition, most engineers read "real world" to imply *real world data*, i.e. what most people would call experimental data. Surprisingly, not everyone agrees with this interpretation. (In [1-3] the distinction was not specifically addressed; in [4,5] the requirement was clear and unequivocal, although some members of the committees disagreed.) The apparent motivation is to try to gain the approval implicit in "validation" without the onerous requirement for obtaining real experimental data. There are difficult problems, e.g. nuclear stockpile, for which further testing is outlawed. It is not always clear what these

proponents would substitute. Some look for agreement between different models. As noted in [8, p. 276], if one code has been previously validated, it can be regarded as a repository of experimental information, a set of second-hand experimental data plus smoothing and interpolation/extrapolation to parameter values other than experimental set-points. But in general, code-to-code comparison is not validation. The recommended view, agreeing with [5-8], is uncompromising: no experimental data means no validation.

Regarding validation by comparison with a previously validated code, a reviewer has noted that, if a second code is being validated at a set point, the original data could be used, not the first code that has been "validated" at that same set point. In principle, this would usually be the preferred approach. However, for some practical situations the use of a previously validated code could be preferable and certainly more convenient and, I believe, acceptable. First, note that previous multiple validation experiments may not agree with each other, even within the experimental uncertainties (if indeed these have been presented), and they may not be at exactly the same set points. Second, suppose that a new model to be validated is not expected to be as accurate as previous models (but perhaps has an advantage of simplicity, or computational speed, or numerical stability, or lack of sensitivity to grid resolution and is therefore cheaper to run). Then it would make sense to compare the results of the new model with those of a previously validated (but perhaps more complex, slower, less robust, or more expensive) model. (A ready example is a turbulence model using new wall functions, which could be validated against previously validated models employing integration to the wall.) It would be impossible to justify if the new model were intended to be more accurate than the old model taken as a benchmark, except as an interim validation exercise used to justify further validation work (perhaps with new and improved validation experiments).

Issue #3. Intended Use

The requirement for "intended use" sounds good at first, but it fails upon closer thought. Did D. C. Wilcox [13] need to have an "intended use" in mind when he evaluated the k-ω RANS turbulence models for adverse pressure gradient flows? He may very well have had uses in mind, but does a modeler need to have the same use in mind two decades later? If not, must the validation comparison be repeated? Certainly not.

The "intended use" phrase also bears on pass/fail criteria (Issue #1), seeming to indicate that pass/fail criteria are to be included in the definition of validation. There is widespread misunderstanding of the AIAA Guide [4] on this point, as acknowledged by W. Oberkampf [11], a principal architect of [4]. He stated that pass/fail criteria are not included: "We argue that this is what the words mean in the definition ..." The fact that the authors must "argue" the interpretation indicates that the document is unclear, which is understandable given the phrase "from the perspective of the intended uses of the model." Oberkampf insists that "intended use" applies not to a pass/fail tolerance but rather to the metrics involved. Although this observation is relevant, it is not complete, because the same metrics might be applicable to different end uses, just as the same pass/fail tolerances might be. Although [1-4] are not emphatic about specificity of intended use, they are suggestive. V&V 10 [5] is admirably clear but unrealistically strong, even to the point of insisting on *a priori* specification of validation criteria, which if taken seriously would effectively eliminate the possibility of validation in any basic research sense, in my opinion. All these documents [1-5] have a strong orientation to management of large engineering projects, which deters from their applicability to basic research, unlike V&V 20 [6].

Clearly, much of the confusion is the result of trying to use the same word for different needs. Project oriented engineers are more concerned with specific applications, and naturally tend to rank *acceptability* within *validation* (which term is used more often than *accreditation* or *certification*). Research engineers and scientists tend to take a broader view, and often would prefer to use *validation* to encompass only the assessment of accuracy level, rather than to make decisions about whether that level is adequate for

unspecified future uses. It is also significant to recognize that these project-specific requirements on accuracy are often ephemeral, so it is difficult to see a rationale for *a priori* rigid specifications of validation requirements [5,11] when the criteria so often can be re-negotiated if the initial evaluation fails narrowly.

Recommended Interpretation and Alternative Description

My recommendations, consistent with V&V 20 [6], are that choices for the interpretation of the validation definition be made as follows.

Recommendation on Issue #1.

Criteria for *acceptability* of accuracy (adequacy, or pass/fail criteria, or accuracy tolerance) are not part of validation, but analysts performing validation exercises should be wary of appearing to bless a code as "validated" when it is clearly unsatisfactory for any reasonable application (e.g. it cannot even predict correct qualitative trends). In an engineering project, the acceptability of the agreement is part of the next project step, variously called accreditation, certification, or other. It is an engineering management decision, not a scientific evaluation.

Recommendation on Issue #2.

Experimental data is necessary for Validation. Many have said unequivocally [5-8,11] that experimental data are the *sine qua non* of validation.

No experimental data => No validation

Many other factors remain, of course, including the quality and quantity of the data, the necessity for uncertainty estimates for both modeling and experiments [6], the extent of the domain of validation (the range of parameter space in the set points of the experiments and the interpolation/extrapolation of experimental and computational results), whether previously validated codes can be used as a secondary database, whether scaled experiments are adequate, etc. But as a minimum, some experimental data are required. This data can include historical observations and already established scientific facts (especially obvious for invalidation), as pointed out by a reviewer, but it is noteworthy that [4,5] disagree, adopting a literal sense of temporal "prediction" which is at odds with scientific practice.

Recommendation on Issue #3.

Intended use, at least in its specific sense, is not required for validation. The common validation definition could be salvaged by re-defining *intended use* to include very general intentions, but frankly this appears to be a hollow exercise. The fact is that a useful validation exercise does not necessarily require an intended use, specific or general. For example, the well-known data on turbulent backstep flow of Driver and Seegmiller [14] in the ERCOFTAC database can be used for code/model validation, with neither the experimenters in 1985 nor modelers in (say) 2008 having a specific use in mind. This is precisely the situation for the Lisbon III Workshop on V&V [15].

However, it is also true and very important that (as recommended strongly in [4-6,11]) experiments designed specifically for a validation exercise, and with a specific application in mind, and with collaboration between experimenters and modelers in the design of the experiments, are much more likely to

produce data on the relevant metrics with relevant precisions than are experiments designed without applications in mind.

Alternative Description.

Alternately, for science-based engineering, we can *describe* validation rather than rigidly define it. First (and virtually universally agreed upon [4-8,11]) is the distinction between verifications and validation. Verifications (first of the code and then of particular calculations or solutions) are simply matters of mathematics, and address questions of correct coding and discretization accuracy of particular solutions, whereas validation involves comparison with reality, i.e. science (or physics, in its most general sense). In general terms, validation involves comparison of modeling results with experimental results. This has been used in the past, but I agree with [4-6,11] that it is too soft. The trouble (as noted in [11]) is that the difference between model result and experiment is too easily taken to be the accuracy when in fact the story is more difficult. It is time to improve standards somewhat on even the minimal requirements for the term *validation*.

The minimal required improvement is contained in one word: *uncertainty*. We can describe validation (legitimate, minimal validation) as the comparison of model results *and their associated uncertainties* with experimental results *and their associated uncertainties*. A specific methodology for this comparison including interpretation of the answers is given in [6] using accepted, well established quantitative techniques for every aspect of the entire process, and using definitions and statistical techniques that are consistent between experimental and modeling methodologies. I believe that such a descriptive approach is all that is needed for science-based engineering and for journal publication standards. In any case, the warning [12] still applies: it is meaningless to talk about "validation" without significant further qualifications.

Calibration is Not Validation

Whether one takes a definition-deduction approach or a less rigid descriptive approach, it is necessary to be clear that calibration, the adjustment or tuning of free parameters in a model to fit the model output with experimental data, is not validation. (This distinction is emphasized in each of [4-7] but earlier uses [8] often described calibration as just validation for a restricted range of physical parameters.) Calibration is a sometimes necessary component of (strong sense) model development. But this calibration is not to be considered as validation, which occurs only when the previously calibrated model predictions are evaluated against a set of data not used in the tuning [4-8]. There is no value in tuning free parameters to obtain a drag coefficient to match an experimental value, and then claiming code/model validation because the "prediction" agrees with the same experiment. Historically, this has been a common failing of free-surface flow modeling projects [7]. Of course, if all point-values and functionals of interest are well matched using a small set of free parameters with physically realistic values, this will tend to be convincing in itself, but another data set not used in the tuning will be more so.

Implications for Contractual and Regulatory Requirements

Although bottom-up descriptions of validation may be adequate for research journals, rigid and legalistic definitions will be required for contracts specifications and regulatory requirements. If a contract specifies that a "validated code" must be used in the modeling, then all parties must know what is meant by "validation" as well as verification, accreditation, etc. My preferences for the definition interpretations are given above, but whatever the contracting or regulating body decides, what is clear from the history of this controversy is the following. Although a rigid, legalistic definition may be required, it is not sufficient. As

with questions of constitutional law, interpretations will differ. No matter how carefully the words are crafted, one cannot expect all readers to make the same interpretations.

To better ensure that the intent is correctly interpreted, the contract or regulation specifications should amplify the definitions used with specific interpretations. For example, if the above definition is adopted, the specifications should not just say "real world" and expect the analyst or contractor to know that experimental data is required. The bare legalistic definitions should be expanded to describe the definition, as done notably in V&V 10 [5]. The definition - deduction approach alone is not adequate; the human capacity for equivocation assures that no legalistic definition is inviolable.

Acknowledgements

The author gratefully acknowledges discussions with H. W. Coleman and S. S. Y. Wang, and improvements suggested by four reviewers. This work has been partially supported by Idaho National Laboratories, Battelle Energy Alliance Contract no. 54525.

References for Appendix B

1. Department of Defense, 1996, *DoD Modeling and Simulation (M&S) Verification, Validation, and Accreditation (VV&A)*, DoD Instruction 5000. 61, April 29, 1996. Re-issued 13 May 2003.
2. Department of Defense, 1994, *DoD Modeling and Simulation (M&S) Management*, (DoD Directive 5000.59, January 4, 1994.
3. Military Operations Research Society, 1994, *Simulation Validation (SIMVAL) 1994*, Mini-Symposium Report, 28-30 September 1994.
4. AIAA, 1998, *Guide for the Verification and Validation of Computational Fluid Dynamics Simulations*, AIAA G-077-1998, American Institute of Aeronautics and Astronautics, Reston, VA.
5. ASME Committee PTC-60, 2006, *ANSI Standard V&V 10. ASME Guide on Verification and Validation in Computational Solid Mechanics*, 29 December 2006.
6. ASME Committee PTC-61, 2008, *ANSI Standard V&V 20. ASME Guide on Verification and Validation in Computational Fluid Dynamics and Heat Transfer*, (expected) 2009.
7. ASCE/EWRI Task Committee, *3D Free Surface Flow Model Verification/Validation*, (expected) 2009.
8. Roache, P. J., 1998, *Verification and Validation in Computational Science and Engineering*, Hermosa Publishers, New Mexico.
9. Pace, D. K., 2008, personal communication, 23 May 2008.
10. Celik, I., 2006, personal communication.
11. Oberkampf, W. L., Trucano, T. G., and Hirsch, C., 2004, "Verification, Validation, and Predictive Capability in Computational Engineering and Physics," *Applied Mechanics Reviews*, 57(5), pp. 345–384.
12. Tsang, C.-F., 1991, "The Modeling Process and Model Validation," *Ground Water*, November-December 1991, pp. 825-831.
13. Wilcox, D. C., 2006, *Turbulence Modeling for CFD*, DCW Industries, La Canada, CA.
14. Driver, D. M. and Seegmiller, H. L., 1985, "Features of a Reattaching Turbulent Shear Layer in Divergent Channel Flow," *AIAA Journal*, Vol. 23, No. 1, pp. 163-171. ERCOFTAC Classic Database C-30.
15. Eça, L. and Hoekstra, M., 2007, Announcement for the *3rd Workshop on CFD Uncertainty Analysis*, Lisbon, 23-24 October 2008, Instituto Superior Técnico, Lisbon. 1 October 2007.

APPENDIX C.

§ TUTORIAL ON CODE VERIFICATION BY THE METHOD OF MANUFACTURED SOLUTIONS

The following is a reproduction of Roache (2002), Roache, P. J. (2002), "Code Verification by the Method of Manufactured Solutions", *ASME Journal of Fluids Engineering*, Vol. 114, No. 1, March 2002, pp. 4-10.

Code Verification by the Method of Manufactured Solutions

Patrick J. Roache
Consultant

Abstract

Verification of Calculations involves error *estimation*, whereas Verification of Codes involves error *evaluation*, from known benchmark solutions. The best benchmarks are exact analytical solutions with sufficiently complex solution structure; they need not be realistic since Verification is a purely mathematical exercise. The Method of Manufactured Solutions (MMS) provides a straightforward and quite general procedure for generating such solutions. For complex codes, the method utilizes Symbolic Manipulation, but here it is illustrated with simple examples. When used with systematic grid refinement studies, which are remarkably sensitive, MMS produces strong Code Verifications with a theorem-like quality and a clearly defined completion point.

Introduction

In the semantic tangle of the subject of "Quantification of Uncertainty" (a term which itself generates some disagreement) the three most important terms, and the most universally agreed upon, are Verification of Codes, Verification of Calculations, and Validation. For reasons both logical and practical, these activities must be performed in this order [1,2]. Verification of a Calculation involves error *estimation*, whereas Verification of a Code involves error *evaluation*, from a known solution. Both Verifications are purely mathematical activities, with no concern whatever for the accuracy of physical laws. That is the concern of Validation, i.e., the agreement of the mathematics with science.

Journal Policy Statements on reporting of numerical uncertainty, of which this journal's 1986 statement [3] was the original, refer only to Verification of Calculations; the code used is assumed to be correct. "Correct" is perhaps preferable to "accurate." It can be misleading to describe a code as "accurate," because naive users of commercial software may think that, if the code they use is accurate, then their calculation will be accurate. This neglects their own burden to perform systematic discretization convergence tests for their particular calculation, i.e. Verification of a Calculation. Determining the correctness of the code itself can only be done by systematic discretization convergence tests using a known solution or "benchmark" (another term with inconsistent connotations). The best benchmark solution or standard of comparison is an exact analytical solution, i.e. a solution expressed in simple primitive functions like *sin*, *exp*, *tanh*, etc. Note that benchmark solutions involving infinite series are not desirable, typically being more numerical trouble to evaluate accurately than the CFD code itself [1]. It is not sufficient that the analytical solution be exact; it is also necessary that the solution structure be sufficiently complex that all terms in the governing equation being tested are exercised. For example, some early and misleading claims of accuracy of commercial codes which used the notoriously inaccurate first-order upstream differencing for advection terms were based on comparisons with Poiseuille, Couette or Rayleigh problems, which do not even "turn on" the advection terms.

It has often been stated in research journal articles that general accuracy Verification of Codes for difficult problems, e.g. the full Navier-Stokes equations of fluid dynamics, is not possible because exact solutions exist only for such relatively simple problems that do not fully exercise the code. Many papers and reports approach accuracy Verification of Codes in a haphazard and piecemeal way, comparing single-grid results for a few exact solutions on problems of reduced complexity. In fact, a very general procedure exists for generating analytical solutions for accuracy Verification of Codes. I first presented the method in [4], and later expanded the applications [1,2]. Although a few respected authorities (e.g., [5-9]) have recognized the power of the method, acceptance has been slow and misunderstanding is not uncommon. Based on my experience in many discussions with professional colleagues including teaching short courses with participations by senior researchers, the misunderstanding is due to the deceptive simplicity (elegance?) of the concept. This article is written in an attempt to clarify the concepts with simple examples, to dispel concerns often voiced, to add a few fine points, and to provide some recent references. It is hoped that the reader will bear with the somewhat conversational style, since the paper is part tutorial, part review.

The methodology provides for convincing, rigorous Verification of the numerical accuracy of a code via systematic grid convergence testing. This procedure is straightforward though somewhat tedious to apply, and verifies all accuracy aspects of the code: formulation of the discrete equations (interior and boundary conditions) and their order of accuracy, the accuracy of the solution procedure, and the user instructions.

The Method of Manufactured Solutions

The Method of Manufactured Solutions (MMS) provides a general procedure for generating an analytical solution for code accuracy verification.

The basic idea of the procedure is to simply manufacture an exact solution, without being concerned about its physical realism. (The "realism" or lack thereof has nothing to do with the mathematics, and Verification is a purely mathematical exercise.) In the original, most straightforward and most universally applicable version of the method, one simply includes in the code a general source term, *Q(x,y,z,t)* and uses it to generate a non-trivial but known solution structure. We follow the counsel of G. Polya [10]: *Only a fool starts at the beginning; the wise one starts at the end.*

We first pick a continuum solution. Interestingly enough, we can pick a solution virtually independent of the code or of the hosted equations. That is, we can pick a solution, then use it to verify an incompressible Navier-Stokes code, a Darcy flow in porous media code, a heat conduction code, an electrode design code, a materials code, etc.

We want a solution that is non-trivial but analytic, and that exercises all ordered derivatives in the error expansion and all terms, e.g., cross-derivative terms. For example, chose a solution involving *tanh*. This solution also defines boundary conditions, to be applied in any (all) forms, i.e., Dirichlet, Neumann, Robin, etc. Then the solution is passed through the governing PDEs to give the production term *Q(x,y,z,t)* that produces this solution. Since this description sounds circular, we will demonstrate with concrete examples. In [4] we used Symbolic Manipulation to generate *Q*, and this is still recommended for complex multidimensional CFD codes. However, for illustration purposes, we can consider simple one-dimensional transient problems and generate the results by hand, in unambiguous steps.

Three Example Problems in MMS

To emphasize the generality of the concept, we pick the first example solution *before we specify the governing equations*. Then we will use this same solution for two different problems, i.e. set of governing PDEs and boundary conditions. The chosen solution U(t, x) is the following.

$$U(t,x) = A + \sin(B),\ B = x + Ct \tag{1}$$

Example 1

First, let us apply this 1-D transient solution to the nonlinear Burgers equation, often taken as a model for CFD algorithm development [2].

$$u_t = -uu_x + \alpha\, u_{xx} \tag{2}$$

Incidentally, this specified solution U(t, x) is the exact solution for the constant velocity advection equation with boundary condition of u(t,0) = A + sin(Ct), so for the high Reynolds number problem (small α) it may look "realistic" in some sense, but it is not a solution to our governing Equation (2), and its "realism" or lack thereof is irrelevant to the task of Code Verification.

We determine the source term Q(t, x) which, when added to the Burgers equation for u(t, x), produces the solution u(t, x) = U(t, x). We write the Burgers equation as an operator (nonlinear) of u,

$$L(u) \equiv u_t + uu_x - \alpha\, u_{xx} = 0 \tag{3}$$

Then we evaluate the Q that produces U by operating on U with L.

$$\begin{aligned} Q(t,x) &= L(U(t,x)) \\ &= \partial U / \partial t + U \partial U / \partial x - \alpha \, \partial^2 U / \partial x^2 \end{aligned} \tag{4}$$

By elementary operations on the manufactured solution U(t, x) stated in Eq. (1),

$$Q(t,x) = C\cos(B) + [A + \sin(B)]\cos(B) + \alpha \sin(B) \tag{5}$$

If we now solve the modified equation (the Manufactured Solution)

$$L(u) \equiv u_t + u u_x - \alpha \, u_{xx} = Q(t,x) \tag{6}$$

or

$$u_t = -u u_x + \alpha \, u_{xx} + Q(t,x) \tag{7}$$

with compatible initial and boundary conditions, the exact solution will be U(t, x) given by Eq. (1).

The initial conditions are obviously just u(0,x) = U(0,x) everywhere. The boundary conditions are determined from the manufactured solution U(t, x) of Eq. (1). Note that we have not even specified the domain of the solution as yet. If we want to consider the usual model $0 \le x \le 1$ or something like $-10 \le x \le 100$, the same solution Eq. (1) applies, but of course the boundary values are determined at the corresponding locations in x. Note also that we have not even specified the *type* of boundary condition as yet. This aspect of the methodology has often caused confusion. Everyone knows that different boundary conditions on a PDE produce different answers; not everyone recognizes immediately that the same solution U(t, x) can be produced by more than one set of boundary condition types. The following combinations of inflow (left boundary, e.g. x = 0) or outflow (e.g., x = 1) boundary conditions will produce the same solution U(t, x) over the domain $0 \le x \le 1$.

Dirichlet - Dirichlet:

$$u(t,0) = U(t,0) = A + \sin(Ct), \, u(t,1) = A + \sin(1 + Ct) \tag{8}$$

Dirichlet - Outflow Gradient (Neumann):

$$u(t,0) = U(t,0) = A + \sin(Ct), \, \partial u / \partial x \big| (t,1) = \cos(1 + Ct) \tag{9}$$

Robin (mixed) - Outflow Gradient (Neumann) at $x = \pi$:

$$\begin{aligned} &a u + b u_x = c \text{ at } (t,0) \rightarrow \textit{given } a \textit{ and } b, \textit{ select } c = a[A + \sin(Ct)] + b\cos(Ct) \\ &\partial u / \partial x \big| (t,\pi) = \cos(\pi + Ct) \end{aligned} \tag{10}$$

For this time-dependent solution, the boundary values are time-dependent. It also will be possible to manufacture time-dependent solutions with steady boundary values, if required by the code.

Example 2

To further clarify the concept, we now apply the same solution to a different problem, choosing as the new governing PDE a Burgers-like equation that might be a candidate for a 1-D turbulence formulation based on the mixing length concept.

$$\begin{aligned} u_t &= -uu_x + \alpha u_{xx} + \lambda\, \partial / \partial x[(x\partial u / \partial x)^2] \\ &= -uu_x + \alpha u_{xx} + 2\lambda[x(u_x)^2 + x^2 u_{xx}] \end{aligned} \tag{11}$$

Writing the mixing-length model equation as a nonlinear operator of u,

$$L(u) \equiv u_t + uu_x - \alpha\, u_{xx} - 2\lambda[x(u_x)^2 + x^2 u_{xx}] = 0 \tag{12}$$

we evaluate the Q_m that produces U by operating on U with L_m.

$$\begin{aligned} &Qm(t,x) = Lm(U(t,x)) \\ &= \partial U / \partial t + U\partial U / \partial x - \alpha\partial^2 U / \partial x^2 - 2\lambda\,[x(\partial U / \partial x)^2 + x^2\partial^2 U / \partial x^2] \end{aligned} \tag{13}$$

By elementary operations on the (same) manufactured solution U(t, x) stated in Eq. (1),

$$Q(t,x) = C\cos(B) + [A + \sin(B)]\cos(B) + \alpha\sin(B) - 2\lambda[x\cos^2(B) - x^2\sin(B)] \tag{14}$$

If we now solve the modified model equation

$$Lm(u) \equiv u_t + uu_x - \alpha\, u_{xx} - 2\lambda[x(u_x)^2 + x^2 u_{xx}] = Qm(t,x) \tag{15}$$

or

$$u_t = -uu_x + \alpha\, u_{xx} + 2\lambda[x(u_x)^2 + x^2 u_{xx}] + Qm(t,x) \tag{16}$$

with compatible initial and boundary conditions, the exact solution for this turbulent problem again will be U(t, x) given by Eq. (1), as it was for the previous laminar problem.

Note: the same initial and boundary conditions and boundary values from the previous problem can apply, since these are determined from the solution, not from the governing PDE, nor from Q or Q_m.

Example 3

We have shown how the same solution can be used as the exact solution to verify two different codes with different governing equations, with different source terms being created to Manufacture the same solution. A third example will demonstrate the arbitrariness of the solution form. Rather than the somewhat "realistic" solution to the constant velocity advection equation given by Eq.(1), let us consider the "unrealistic" but equally valuable solution as follows.

$$U_e(t,x) = \sin(t)\, e^x \tag{17}$$

Following the same procedure for the Burgers Equation (2), we evaluate the terms in Eq. (4) from the solution U_e of Eq. (17) and obtain

$$Q_e(t,x) = \cos(t)e^x + [\sin(t)e^x]^2 - \alpha\sin(t)e^x \tag{18}$$

(arranged for readability rather than compactness). This Q_e, when added to Eq. (2), produces the Manufactured Solution Eq. (17) when compatible initial and boundary conditions are evaluated from Eq. (17).

Application to Verification of Codes

Once a non-trivial exact analytic solution has been generated, by this Method of Manufactured Solutions or perhaps another method, the solution is now used to Verify a Code by performing systematic discretization convergence tests (usually, grid convergence tests) and monitoring the convergence as $\Delta \to 0$, where Δ is a measure of discretization (e.g. Δx, Δt in a finite difference or finite volume code, and element size in a finite element code, etc.).

The principle definition of "order of convergence" is based on behavior of the error of the discrete solution. There are various measures of error, but in some sense we are always referring to the difference between the discrete solution $f(\Delta)$ (or a functional of the solution, such as lift coefficient) and the exact (continuum) solution,

$$E = f(\Delta) - f^{exact} \tag{19}$$

For an order p method, and for a well-behaved problem (exceptions are discussed in Chapters 6 and 8 of [1]), the error in the solution E asymptotically will be proportional to Δ^p. This terminology applies to every "consistent" methodology: finite difference methods (FDM), finite volume methods (FVM), finite element methods (FEM), block[156] spectral, pseudo-spectral, vortex-in-cell, etc., regardless of solution smoothness. Thus,

$$E = f(\Delta) - f^{exact} = C\,\Delta^p + H.O.T \tag{20}$$

where H.O.T. are higher order terms. (For smooth problems, it may be possible in principle to evaluate the coefficient C and the H.O.T. from the continuum solution, but as a practical matter, we do not do this in the accuracy Verification procedure.) We then monitor the numerical error as the grid is systematically refined. Successive grid halving is not required, just refinement. (See [1] for examples, analysis and extensive discussion.) Thorough iteration convergence is required. Theoretically (from Eq. 20), values of $C = error/\Delta^p$ should become constant as the grid is refined for a uniformly *p-th* order method ("uniformly" implying at all points for all derivatives). Details and many examples are given in [1]. The following summary points from [1] are worth noting.

The procedure detects all ordered errors. It will not detect coding mistakes that do not affect the answer obtained, e.g. mistakes in an iterative solution routine which affect only the iteration convergence rate. In the present view, these mistakes are not considered as Code Verification issues, since they affect only code efficiency, not accuracy.

The procedure does not evaluate the adequacy of non-ordered approximations, e.g. distance to an outflow boundary, distance to an outer (wind-tunnel wall-like) boundary, use of $\partial p / \partial y = 0$ at a wall as a boundary condition (this is not a rigorous physical boundary condition for Navier-Stokes equations). The errors of these approximations do not vanish as $\Delta \to 0$, hence are "non-ordered approximations." The adequacy of these approximations must be assessed by sensitivity tests which may be described as "Justification" exercises [1]; these are similar to Verification of Calculations in that they involve only mathematics, but are simply the results of calculations. If the code manual says it uses a 2nd order accurate

[156] Added.

discretization of $\partial p / \partial y = 0$ at walls and the MMS procedure shows that it does, then the Code is Verified on this point.

When this systematic grid convergence test is verified (for all point-by-point values), we have verified

(1) any equation transformations used (e.g., nonorthogonal boundary fitted coordinates),
(2) the order of the discretization,
(3) the encoding of the discretization, and
(4) the matrix solution procedure.

This technique was originally applied in [4] to long Fortran code produced by Artificial Intelligence (Symbol Manipulation) methods. The original 3-D nonorthogonal coordinate code contained about 1800 lines of dense Fortran. It would be impossible to check this by reading the source code, yet the procedure described Verified the code convincingly. Roundoff error was not a problem.

The arbitrary solution, produced inversely by the specification of the source term *Q*, has been aptly described by Oberkampf et al. [5] and Reed et al. [9] as a "Manufactured Solution." The approach was independently developed and named the "Prescribed Solution Forcing Method" by Dee [11]. Others who independently developed the same philosophy and essentially the same methodology are Ethier and Steinman [12] and Powers [13-16]. The first systematic exposition of the method with application to multidimensional nonlinear problems appears to be [4], but in retrospect, it seems that early instances of the use of what we now call the Method of Manufactured Solutions were cited in 1972 (the original version of [2], p. 363-365). Although the authors did not mention the method they used, it seems clear that they used this approach to generate an *ad hoc* exact solution for time-dependent model equations. Obviously, the simple solution form was chosen first, then passed through the PDE to generate the problem; see the "Errata and Addenda" section of the website *www.hermosa-pub.com/hermosa* for references and details. Undoubtedly, many of the non-infinite-series classical solutions in engineering were obtained this way, i.e. beginning with a solution form. What is strange is that the notion persisted, often repeated, that we did not have any non-trivial solutions to the full nonlinear Navier-Stokes equations, when all we have to do is "complicate" the problem a little with the addition of a source term, and we can generate all the solutions we want. The key concept is that, for Verification of Codes, these solutions need not be physically realistic.

The technique is applicable to systems of equations, including full Navier-Stokes in general non-orthogonal coordinates (e.g., see [17,18]), provided that the code is capable (or modifiable) to treat source terms in each PDE.

The technique of Code Verification by monitoring grid convergence is extremely powerful. Upon initial exposure to the technique, engineers are often negative about the method because they intuit that it cannot be sensitive enough to pick up subtle errors. After exposure to numerous examples, if they remain negative it is usually because the method is *excessively* sensitive, revealing minor inconsistencies such as first-order discretizations at a single boundary point in an elliptic problem that effects the size of the error very little (as correctly intuited) but still reduces the rate of convergence to first order for the entire solution. For examples, see [1].

The fact that the Manufactured Solution may bear no relation to any physical problem does not affect the rigor of the accuracy Verification of Codes. The only important point is that the solution (manufactured or otherwise) be non-trivial, i.e., that it exercise all the terms in the error expansion. The algebraic complexity may be something of a difficulty, but is not insurmountable, and in practice has been easily handled using Symbolic Manipulation packages like Macsyma, Mathematica, Maple, etc. Using the source-code (Fortran) writing capability of Macsyma, it is not even necessary for the analyst to look at the form of *Q*. Rather, the specification of the solution (e.g., ***tanh*** function) to the Symbolic Manipulation code results in some complicated analytical expression that can be directly converted by the Symbolic Manipulation code to a Fortran (or Pascal, C, etc.) source code segment, which is then readily emplaced in a source code module (subroutine, function, etc.) that then is called in the accuracy Verification of Code procedure. (This

"emplacement" can be performed by hand by the analyst, without actually reading the complicated source code expressions, or can itself be automated in the Symbolic Manipulation code.)

The procedure has been applied successfully to nonlinear systems of equations, with separate *Q*'s generated for each equation. Both unsteady and steady solutions are possible. (It may be useful to avoid exponential solution growth in time so as to avoid confusion with instabilities; e.g., see the fully 3-D incompressible Navier-Stokes analytical solutions of Ethier and Steinman [12].) Nonlinearity is an issue only because of uniqueness questions; otherwise, the source term complexity may be worse because of nonlinearity, but that is the job of the Symbolic Manipulation code. Non-uniqueness could be an issue because the code could converge to another legitimate solution other than the Manufactured Solution, producing a false-negative accuracy Verification test for a correct code. However, it would be difficult to contrive a situation in which a false positive accuracy Verification was obtained. In much experience, non-uniqueness has not been an issue. In [4], we applied the procedure to the nonlinear (quasi-linear) PDEs of the elliptic grid generation method for non-orthogonal coordinates. Here, the Manufactured Solution was an analytical 3-D coordinate transformation; see examples in [1].

While the simple example problems herein were chosen for transparency, complex nonlinear systems (like Navier-Stokes equations in non-orthogonal coordinates) benefit from use of computer Symbolic Manipulation routines to perform the differentiation and algebra which generate the source term. As noted above, in this approach it is not necessary to even examine the source term; using the Fortran or C code writing capabilities of software packages like Macsyma, Maple, etc. a subroutine can be produced to generate the pointwise values of the source terms for inclusion in the governing PDEs. For coupled nonlinear PDEs like those of 3-D elliptic grid generation equations, the pointwise evaluation requires simultaneous solution of 3 coupled (non-dimensional) nonlinear equations at each point. We have always used full Newton-Raphson iteration methods, the Jacobians of which are also produced by Symbol Manipulation and Fortran source code writing, so the process remains automated; i.e., one never performs any algebra or calculus manipulations by hand. In fact, for our work in grid generation via variational methods [19-24], we never even looked at the governing PDEs themselves. We considered only the variational principle itself. The Symbolic Manipulation toolkits developed by Prof. Steinberg were used to automatically generate the PDEs by (symbolic) differentiation of the variational equations to produce the Euler-Lagrange equations, to substitute 2nd order difference expressions for the PDEs, to gather terms, to write Fortran subroutines for their evaluation, to generate a specified Manufactured Solution (i.e. "a continuum grid" or parameterization which, when discrete values are evaluated, produces a computational grid), to write Fortran code for the source term including Newton-Raphson point solutions, and to perform the entire Code Verification procedure, without the researcher ever having to look at either the continuum or discretized PDEs or source terms.

Note that the Manufactured Solution should be generated in original ("physical space") coordinates (x,y,z,t) . Then the same solution can be used directly with various non-orthogonal grids or coordinate transformations.

The only disadvantage of the procedure is the requirement that the CFD code being Verified must include accurate treatment of a source term and that the code's boundary condition values not be hard-wired. Many codes are built with source terms included, and many algorithms allow trivial extension to include *Q*'s. However, in directionally-split algorithms such as Approximate Factorization [2] the time-accurate treatment of $Q(x,y,z,t)$ involves subtleties and complexities at boundaries, especially for non-orthogonal coordinates [2,18]. Thus, CFD code extensions may be required in order to apply this procedure involving "Manufactured Solutions" for Code Verification. Likewise, some groundwater flow codes are built with hard-wired homogeneous Neumann boundary conditions, $\partial f/\partial n = 0$. In order to use an arbitrary solution function, non-homogeneous boundary values like $\partial f/\partial n = g$ would be required. Alternately, one could restrict the choice of Manufactured Solution functions to fit the hard-wired values.

Likewise, to test periodic boundary conditions, one must chose a periodic function for the Manufactured Solution.

The paper by Pelletier and Ignat [8] (see also [1], p. 162-163) will be of interest to turbulence modelers interested in Code Verification. It provides simple analytical solutions for an incompressible free shear layer applicable to k-ε, k-ω and k-τ models.

The practical difficulties arising from large numbers of option combinations are discussed extensively in [1]. Briefly, option combinations are countable, and pessimistic computer science conclusions about complex codes being unverifiable are based on unrealistic conditions like "arbitrary complexity." Furthermore, the number of option combinations required often can be greatly reduced by "partitioning the option matrix" [1] based on common sense and knowledge of code structure (a "glass box" philosophy [7] as opposed to the more demanding "black box" philosophy). Failing this, codes can be Verified only for a subset of option combinations. In any case, these issues are an essential part of Code Verification by any method; they are not unique to the MMS, and in fact the generality of the MMS approach will reduce the difficulties arising from option complexity because less testing will be required for each option combination compared to the usual haphazard and piecemeal approach to Code Verification.

Also see [1] for the following topics: early applications of MMS concepts, discussions and examples of mixed 1st- and 2nd-order differencing, the small parameter (high Reynolds number) problem, economics of dimensionality, applications of MMS to 3-D grid generation codes, effects of strong and inappropriate coordinate stretching, debugging with Manufactured Solutions (when the Code Verification initial result is negative), examples of many manufactured or otherwise contrived analytical solutions in the literature, approximate but highly accurate solutions (often obtained by perturbation methods) that can also be utilized in Code Verification, the possibility of a useful theorem related to MMS, special considerations required for turbulence modeling and other fields with multiple scales, example of MMS Code Verification with a 3-D grid-tracked moving free surface (see [17]), code robustness, examples of the remarkable sensitivity of Code Verification via systematic grid convergence testing, and several methodologies for Verification of Calculations, including the recommended use of the Grid Convergence Index (GCI) for uniform reporting of systematic grid convergence studies.

Recent Work and Further Discussion

Blind Study

Salari and Knupp [17] have exercised the MMS in a blind study, in which one author (Knupp) modified a CFD code previously developed and Verified by the other (Salari), deliberately introducing errors. Then the code author tested the sabotaged code with the MMS. This exercise was not performed on simple model problems, but on a full time-dependent, compressible and incompressible, Navier-Stokes code with plenty of options. In all, 21 cases were studied, including one "placebo" (no mistake introduced) and several that involved something other than the solution (e.g., wrong time step, post-processing errors). Several formal mistakes (not order-of-convergence errors) went undetected, as expected.

Two cases showed possible limitations or cautions of MMS. Case E.4 involved an error in a DO loop for updating density arrays. Although MMS was successful, it would not have been if my suggestion (on page 78 of [1]) had been followed to use exact continuum solutions as the initial conditions to reduce run time. (This is a caution note not just for the MMS but for any Code Verification by systematic grid convergence testing using any benchmark solution.) Also, Case E.12 showed that an error in a convergence test of one variable (a ".le." test replaced with a ".ge." test) could go undetected on a particular problem because the convergence test was successfully implemented for another variable.

All ten of the OAM (Order-of-Accuracy Mistake) errors, i. e. all that could prevent the governing equations from being correctly solved, were successfully detected. In addition, several less serious mistakes were detected using the procedure.

The report also discusses error (and mistake) taxonomies, provides examples and Manufactured Solutions (with source terms) from compressible Navier-Stokes codes as well as heat conduction and 2-D Burgers equation codes in both Cartesian and curvilinear coordinates, and discusses approaches for developing Manufactured Solutions without using source terms.

From the Abstract: "The principle advantage of the MMS procedure over traditional methods of Code Verification is that code capabilities are tested in full generality. The procedure thus results in a high degree of confidence that all coding mistakes which prevent the equations from being solved correctly have been identified."

The understanding and experience of the authors is profound, and the report should be read in its entirety by anyone interested in pursuing the Method of Manufactured Solutions.

Two Multidimensional Features

In the first 1-D example problem above, we noted that the Manufactured Solution, since it is analytic, can be applied over any range of the dependent spatial variable x, e.g. the domain could extend over $x \in [0,1]$ or $x \in [-\pi,+\pi]$ etc. This feature extends to multidimensions, e.g. the same multidimensional analytic solution could be applied to a square driven cavity problem, a rectangular cavity, a backstep, a wing, etc. Also, multidimensional problems might require a little more thought to assure that all terms of the governing equations are exercised. For example, a Manufactured Solution of form U(t,x,y) = F1(t) + F2(x) + F3(y) will not be adequate to exercise governing equations containing cross derivative terms such as $\partial^2 u / \partial x \partial y$ since these are identically zero no matter how complex are the F's.

Mixed Order Methods

Roy [25,26] has shown how to treat mixed-differencing (e.g. first-order upstream differencing for advection and second-order differencing for other terms) in the systematic grid convergence tests. These two papers present the resolution, in an elegant manner, of a long-standing and practical difficulty in grid convergence studies and the GCI (Grid Convergence Index), namely, the treatment of mixed-order convergence. The mixed order behavior can arise either from the explicit use of 1st-order advection discretization and 2nd-order diffusion, or from the 1st-order observed convergence rate of nominally 2nd-order methods caused by shocks. The procedure simply involves another grid level to evaluate the *two* leading coefficients in the error expansion. The analysis includes non-integer grid refinement factors ***r***. The papers also demonstrate how non-monotonic convergence occurs from mixed-order methods in the non-asymptotic range. The method is applicable to both Verification of Codes and Verification of Calculations, and would enable more accurate (less conservative) error estimation by way of the GCI [1] for QUICK and similar methods using 2nd order accurate diffusion terms and 3rd order accurate advection terms.

Radiation Transport Code including Eigenvalue Problems

Pautz [27] presented his experience applying MMS to the radiation transport code ATTILA. The application was inspired by Salari and Knupp [17] and contains some early 1970's references on the basic ideas, but these are "...more limited than the more recent and general treatment by Salari and Knupp."

The Code Verification described includes angular flux, scattering cross sections, and spherical harmonics. In the approach used, by choosing the term f(**r**) = 1 (vs. a more general form) in the assumed

form of the Manufactured Solution, one can isolate spatial differencing terms (they cancel) and Verify whether the code handles angular terms correctly.

The code uses 3-D tetrahedral elements (utilizing linear discontinuous finite element discretization) in space and discrete ordinates in the angular discretization. The study treated the following features: Steady State - Monoenergetic, Steady State - Multigroup, Monoenergetic K-Eigenvalue, Gray Infrared.

The experience with the MMS approach was quite successful. With f(**r**) = 1, the author discovered coding mistakes in input routines (and a divide by zero for a particular combination of input options). Also, the procedure revealed mistakes in discretization of certain boundary data for the gray infrared problem. The approach Verified 2nd order convergence for norms and 3rd order convergence for average scalar flux. A subtle aspect required for successful application of the MMS procedure was the consistent finite element weighting on the MMS source term. Based on earlier 1-D analysis in the literature, it was expected that all the examined quantities (norms and average scalar flux) would exhibit 3rd order convergence, but the results of the MMS procedure demonstrated only 2nd order convergence for the norms in multidimensions.

The author concluded that MMS is "... a very powerful verification tool" [27]. Further, [pers. comm.] the author says "The power and conceptual simplicity of MMS make it an indispensable tool for code development" and recommends that MMS be required in any formal Code Verification system.

Nonhomogeneous Boundary Conditions

An arbitrary Manufactured Solution will not necessarily have homogeneous boundary conditions, e.g. u ≠ 0 or ∂u/∂x ≠0. To use such a solution, the code would require this capability. This might be inconvenient, e.g. many CFD codes have hard-wired no-slip conditions at a wall, e.g. u = 0. Rather than modify the code, some thought will produce Manufactured Solutions with homogeneous boundary values.

Nonlinear Boundary Conditions

So-called "radiation" outflow conditions are usually linear and are already covered by the previous discussion. Nonlinear boundary conditions, e.g. simple vortex conditions at outflow, or true (physical) heat-transfer radiation boundary conditions, are possible. It may be possible to select a Manufactured Solution that meets the nonlinear boundary condition; otherwise, a source term would have to be added (if it is not already present) in the nonlinear boundary equations to retain the generality of the MMS.

Shocks

Shock solutions are treatable by the MMS, with additional considerations. See pages 89-90 of [1], which include the work of J. Powers and associates [13-16]. The simplest approach may be to Verify the shock capturing algorithms separately on inviscid benchmark problems such as oblique shock solutions, if shock curvature is not viewed as a major question, or if it is, by using attached curved shock solutions obtained by the method of characteristics and/or detached bow shock solutions obtained by the classical inverse method. The benchmark solutions may involve asymptotic approximations in geometry or Mach number, e.g. an analysis [16] neglecting terms of $O(\varepsilon^2)$ where $\varepsilon = 1/M^2$. This approximation can be made very accurate by choosing high M, say M ~ 20, for the Code Verification exercise. Note again the distinction of mathematics vs. science; it is not a concern that the code being tested might be built on perfect gas assumptions that are not valid at such high M. This does not affect the mathematics of Code Verification; the code would not be applied at such high M when accuracy of the physics becomes important, during Code Validation.

The assumption involved in this approach is that the option matrix of the code may be partitioned (see [1], Chapter 6 for an example). That is, the Verification of the shock-capturing algorithm and coding will not be affected by the later inclusion of viscous terms, boundary conditions, etc. Other option-partitioning assumptions will occur to the reader: separated Verification of a direct banded Gaussian elimination routine in a FEM code; Verification of shock-capturing algorithm separate from non-ideal gas effects; radioactive decay option (which is dimensionless) verified separately from spatial discretization of flow equations. This partitioning approach requires the "black-box" Verification philosophy to be modified to a "glass-box" [7], i.e. some knowledge of code structure is required to justify the approach, and it will be more difficult to convince reviewers, editors, contract monitors, regulators, etc. that the approach is justified. The work savings can be enormous, of course, avoiding the factorial increase of complexity inherent in option combinations.

Another straightforward approach for shocks that does not involve partitioning the option matrix is to generate a Manufactured Solution that is in fact (in the continuum) C^{∞} smooth, but that has such strong gradients in some region that it appears as a shock over the targeted range of grid resolutions. The possible difficulty here is that some shock capturing algorithms are based on the conservation equations without source terms, e.g. Godunov's method and modern variants, and these could conceivably fail where source terms are present. Any shock-capturing algorithm based purely on geometric limiters will be oblivious to the source terms and should work without modification.

Requirement for Source Terms

In the version demonstrated, the MMS requires that the code be capable of treating source terms in each PDE. For some engineering codes, this is always the case, e.g. time-dependent chemistry codes, grid generation codes based on elliptic generation. Another approach to MMS developed by Knupp (see [17] and [1], Chapters 3 and 6) is applicable to variable coefficient problems, e.g. groundwater transport codes or heat conduction codes with variable properties. In this method, a Solution is Manufactured by solving directly for the distribution of variable coefficients that produces it. Generally, for Navier-Stokes codes, distributed source terms are non-physical and would not have been included, so these codes would have to be modified to use the method. For codes developed "in house," this is very little trouble. The only tricky situation we know of occurs with implicit Approximate Factorization codes built for second-order time accuracy, as noted above and in [1]; in this case, the consistently second-order treatment of source terms is subtle. Still, the trouble of adding source terms is small compared to the alternative of a haphazard and piecemeal approach to Code Verification. It is certainly trivial for FEM code using direct solvers. For general purpose commercial codes, it is not an undue hardship on vendors to be required to include source terms so that users can Verify the Codes themselves. (CFD Software vendors are notoriously uninterested in performing V&V and in sharing results with customers, apparently for good reasons.)

Solution Realism

The MMS as presented generates solutions to PDEs, modified to include source terms for all dependent variables, with no concern for realism of the solution. Thus, acceptance requires that the judge recognize that Code Verification is a purely mathematical exercise. Physical realism and even physical realizability are irrelevant. Actually, there is no *requirement* that the Manufactured Solution look unrealistic, and we can invent appealing solutions if necessary to satisfy managers, regulators, public stakeholders, etc. But it is worthwhile to understand that this "realism" is mere window dressing. It is also risky, in that it encourages a dangerous misconception and opens the door to criticism and arguments about adequate

"realism," which after all is a qualitative concept and again opens the door to piecemeal and perpetual Code Verification exercises.

Furthermore, "realistic" solutions can actually be less desirable. For example, a realistic solution for Navier-Stokes equations would have a boundary-layer behavior at walls. But then the terms neglected in classical boundary layer theory (e.g., $\partial p/\partial y$) will not be strongly exercised in the code, and a minor error might conceivably slip by undetected, e.g. a one-sided difference expression for $\partial p/\partial y$ near the wall thought to be 2nd order but actually 1st order will not affect the answer if $\partial p/\partial y = 0 + O(\Delta^2)$ whereas an "unrealistic" solution based on ***tanh***, etc. would exercise these terms.

Code Verification with a Clearly Defined Completion Point

As noted earlier, it is now well-recognized that benchmark solutions for Code Verification must exhibit sufficiently complex structure that all terms in the governing equation being tested are exercised. What apparently is not so widely recognized today is that, once a code (or rather, a specific set of code option combinations) has been convincingly Verified on such a solution, it is nearly pointless to continue exercising it on simpler problems. I say "nearly" because the exercises have some value, but should not be thought of as Verification, but as Confirmation exercises; see Chapter 1 of [1]. In the present view, Code Verification has a theorem-like quality, and therefore terminates. Like a high-school student plugging numbers into the solution for the quadratic equations, a code user who performs Confirmation exercises gains confidence in the code and in his ability to set up the code and to interpret the results. Such Confirmation exercises are valuable, indeed necessary, as part of user training, but these should not be confused with Code Verification. Similarly, we recognize that simple problems (e.g. 1-D linear wave propagation) are useful in algorithm development, in exploring algorithm and code characteristics, and in comparing performance of different codes, but once again, these comparison exercises should not be confused with Code Verification. For example, one could have three codes of 1st, 2nd and 4th order accuracy, each of which was rigorously Verified to be so. Then a comparison exercise based on simple wave propagation with a linear advection-diffusion equation would be expected to show increasing accuracy; however, this does not alter the previously determined and completed Code Verifications.

Proof?

Does such a Code Verification process deserve the term "proof"? This is another semantic question whose answer depends on the community context. Logicians, philosophers and pure mathematicians clearly view "proof" differently from engineers, with an often other-worldly standard. For example, Fermat's Last Theorem is easily demonstrable; anyone can readily convince themselves of its correctness, and a straightforward computer program can be written to convincingly demonstrate its correctness for systematic millions of cases. No one, not even the philosophers or logicians or pure mathematicians, doubts it. Indeed, if one were to put forward a counter-proof, it would be rejected by all. Yet only recently has a book-length "proof" been put forward (and doubts about it remain). Since some philosophers maintain that it is not possible even in principle to prove relativity, or Newton's laws of gravity (which are certainly provable within engineering accuracy) they are not going to accept the notion of proof of correctness of a complex code, i.e. Verification of Code.

The notion of proof is at the heart of very important criticisms, not just of the subject MMS, but of the concepts of Code Verification and especially Certification [1] for large public-policy projects. One might agree with some philosophers who maintain it is not possible to prove relativity or Newton's Laws, but would one be willing to cancel a public policy project (e.g. a nuclear waste project) because the modeling used Newton's Laws? Presumably not, but stakeholders are willing to cancel such projects under the guise

of unprovability of code correctness. The harm is done when these standards for proof of philosophers, mathematicians or logicians are applied to down-to-earth engineering projects. If we accept such out-of-context standards for proof, we cannot do anything, literally. For example, we have no proof of convergence for real systems, because the Lax Equivalence theorem only holds for linear systems. The word "proof" is itself a technical term, with different appropriate standards in logic, pure mathematics, applied mathematics, engineering, criminal law vs. torts vs. civil law (e.g. "beyond a reasonable doubt"), etc. The first definition in one dictionary for "proof" is "The evidence or argument that compels the mind to accept an assertion as true." In this sense, if not in a strict mathematical sense, one could claim that the MMS approach can provide proof of Code Verification.

I am unhesitating in claiming "convincing demonstration" and "robust Verification" for the present MMS approach. A mathematical proof would require the formalism of a theorem; as noted, it would seem that a theorem is possible, for some related problem(s). Further, if one allows the legitimacy of a non-mathematical proof in principle, then I would claim that this method provides it. It is highly unlikely that a code embodying the Burgers equation (2) and passing the Verification test for the solution of the above example could be wrong (without a contrived counter-example). More complex codes with option combinations require more tests, obviously. Computer Scientists like to negate the possibility of complete Code Verification by considering codes of "arbitrary complexity" or "arbitrary number of options," but in fact real codes have a countable, exercisable number of options; in any case, a code can be conditionally Verified just for those sets of options exercised. Geometry complexity is hard to address in general, but once again the claim of Code Verification can be conditionally stated to restrict geometry families to those tested. In practice, difficulties with complex geometries (e.g. singularities) are often not Code Verification issues at all, but are simply difficulties with Verification of Calculations; i.e. they are not issues of code correctness.

An Alternative View on Code Verification with a Clearly Defined Completion Point

The present view of Code Verification as a theorem-like process with a fixed termination is not universally accepted. In an alternative view [5-7,28] held by respected authorities, Code Verification (and Validation) are "ongoing activities that do not have a clearly defined completion point" [28], more akin to accumulating evidence for a legal case than to proving a theorem [6,7]. Both viewpoints recognize, obviously, that if the code is modified, it is a new code (even if the name of the code remains) and the new code must be re-Verified. Also, both viewpoints recognize that all plausible non-independent combinations of input options must be exercised so that every line of code is executed in order to claim that the entire code is Verified; otherwise, the Verification can be claimed only for the subset of options exercised. And both viewpoints recognize the value of ongoing code exercise by multiple users, both in an evidentiary sense and in user training. In this alternative view these activities could be part of formal Code Verification itself, rather than as Code "Confirmation" as in the present view [1].

The decision whether or not to include these activities under "Code Verification" rather than "Code Confirmation" is semantic but it must be recognized that it has practical and possibly serious consequences. For example, contractual and/or regulatory requirements for delivery or use of a "Verified Code" might be ambiguous in this view, since "Code Verification" by definition is never-ending. Also, any test (even a superficial one) could be claimed as "partial Verification." However, some advantages exist for this view, e.g. encourages more precision of the meaning of "Verified Code," and it more explicitly recognizes the value of ongoing code exercise by the user community. Both viewpoints recognize that ongoing code use and exercise can possibly uncover mistakes missed in the Code Verification process (just as a theorem might turn out to have a faulty proof or to have been misinterpreted) but in this alternative view Code Verification cannot be completed, except by specification (perhaps negotiated) of the meaning of

"Verified Code." Verification of individual calculations, and certainly Validations, are still viewed as ongoing processes in both views, of course. [1,2,5-7,28]

Concluding Remarks

The Method of Manufactured Solutions for Code Verification is typically met with skepticism, but in the experience of Oberkampf and Trucano [7] and my own, people who actually try it are enthusiastic. The MMS enables one to produce many exact analytical solutions for use as benchmarks in systematic discretization refinement tests, which tests are remarkably sensitive for Code Verification. The method is straightforward and, when applied to all option combinations in a code, can lead to complete and final Code Verification, with a well-defined completion point. It eliminates the typical haphazard, piecemeal and never-ending approach of partial Code Verifications with various highly simplified problems that still leave the customer unconvinced.

Acknowledgments

I gratefully acknowledge many clarifying discussions with Dr. W. L. Oberkampf.

References for Appendix C

1. Roache, P. J., 1998, *Verification and Validation in Computational Science and Engineering*, Hermosa Publishers, Albuquerque, NM.
2. Roache, P. J., 1999 *Fundamentals of Computational Fluid Dynamics, Hermosa Publishers*, Albuquerque, NM, Chapter 18.
3. Roache, P. J., Ghia, K. and White, F., 1986, "Editorial Policy Statement on the Control of Numerical Accuracy," *ASME Journal of Fluids Engineering*, Vol. 108, No. 1, March 1986, pg. 2.
4. Steinberg, S. and Roache, P. J., 1985, "Symbolic Manipulation and Computational Fluid Dynamics," *Journal of Computational Physics*, Vol. 57, No. 2, January 1985, pp. 251-284.
5. Oberkampf, W. L., Blottner, F. G., and Aeschliman, D. P. 1995, "Methodology for Computational Fluid Dynamics Code Verification / Validation," AIAA Paper 95-2226, 26th AIAA Fluid Dynamics Conference, 19-22 June 1995, San Diego, California.
6. Oberkampf, W. L. and Trucano, T. G., 2000, "Validation Methodology in Computational Fluid Dynamics," AIAA Paper 2000-2549, 19-22 June 2000, Denver, CO.
7. Oberkampf, W. L. and Trucano, T. G., "Verification and Validation in Computational Fluid Dynamics," AIAA Progress in Aerospace Sciences, 2002 (to appear).
8. Pelletier, D. and Ignat, L., 1995, "On the Accuracy of the Grid Convergence Index and the Zhu-Zienkiewicz Error Estimator," pp. 31-36 in ASME FED Volume 213, Quantification of Uncertainty in Computational Fluid Dynamics, August 1995, R. W. Johnson and E. D. Hughes, eds.
9. Reed, H. L., Haynes, T. S., and Saric, W. S. 1998, "CFD Validation Issues in Transition Modeling," AIAA Journal, Vol. 36, No. 5, May 1998, pp. 742-749. See also AIAA Paper 96-2051, 27th AIAA Fluid Dynamics Conference, 17-20 June 1996, New Orleans, Louisiana.
10. Polya, G. 1957, *How to Solve It; A New Aspect of Mathematical Method*, Princeton University Press, Princeton, New Jersey.
11. Dee, D. P. 1991, "Prescribed Solution Forcing Method for Model Verification in Hydraulic Engineering," Proc. 1991 National Conference on Hydraulic Engineering, ASCE, Nashville, July 29-August 2, 1991.

12. Ethier, C. R. and Steinman, D. A. 1994, "Exact Fully 3-D Navier-Stokes Solutions for Benchmarking," *International Journal for Numerical Methods in Fluids*, Vol. 19, 1994, pp. 369-375.
13. Powers, J. M. and Gonthier, K. A. 1992, "Reaction Zone Structure for Strong, Weak Overdriven and Weak Underdriven Oblique Detonations," *Physics of Fluids A*, Vol. 4, No. 9, September 1992, pp. 2082-2089.
14. Powers, J. M. and Stewart, D. S. 1992, "Approximate Solutions for Oblique Detonations in the Hypersonic Limit," *AIAA Journal*, Vol. 30, No. 3, March 1992, pp. 726-736.
15. Grismer, M. J. and Powers, J. M. 1992, "Comparison of Numerical Oblique Detonation Solutions with an Asymptotic Benchmark," *AIAA Journal*, Vol. 30, No. 12, December 1992, pp. 2985-2987.
16. Grismer, M. J. and Powers, J. M. 1996, "Numerical Predictions of Oblique Detonation Stability Boundaries," *Shock Waves*, Vol. 6, pp. 147-156.
17. Salari, K. and Knupp, P., 2000, "Code Verification by the Method of Manufactured Solutions," SAND2000-1444, Sandia National Laboratories, Albuquerque, NM 87185, June 2000.
18. Salari, K. and Roache, P. J. (1990), "The Influence of Sweep on Dynamic Stall Produced by a Rapidly Pitching Wing," AIAA Paper 90-0581.
19. Roache, P. J. and Steinberg, S. 1985, "A New Approach to Grid Generation Using a Variational Formulation," AIAA 85-1527-CP, *Proceedings AIAA 7th Computational Fluid Dynamics Conference*, 15-17 July 1985, Cincinnati, Ohio.
20. Steinberg, S. and Roache, P. J. 1986, "Using MACSYMA to Write Fortran Subroutines," *Journal of Symbolic Computation*, Vol. 2, 1986, pp. 213-216. See also *MACSYMA Newsletter*, Vol. II-2, 1985, pp. 10-12.
21. Steinberg, S. and Roache, P. J. 1986, "Variational Grid Generation," *Numerical Methods for Partial Differential Equations*, Vol. 2, 1986, pp. 71-96.
22. Steinberg, S. and Roache, P. J. 1986, "A Tool Kit of Symbolic Manipulation Programs for Variational Grid Generation," AIAA Paper No. 86-0241, AIAA 24th Aerospace Sciences Meeting, 6-9 January 1986, Reno, Nevada.
23. Steinberg, S. and Roache, P. J. 1986, "Grid Generation: A Variational and Symbolic-Computation Approach," *Proceedings Numerical Grid Generation in Fluid Dynamics Conference*, July 1986, Landshut, W. Germany.
24. Steinberg, S. and Roache, P. J. 1992, "Variational Curve and Surface Grid Generation," *Journal of Computational Physics*, Vol. 100, No. 1, pp. 163-178.
25. Roy, C. J., McWherter-Payne, M. A., and Oberkampf, W. L. 2000, "Verification and Validation for Laminar Hypersonic Flowfields," AIAA 2000-2550, June 2000 (Denver).
26. Roy, C. J. 2001, "Grid Convergence Error Analysis for Mixed-Order Numerical Schemes," AIAA Paper 2001-2606, June 2001 (Anaheim). See also this issue of *ASME JFE*.
27. Pautz, S. D. 2001, "Verification of Transport Codes by the Method of Manufactured Solutions: the ATTILA Experience", Proc. ANS International Meeting on Mathematical Methods for Nuclear Applications, M&C 2001, Salt Lake City, Utah, Sept. 2001.
28. AIAA, 1998, "Guide for the Verification and Validation of Computational Fluid Dynamics Simulations," American Institute of Aeronautics and Astronautics, AIAA-G-077-1998, Reston, VA.

APPENDIX D.

§ BASIC FORMULAS FOR V&V

The discussions of V&V semantics, philosophy, disputes, and approaches can obscure the fact that straightforward applications of V&V formulas are often possible. This appendix provides a compendium of these most straightforward formulas. Only a brief description of the terms is given here. The original equation numbers are retained so that the reader can easily refer to the main text for further discussions.

D.1 CODE VERIFICATION

Measure of Discretization Error

The basic measure of discretization error for the quantity f is the difference between the discrete solution $f(\Delta)$ (or a functional of the solution) and the exact (continuum) solution,

$$E = f(\Delta) - f^{exact} \tag{3.3.1}$$

For a consistent order p method, and for a well-behaved problem

$$E = f(\Delta) - f^{exact} = C \cdot \Delta^p + H.O.T. \tag{3.3.2}$$

where H.O.T. are higher order terms.

Method of Manufactured Solutions (MMS)

From Appendix A, the Method of Manufactured Solutions gives a means of constructing an exact solution. For 1-D time-dependent example problems, the chosen solution U(t, x) was the following.

$$U(t,x) = A + \sin(B),\ B = x + Ct \tag{1}$$

This solution is applied to the nonlinear Burgers equation,

$$u_t = -uu_x + \alpha\, u_{xx} \tag{2}$$

We determine the source term Q(t, x) which, when added to the Burgers equation for u(t, x), produces the solution u(t, x) = U(t, x). We write the Burgers equation as an operator (nonlinear) of u,

$$L(u) \equiv u_t + uu_x - \alpha\, u_{xx} = 0 \tag{3}$$

Then we evaluate the Q that produces U by operating on U with L.

$$\begin{aligned} Q(t,x) &= L(U(t,x)) \\ &= \partial U / \partial t + U \partial U / \partial x - \alpha\, \partial^2 U / \partial x^2 \end{aligned} \tag{4}$$

By elementary operations on the manufactured solution U(t, x) stated in Eq. (1),

$$Q(t,x) = C\cos(B) + [A + \sin(B)]\cos(B) + \alpha \sin(B) \tag{5}$$

If we now solve the modified equation (the Manufactured Solution)

$$L(u) \equiv u_t + uu_x - \alpha\, u_{xx} = Q(t,x) \tag{6}$$

or

$$u_t = -uu_x + \alpha\, u_{xx} + Q(t,x) \tag{7}$$

with compatible initial and boundary conditions, the exact solution will be U(t, x) given by Eq. (1).

D.2 SOLUTION VERIFICATION

Generalized Richardson Extrapolation

Without assuming the absence of odd powers in the Taylor series expansion, the usual Richardson Extrapolation is generalized to *p*-th order methods and *r*-value of grid ratio as follows.

$$f_{exact} \cong f_1 + \frac{f_1 - f_2}{r^p - 1} \tag{5.4.1}$$

If the next term in the series of Eq. (5.3.1) is zero, e.g. if centered differences were used, then the extrapolation is (p + 2) order accurate. But generally, and notably if upstream-weighted methods for advection have been used, the extrapolation is (p + 1)-order accurate.

Grid Convergence Index (GCI)

The (original, most conservative) Grid Convergence Index for the fine grid solution is

$$\text{GCI}\left[\text{fine grid}\right] = F_s \frac{|\varepsilon|}{r^p - 1} \quad , \quad F_s = 3 \tag{5.6.1}$$

or if observed p (minimum three grids) is dependable, the more optimistic value $F_s = 1.25$ is recommended.

If the convergence conditions are optimal and the extrapolated (or corrected) solution will be used in place of f_1, Eq. (5.6.1) may be excessively conservative. There are presently two possible heuristic extensions for the GCI. The first is

$$GCI[\textit{extrapolated solution}] \cong |E_1[\textit{fine grid}]| = |\varepsilon|/3 \tag{5.6.1.1}$$

which is based on $F_s = 3$, but $F_s = 1.25$ is perhaps justifiable. An alternative is

$$GCI[\textit{extrapolated solution}] \cong (F_s - 1)|E_1[\textit{fine grid}]| = (F_s - 1)|\varepsilon| \tag{5.6.1.2}$$

with no limitation stated for F_s but only $F_s = 1.25$ seems appropriate and consistent. These heuristic uncertainty estimates for the extrapolated solution are not as well founded as those for the fine grid solution, either in theory or experience.

If the coarse grid solution f_2 will be used instead of the fine grid solution f_1 (usually estimated for nearby problems) the GCI reported must be the following.

$$\text{GCI}\left[\text{coarse grid}\right] = r^p \text{GCI}\left[\text{fine grid}\right] \tag{5.7.3a}$$

The summary recommendations (Section 5.9.2) on the Factor of Safety for the GCI are as follows.

(a) Use $F_s = 1.25$ for convergence studies with a **minimum** of three grids to experimentally confirm that the observed order of convergence p_{obs} for the actual problem is reasonable, and
(b) use $F_s = 3$ for two-grid convergence studies (since a p_{obs} cannot be calculated and therefore there is no way to demonstrate that the grids are in or at least near the asymptotic regime).

$F_s = 1.25$ should not be used if the results from the minimum three grids produce a suspicious observed p. It is imprudent to use observed p > theoretical p in the GCI formulas.

Effective Grid Refinement Ratio

Use of GCI in an unstructured grid convergence study requires a heuristic definition of effective r.

$$\textit{effective } r = \left(\frac{N_1}{N_2}\right)^{1/D} \tag{5.10.3.3.1}$$

Extraction of Observed Order of Convergence p

If the grid refinement is performed with constant r (not necessarily $r = 2$), the order can be extracted directly from three grid solutions. With "1" being the solution on the finest grid,

$$p = \ln\left(\frac{f_3 - f_2}{f_2 - f_3}\right) / \ln(r) \tag{5.10.6.1}$$

A generalization of this procedure, not restricted to constant *r*, is possible using the more general procedure of solving the following nonlinear equation for *p*.

$$\frac{\varepsilon_{23}}{r_{23}^p - 1} = r_{12}^p \left(\frac{\varepsilon_{12}}{r_{12}^p - 1}\right) \tag{5.10.6.3}$$

Usual solution techniques can be applied, e.g., direct substitution iteration, Newton-Raphson, etc. (even graphical). $r \sim 2$ will be easier to solve than $r \sim 1$, and $r >> 2$ is probably not of much interest. For well behaved synthetic cases, direct substitution iteration with a relaxation factor $\omega \sim 0.5$ works well. With ρ = previous iterate for *p*, the iteration equation is

$$p = \omega\rho + (1-\omega)\frac{\ln(\beta)}{\ln(r_{12})} \tag{5.10.6.5a}$$

$$\beta = \frac{(r_{12}^\rho - 1)}{(r_{23}^\rho - 1)}\frac{\varepsilon_{23}}{\varepsilon_{12}} \tag{5.10.6.5b}$$

Grid Refinement to Achieve Target Accuracy

Once *p* is known with some confidence, one may predict the next level of grid refinement r^* necessary to achieve a target accuracy, expressed as a target Error Estimate E_1 or GCI_1, call it GCI^*. With GCI_{23} being the value for the previous two grids,

$$1/r^* = \sqrt[p]{\frac{GCI^*}{GCI_{23}}} \tag{5.10.6.6}$$

Extracting Observed *p* when Exact Solution is Known

For special cases in which an exact solution is known (e.g. in Code Verification, or for special metrics like dilatation = 0 for incompressible flow) the observed *p* may be extracted from only two grid solutions by

$$\begin{aligned} d &= (f_1 - f_2)/(f_{exact} - f_1) \\ p &= \ln(1+d)/\ln(r) \end{aligned} \tag{5.10.6.7}$$

Error Estimation without Explicit Evaluation of *p*

The following formula for the error estimate does not involve *r* or *p* (although this could be misleading, since constancy of *r* and *p* are necessary requirements).

$$E_1 = [f_2 - f_1]^2 / [f_3 - 2f_2 + f_1] \tag{5.10.11.4}$$

Characterization of Apparent Grid Convergence Behavior

The characterization of *apparent* grid convergence behavior is based on the discriminating ratio *R*, from which we recognize four *apparent* convergence conditions.

$$R = [f_1 - f_2]/[f_2 - f_3] \tag{5.11.1.2a}$$

(i) Monotone convergence for $0 < R < 1$ (5.11.1.2b)

(ii) Oscillatory convergence for $R < 0$ and $|R| < 1$ (5.11.1.2c)

(iii) Monotone divergence for $R > 1$ (5.11.1.2d)

(iv) Oscillatory divergence for $R < 0$ and $|R| > 1$ (5.11.1.2e)

Least Squares GCI

The formulas for Least Squares GCI are too complex to repeat here. See Section 5.11.

Incremental Cost of Grid Convergence Studies

Consider a base grid of 200 cells (or time steps) in each computational dimension, and coarsen by ~ r = 1.3 to produce the grid sequence n = 1,2,3,4 with N(n) = 200, 154, 118, 90. With the cost of computing a solution on the base grid $n = 1$ denoted as *Cost(1)*, we have the *Cost(n)* given by

$$Cost(n) = Cost(1) \Big/ r^{D(n-1)} = Cost(1) \Big/ 1.3^{4(n-1)} \tag{5.17.1}$$

Normalizing the cost of computing a solution on the base grid $n = 1$ to *Cost(1)* = 1, the *Cost* for each grid and the total cost Σ for the 4 grid sequence is

$$Cost(n) = 1, \quad 0.350, \quad 0.123, \quad 0.043$$
$$\Sigma = 1.516 \tag{5.17.1}$$

D.3 VALIDATION

If the three errors (from the discretization or numerical solution, the input parameters, and the experimental Data) δ_{num} , δ_{input} and δ_D are effectively independent, then the corresponding (standard) uncertainties u_{num} , u_{input} and u_D can be easily combined by the usual statistical assumption.

$$u_{val} = \sqrt{u_{num}^2 + u_{input}^2 + u_D^2} \tag{11.5.1}$$

Eq. (11.5.1) for combining standard uncertainties extends to "expanded" or probabilistic uncertainties with the same requirement that the errors δ_{num} , δ_{input} and δ_D are effectively independent.

$$U_{val} = \sqrt{U_{num}^2 + U_{input}^2 + U_D^2} \tag{11.9.1a}$$

$$U_{val,95\%} = \sqrt{U_{num,95\%}^2 + U_{input,95\%}^2 + U_{D,95\%}^2} \tag{11.9.1b}$$

Interpretation of Validation Results using Probabilistic Uncertainties;

We use the specific and common engineering target U_{val} with 95% confidence. Note that once a validation effort reaches the point where the simulation value S and the experimental value D of a validation variable have been determined, the sign and magnitude of $E = S - D$ are known. Now, $(E \pm U_{val})$ defines an interval within which δ_{model} falls, with ~95% "level of confidence", or certainty,

$$\delta_{model} \in [E - U_{VAL}, E + U_{VAL}], \quad confidence\,level \sim 95\%. \tag{11.10.1}$$

Case 1. If

$$|E| >> U_{val} \tag{11.10.2}$$

then probably $\delta_{model} \approx E$.

Case 2. If

$$|E| \leq U_{val} \tag{11.10.3}$$

then probably δ_{model} is of the same order as, or less than, $(\delta_{num} + \delta_{input} - \delta_d)$.

Extending the Domain of Validation and Reporting New Modeling Results

The objective is to estimate the Validation Uncertainty at a new application point (A) that does not correspond to an experimental set point. Known values at A are denoted by subscript A, and interpolated values there by subscript Ai. The new computation result S_A cannot reduce uncertainties at the validation points 1,2,3,4. It can only add new uncertainties at the application point A. The relation is

$$U_{\delta model,A} = \{U_{\delta model,Ai}^2 + U_{fit}^2 + U_{num,A}^2 + U_{input,A}^2\}^{1/2} \tag{11.12.3.1}$$

where $U_{fit} = 2\,u_{fit}$., the standard deviation of the least squares fit of the interpolation (may be small or zero).

The modeler reports new results at the application point A as follows.

(1) If the modeler decides to use the corrected solution S_{CA} at A, with

$$S_{CA} = \{S_A + \delta_{model,Ai}\} \tag{11.12.3.2}$$

then the reported results would be

$$S_{CA} \pm U_{\delta model,A} \tag{11.12.3.3}$$

(2) If the modeler decides to use the new calculated solution S_A at A, then the reported solution and uncertainty would be

$$S_A \pm \text{some combination of } \{U_{\delta model,A} \text{ and } |\delta_{model,Ai}|\} \tag{11.12.3.4}$$

At the time of this writing, a proper combination has not been determined. A possibly justifiable combination of unlike terms U and δ would use RMS,

$$S_A \pm \{U_{\delta model,A}^2 + \delta_{model,Ai}^2\}^{1/2} \tag{11.12.3.5a}$$

A more conservative combination would be

$$S_A \pm \{U_{\delta model,A} + |\delta_{model,Ai}|\} \tag{11.12.3.5b}$$

APPENDIX E.

Δ BIOGRAPHICAL SKETCH OF LEWIS FRY RICHARDSON

Lewis Fry Richardson (1881–1953) is a central figure in this book because of his papers on the "deferred approach to the limit", now more commonly known as "Richardson Extrapolation" (Richardson, 1910, 1927. See especially Chapter 5.) However, the present minor biographical sketch and tribute has little to do with error estimation, but much to do with personal admiration.

Considered for his scientific contributions and for all his humanity, Richardson is a singular figure of the 20th century. When I wrote the first edition of my CFD book in 1972, I opined that Richardson's 50-page paper presented to the Royal Society in 1910 "must be considered the cornerstone of modern numerical analysis of partial differential equations... He treated the iterative solution of Laplace's equation, the biharmonic equation, and others. He distinguished between steady-state problems 'according as the integral can or cannot be stepped out from a part of the boundary', i.e., between hyperbolic and elliptic problems, in modern terminology. He carefully treated numerical boundary conditions, including those at a sharp corner and those as infinity. He obtained error estimates and gave an accurate method of extrapolating answers toward the zero grid space limit, and further suggested checking the accuracy of numerical methods with exact solutions of simple geometries such as a cylinder. Finally, he was the first to actually apply these methods to a large-scale practical problem, that of determining stresses in a masonry dam." (Roache, 1972, p. 2.) Richardson had a hard time getting it published; one reviewer wanted Part A omitted and Part B condensed, while the second reviewer wanted Part B omitted and Part A condensed. (Ashford, 1985, p. 23.)

I also noted in a footnote (Roache, 1972, p. 2) that "Richardson presented what, in modern vocabulary, must be called a 'cost-effectiveness' study of the method, using human computers."

"So far I have paid piece rates for the operation [Laplacian] of about *n*/18 pence per coordinate point, *n* being the number of digits. The chief trouble to the computers has been the intermixture of plus and minus signs. As to the rate of working, one of the quickest boys averaged 2,000 operations [Laplacian] per week, for numbers of three digits, those done wrong being discounted." (Richardson, 1910, p. 325.)

O. Zienkiewicz, a dominant figure in Finite Element theory and applications, told me that he began his career as one of Richardson's later "computing boys." As I noted in (Roache, 1972) and even more applicable today and to the subject of the present book, "We may all be thankful that social conditions have changed since 1910. Many a computational fluid dynamicist would end up in the poorhouse if he were paid a certain fee per calculation, with 'those done wrong being discounted.' "

At that time, I knew nothing of Richardson's personal life, and I was *un*enthusiastic (see p. 177 of Roache, 1972) about Richardson Extrapolation! In Mandelbrot's book on Fractals (Mandelbrot, 1977) he includes a biographical sketch of Richardson because of his major conceptual contributions to this most *avant-garde* area of mathematics, fractals or fractal sets (terms coined subsequently by Mandelbrot). Fluid dynamicists recognize Richardson "for some of the most profound and most durable ideas regarding the nature of turbulence, a field into which he introduced a notion of self-similarity" (Mandelbrot, 1977, p. 30). His name is perpetuated in the Richardson Number, the fraction of turbulent energy from temperature gradients vs. velocity gradients. But I recently found out that the he also made major contributions to the field of experimental psychology, and the intellectual area in which Richardson is *most* widely known - his books and papers (Richardson, 1960a,b, 1961) being a canonical reference - is the sociometric area of quantitative peace studies! The processes in the theory of arms races are sometimes called "Richardson Processes." (See Boulding, 1978, p. 157.)

Richardson earned his B. A. in 1903 at Kings College, Cambridge with studies in physics, mathematics, chemistry, biology, and zoology. In his patent applications in 1912, he identified himself as "physicist." Later, after a career change, at the age of 47 he earned a degree in psychology.

His first scientific position was at the British Meteorological Office. He wrote the first book on numerical weather prediction (Richardson 1922, republished in 1965). His turbulence studies, which earned him election as a Fellow of the Royal Society of London, first elucidated the concept of energy cascade from low to high frequencies. The second section of his numerical weather prediction book was titled "Does the Wind Possess a Velocity?", an ostensibly foolish question that leads to the concept (though not the terminology) of fractals. As Mandelbrot (1977, p. 270) stated: "More important, at least to me, is that most of Richardson's arguments can very easily be translated into the language of the "fractal" vision of turbulence." His fractal concept is even more accessible in his (posthumously published) work on map metrics. Mandelbrot (1977) titles his own expository second chapter, "How Long is the Coast of Britain?", citing Richardson's work and his key (fractal) recognitions: that determination of coastline length from maps depends on the size of the calipers, i.e. on the measuring unit η; that, over a physically practical range of parameters, the imputed length does *not* converge as $\eta \rightarrow 0$; and that the characteristics of the coastline are consistent with the concept of fractal dimension. Mandelbrot's Plate 32 reproduces Richardson's empirical data, found among his papers after he died, on the rate of increase of the measured length of coastlines (Australia, South Africa, Germany, West Coast of Britain, and Portugal) as the caliper length η decreases. (By contrast, the measured "coastline" of a circle converges nicely as $\eta \rightarrow 0$.)

Richardson was a Quaker and a conscientious objector in the Second World War, but put his life on the line in an ambulance unit for three years. After the war, he resigned from the Meteorological Office because it was being merged into the Air Ministry. He then took up the study of the psychology of armed conflicts between nations, self-publishing a small book entitled *Mathematical Psychology of War*, and supporting himself by college teaching until he received an inheritance that allowed him to pursue his peace studies full time. From his study of the statistics and dynamics of war, he became the originator of the classical predator - prey equations.

His clearly expository papers have breadth and originality that are simply startling. Besides the numerical and meteorological work, his papers cover the following sampled diverse subjects: the measurement of mental "nature" [in the "Nature vs. Nurture" debate] and the study of adopted children; international voting power; thresholds when sensation is regarded as quantitative; the quantitative mental estimation of hue, brightness or saturation; the analogy between mental images and sparks; a quantitative view of pain; mathematical psychology of war; mental periodicities; the distribution of wars in time; chaos, international and inter-molecular; war-moods; variation of the frequency of fatal quarrels with magnitude; statistics of deadly quarrels; the problem of contiguity in population maps; the outcome of arms races (could an arms race end without fighting?); the nature of historical "fact" (is it possible to prove any general statements about historical fact?). He also had two patent applications for acoustic devices warning ships of possible collisions with large objects, above and below the water. He was inventive as an experimentalist, including measurements of turbulent dispersion, meteorology, and albedo. He tinkered with building his own analog computer in his retirement (Hunt, 1998, p. xxxiii).

Mandelbrot (1977) stated that "Lewis Fry Richardson was a great scientist whose originality ... mixed with eccentricity, and who did not in his lifetime achieve the fame he deserves... His work is characterized by the conception and execution of experiments of classic simplicity, and by respect for the facts thus revealed. He never hesitated to use precise and refined concepts when he deemed them necessary." In regard to the claim of "experiments of classic simplicity", a famous Richardson anecdote is his (last) field experiment in turbulent dispersion; with H. Stommel, he threw parsnips from a pier into a lake, observed their dispersion with an ingenious non-optical measuring instrument of his own design, and confirmed his famous 4/3 power law for turbulence.

He also had a delightful sense of humor. His *Mathematical Gazette* article (Richardson, 1925 in Ashford et al, 1993) on "How to Solve Differential Equations Approximately by Arithmetic" was a tutorial written for an audience of mathematicians in a day when numerical methods were not the fashion, and their lack of mathematical elegance was frowned upon. Richardson begins thus. "I. *The first obstacle may be one of sentiment*. It is said that in a certain grassy part of the world a man will walk a mile to catch a horse, whereon to ride a quarter of a mile to pay an afternoon call. Similarly, it is not quite respectable to arrive at a mathematical destination, under the gaze of a learned society, at the mere footpace of arithmetic. Even at the expense of considerable time and effort, one should be mounted on the swift steed of symbolic analysis. The following notes are written for those who desire to arrive by the easiest route, and who are not self-conscious about the respectability of their means of locomotion."

Richardson's (1922) book on numerical weather prediction, "arguably the first modern treatise in... dynamic meteorology" (Somerville, 1996, p. 61) is a classic, in part due to what he himself described as his "fantasy." "After so much hard reasoning, may one play with a fantasy? Imagine a large hall like a theatre, ..." filled with 64000 human computers! This vision of massively parallel computation includes consideration of unbalanced processor loads (some people calculate faster than others), inter-processor communication band widths (posting of results on "numerous little 'night signs' so neighbouring computers can read them"), a hierarchical control structure ("each region is coordinated by an official of higher rank" and a conductor in a central pulpit who regulates the slide-rule wielding, calculator-punching processors by shining "a beam of rosy light upon any region that is running ahead of the rest, and a beam of blue light on those who are behind"), and information processing ("four senior clerks" who collect the weather prediction and are "dispatching it by pneumatic carrier to a quiet room" whence it is "coded and telephoned to the radio transmitting station"). His "fantasy" includes humane considerations: "Outside are playing fields, houses, mountains and lakes, for it was thought that those who compute the weather should breathe of it freely."

His vision is charming, yet he was known for being caustic at times, and was never shy of controversy. He started a big one in 1922 with his experimental psychology paper entitled "Imagery, conation and

cerebral conductance." His interest at this time was in the quantitative measurement of thought itself and of subjective sensation, including color (hue, brightness, saturation), loudness, touch, and pain. His ideas sharply divided the scientific community, many of whom thought the entire concept was nonsense. In 1932, the British Association for the Advancement of Science appointed a committee (including Richardson) to resolve the problem. Six years later, they had come to no conclusion! After eight years, they had made "some progress" but were still unable to agree on "certain important points." (Ashford, 1985, pp. 130–133, 147–149.) His concepts are now well accepted. To the last, he was disputatious; in a letter written three days before he died to a friend and fellow war researcher Quincy Wright, Richardson chided him harshly for clinging to the 19th century concept of "balance of power" and ignoring the concept of stability. (Ashford, 1985, p. 235.)

Except for his books, his entire published works (including journal letters and posthumously published work) are available in a two-volume set (Ashford et al, 1993). His work has the quality of perpetually renewing in interest. Mandelbrot (1977, p. 270) noted that his turbulence papers "remain interesting; each fresh glance at them seems to show some angle that has passed unnoticed." I re-read his classic 1910 paper 2 decades after I had already opined (in 1972) that it was the foundation of my own field of CFD, and was astounded at how much I had overlooked or failed to appreciate in several previous unhurried readings. A short and readable biography was written by Hunt (1998) in the 1998 *Annual Review of Fluid Mechanics*. The full biography entitled "Prophet or Professor? The Life and Work of Lewis Fry Richardson" was written by Ashford (1985); eleven years later, the mathematician and humanist Philip J. Davis still considered the book worth reviewing (and his life reflecting upon) in a SIAM NEWS review titled "Weather, War, and Mathematics" (Davis, 1996). Davis also reviewed a newer popular science book on climate change by Somerville (1996), who also considers Richardson "profoundly original" (p. 57) and "a special case in the spectrum of humanity" who "makes ordinary obsessive scientists...look normal by contrast" (p. 151).

In 2006, P. Lynch (2006) published an unusual technical-historical book entitled *The Emergence of Numerical Weather Prediction: Richardson's Dream*. Lynch, like the others cited earlier, identifies Richardson as the originator of the concept of numerical weather prediction and tells the story of his trial forecast with a complete reconstruction of these hand calculations and analysis of the causes of its failure. Lynch goes on to describe how advances in better understanding of the dynamics of the atmosphere, development of stable computational algorithms, regular observations of the free atmosphere, and of course computers, enabled the first computer weather forecast to be made by 1950. He then describes developments over the next 50 years up to current practice, the fulfillment of Richardson's dream of weather prediction and climate modeling.

Richardson's other activities and interests included participation in the simplified spelling movement, rational language movements (Ido and Esperanto), Eugenics Societies (when eugenics was still politically correct and in fact espoused by most leading liberals), hiking and scouting (he was a serious naturalist), writing one unpublished play (he also used Socratic dialogue in his treatise on arms races), and of course religion and philosophy. One might think that his Quaker religion defined him, yet he did not fail to reflect on his religious inheritance, and he rejected much of it; his religious philosophy was far from naive (e.g., see Ashford, 1985, p. 69).

Davis (1996) gave this admiring observation.

Richardson was a man who walked independently, who openly admitted his failures ... even publishing them. He was an unorthodox scientist, never in the mainstream. He never played it safe.

References for Appendix E

Ashford, O. M. (1985), *Prophet or Professor? The Life and Work of Lewis Fry Richardson*, A. Hilger, Boston.

Ashford, O. M. et al (1993), Eds., *The Collected Papers of Lewis Fry Richardson*, (2 Volumes), Cambridge University Press, Cambridge.

Boulding, K. E. (1978), *Ecodynamics: A New Theory of Societal Evolution*, Sage Publications Ltd. London.

Davis, P. J. (1996), "Weather, War, and Mathematics", Book Review, *SIAM News*, November 1996, p. 4.

Hunt, J. C. R. (1998), "Lewis Fry Richardson and His Contributions to Mathematics, Meteorology, and Models of Conflict", *Annual Reviews of Fluid Mechanics*, Volume 30, 1998, pp. xii–xxxvi.

Lynch, P. (2006), *The Emergence of Numerical Weather Prediction: Richardson's Dream*, Cambridge University Press, England.

Mandelbrot, B. B. (1977), *Fractals: Form, Chance, and Dimension*, W. H. Freeman and Company, San Francisco.

Richardson, L. F. (1910), "The Approximate Arithmetical Solution by Finite Differences of Physical Problems Involving Differential Equations, with an Application to the Stresses in a Masonry Dam", *Transactions of the Royal Society of London, Series A*, Vol. 210, pp. 307–357.

Richardson, L. F. (1922), *Weather Prediction by Numerical Process*, Cambridge University Press. (Re-published in 1965.)

Richardson, L. F. (1925), "How to Solve Differential Equations Approximately by Arithmetic", *The Mathematical Gazette*, London, Vol. XII, No. 177, July 1925, pp. 415–421.

Richardson, L. F. (1927), "The Deferred Approach to the Limit", *Transactions of the Royal Society of London, Series A.*, Vol. 226, pp. 299–361. "Part I. Single Lattice." "Part II. Interpenetrating Lattices" by J. A. Gaunt.

Richardson, L. F. (1960a), *Arms and Insecurity: A Mathematical Study of the Causes and Origins of War*, Ed. N. Rashevsky and E. Trucco, Boxwood Press, Pittsburgh (now, Pacific Grove).

Richardson, L. F. (1960b), *Statistics of Deadly Quarrels*, Q. Wright and C. C. Kienau, eds., Boxwood Press, Pittsburgh (now, Pacific Grove).

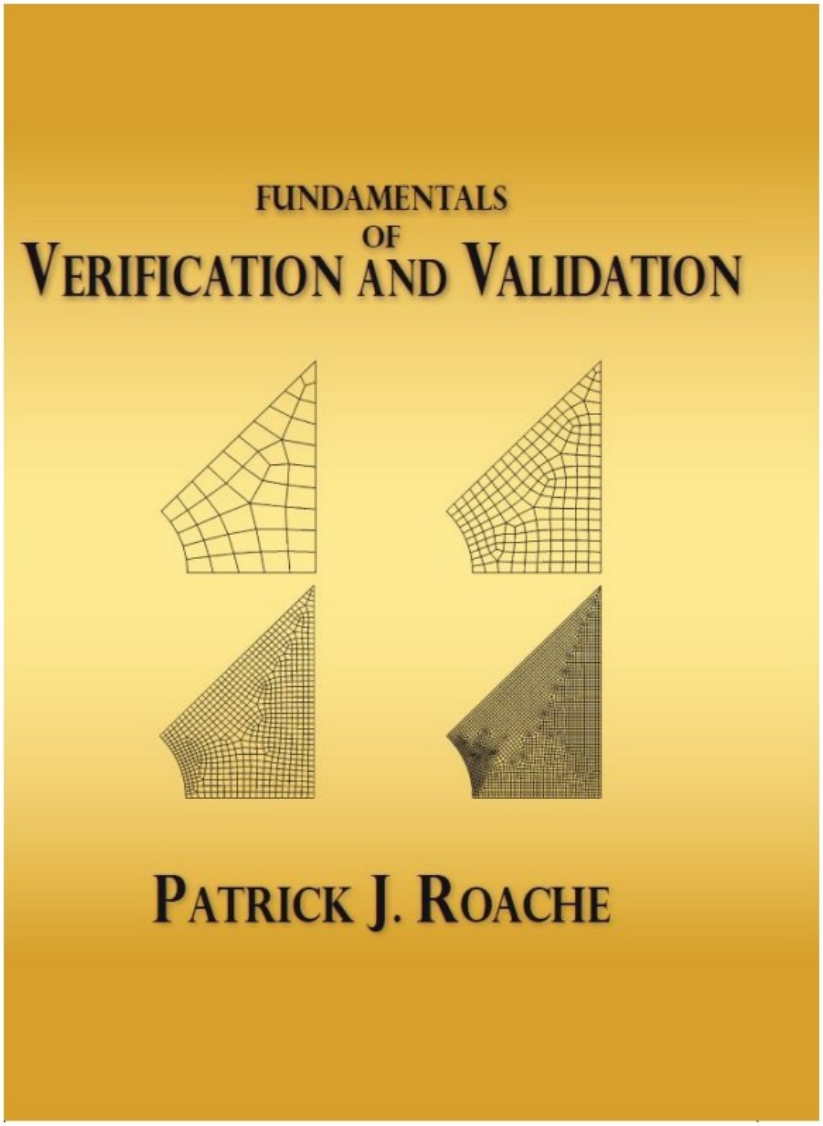
Richardson, L. F. (1961), "The Problem of Contiguity: An Appendix of Statistics of Deadly Quarrels", *General Systems Yearbook*, Vol. 6, 1961, pp. 139–187.

Roache, Patrick J. (1972), *Computational Fluid Dynamics*, First Edition, Hermosa Publishers, P. O. Box 9110, Albuquerque, New Mexico 87119-9110, USA. http://www.hermosa-pub.com/hermosa. Revised Printing 1976. See also Roache (1998).

Somerville, R. C. J. (1996), *The Forgiving Air: Understanding Environmental Change*, University Of California Press, Berkeley, CA.

Subject Index

FVV-Errata & Addenda
19 June 2020
This file contains Errata and Addenda for file
FVV-Roache-06Aug09.pdf
which contains
Roache, P. J. (2009), *Fundamentals of Verification and Validation*, Hermosa Publishers, New Mexico, 2009



ERRATA

Page 125. Table 10.5.1.1 is missing. It did appear in the 1998 edition,
Roache, P. J. (1998), *Verification and Validation in Computational Science and Engineering*, Hermosa Publishers, New Mexico, 1998.

N coarse	N fine	r	% error in $E_1 \times 10^3$
25	**100**	**4**	**–9.5993**
33	**100**	**3.0303**	**–5.6013**
50	**100**	**2**	**–2.4713**
90	**100**	**1.1111**	**–0.7680**
99	**100**	**1.0101**	**–0.6351**

Table 5.10.1.1. The improvement in the accuracy estimate from small r for grid coarsening. Steady-state Burgers equation $-U U_x + U_{xx} / Re = 0$ with $U(0) = 1$ and $U(1) = 0$ (suggestive of stagnation flow), $Re = 10$, $p = 2$. The error shown is the % error in the Richardson Extrapolated value (and therefore the error estimator E_1) for the one-dimensional "shear" dU / dx at $x = 1$. Note that the error estimate from $r = 1.1111$ (a factor 0.9 coarsening) is 3.2 times as accurate (2.4713/0.7680 ~ 3.2) as the error estimate from $r = 2$ (a factor 0.5 coarsening).

Pages 135 and 452. Equation 5.10.6.1, the denominator term $(f_2 - f_3)$ should be $(f_2 - f_1)$. This equation was correct in the precursor 1998 book.

Page 361. Case 2. E g. 11.10.3 gives the condition for Case 2 of a validation exercise. "If ... then probably δ_{model} is of the same order as, or less than, $(\delta_{num} + \delta_{input} - \delta_d)$." This follows ASME V&V20-2009, page 40, Eq. 6-2-2. See also page 355, Eq. 11.8.1.4. The statement is not so much erroneous as of very limited utility. Other efforts to expand the interpretation Case 2 have been given by two references below. Basically, all the necessary information is contained in Eq. 11.10.1, and any engineer or scientist should be able to interpret this simple interval of uncertainty about the expected value for δ_{model} . See also the discussion in the Section 11.11. For suggestions on specific though somewhat arbitrary demarcation criteria, see the following two publications.

Roache, P. J. (2016), "Verification and Validation in Fluids Engineering: Some Current Issues,"*ASME J. Fluids Eng.* 2016; FE-16-1206. DOI 1001205.

Roache, P. J. (2017), "Interpretation of Validation Results Following ASME V&V20-2009," *ASME J. Verification, Validation and Uncertainty Quantification*, June 2017, Vol 2, 024501.

Pages 365 and 454. Equation 11.12.3.2, the + sign should be a – sign. Also, be warned that the notation used on pp 365-366 is clumsy.

Minor typos undetected by a spell-checker are not mentioned herein. E. g. "gird" (5x) for "grid" (2201x), "we am ..." on page 336, etc.

ADDENDA including some publications after 2009

Page 107. General references cited in Section 5.1.1 should have included

Roache, P. J. (2004), "Calculation Verification: an Overview", *Proc. Workshop on CFD Uncertainty Analysis*, 21-22 October 2004. Instituto Superior Técnico, Lisbon, Portugal.

Page 155, Section5.12. The alternative approach to solution verification of Sinclair et al. (2006) has been refined and successfully applied at stress concentrations in 2D and 3D in the following references.

Sinclair, G. B., Beisheim, J. R., and Roache, P. J. (2016), "Effective Convergence Checks for Verifying Finite Element Stresses at Two-Dimensional Stress Concentrations," *ASME Journal of Verification, Validation and Uncertainty Quantification*, Dec. 2016, 041003.

Beisheim, J. R., Sinclair, G. B., and Roache, P. J. (2018), "Effective Convergence Checks for Verifying Finite Element Stresses at Three-Dimensional Stress Concentrations,"*ASME Journal of Verification, Validation and Uncertainty Quantification*, Sept. 2018, 034501.

Page 287, Chapter 9. References cited in Ch. 9 should have included

Roache, P. J. (2004a), "Building PDE Codes to be Verifiable and Validatable," *Computing in Science and Engineering*, Special Issue on Verification and Validation, September/October 2004, pp 30-38.

The following are excerpts from this article. Some of these points were made elsewhere in the book, but here are more specific.

The New Paradigm of Experiments Designed Specifically for Code Validation

In my opinion, the most revolutionary concept in computational physics during my career, other than simulation itself, has been the new paradigm of experiments designed specifically for validation. The new paradigm recognizes that requirements for validation are distinct and that validation experiments are much easier in some respects but more demanding in others.

In aerodynamics, for example, the emphasis in pre-computational days was on wind-tunnel experiments, which attempted to replicate free-flight conditions. Great effort was expended on achieving near-uniform inflow and model fidelity, and on minimizing wall and blockage effects. The latter required small models, which sacrificed parameter fidelity (Reynolds number) and aggravated geometric fidelity.

The new paradigm approaches the problem differently, sacrificing some fidelity

between the wind-tunnel flow and free flight, but requiring that more nearly complete details of the experimental conditions and field data be obtained. No longer is it so important to achieve uniform inflow, but it is critical to report in detail what those spatially varying inflow conditions are, so that they may be input to the computational simulation. The idea is that if the validation is good (by whatever criteria are appropriate) for a flow perturbed from the free-flight conditions, it will probably be good for the free-flight condition. Thus, blockage effects are not such major issues (and the tunnel wall itself may be modeled), and models can be larger (or tunnels smaller and therefore cheaper), thereby improving fidelity of Reynolds number and model geometry. Analogous situations occur in other experimental fields.

Unrealistic Expectations Placed on Experimentalists

It is unrealistic, even arrogant, for a code builder or user to require an experimentalist to match idealized boundary conditions. Simple constant-value boundary conditions that are a mere convenience for the code builder can require major effort, cost, and time for an experimentalist; they often compromise other more desirable qualities of the experiment, and in fact may be literally impossible to achieve. A major contribution by the code builder to the synergistic cooperation between computationalists and experimentalists (which is also part of the new paradigm) is achieved by the relatively simple work of building the code with general boundary conditions. This also happens to be what is most needed for independent code verification (or confirmation) using the MMS (Method of Manufactured Solutions).

Is Western Culture at Risk?

In an age of spreading pseudoscience and anti-rationalism, it behooves those of us who believe in the good of science and engineering to be above reproach whenever possible. Public confidence is further eroded with every error we make. Although many of society's problems can be solved with a simple change of values, major issues such as radioactive waste disposal and environmental modeling require technological solutions that necessarily involve computational physics. As Robert Laughlin **[2002]** noted in this magazine, "there is a serious danger of this power [of simulations] being misused, either by accident or through deliberate deception." Our intellectual and moral traditions will be served well by conscientious attention to verification of codes, verification of calculations, and validation, including the attention given to building new codes or modifying existing codes with specific features that enable these activities.

R.B. Laughlin (2002), "The Physical Basis of Computability," *Computing in Science & Eng.*, vol. 4, no. 3, 2002, pp. 22–25.

Page 365, Section 5.15. The ratio of the actual error to the estimated error is referred to as an **"effectivity index"** e by Prof. D. Pelletier and others. It has been used as a Figure of Merit for single or small sample studies. Note that as $\Delta \rightarrow 0$, then $e \rightarrow 1$ for any ordered error estimator, but the GCI or similar uncertainty estimators will not approach 1; rather, as $\Delta \rightarrow 0$, GCI $\rightarrow Fs$.

Ch. 3 and Appendix C: Method of Manufactured Solutions (MMS)

Roache, P. J. (2019a), "The Method of Manufactured Solutions for Code Verifications", Ch. 12 in Beisbart and Saam (2019). See below.

Most of this article repeats the material of Ch. 3 and Appendix C, but it also includes references to novel applications including ice flow models (next).

Bueler, E., Brown, J., and Lingle, C. (2007), "Exact solutions to the thermomechanically coupled shallow-ice approximation: effective tools for verification", *Journal of Glaciology*, Vol. 53, No. 182, 2007 pp. 499-516.

The authors developed a realistic MMS solution to verify a code for solving glacial ice flows. Solution realism was important to gain acceptance at a time when the glaciology science community was skeptical of models and verifications. The 3D time-dependent model involves many difficult features: a free boundary, thermomechanical coupling between a highly nonlinear power law viscosity and the temperature distribution, and coupling between energy conservation and thin-layer mass conservation PDEs with integrals in the nonlinear PDE coefficients. The solution realism aided acceptance and interpretation of controversial temperature "spokes" in ice flows found by several investigators.

K. I. Aycock, N. Rebelo, B.A. Craven (2020), "Method of manufactured solutions code verification of elastostatic solid mechanics problems in a commercial finite element solver," *Computers and Structures*, 229, 106175.

This article demonstrated rigorous code verification by MMS for difficult 3D solid mechanics problems with various FEM discretizations and nonlinear constitutive equations. MMS was once again shown to be sensitive to even minor errors in documentation of the continuum equations being solved, and applicable to commercial codes even when the source code is not accessible. Also, the authors used MMS to investigate the sensitivity of convergence order to quantitatively minor changes to the underlying mathematical model.

The authors conclude that MMS "provides a powerful tool for investigating whether the equations solved by the numerical model match those a user intends to solve. This is especially relevant for the verification of commercial codes..." They also discuss subtleties of implementing the MMS analytical source terms into commercial software. Overall, this paper is an exemplar of MMS concepts and is of interest to V&V specialists outside of solid mechanics.

For another application of MMS outside of code verification, see also Eça et al (2019) below under the "General" heading.

General

C. Beisbart and N. J. Saam (2019*),Computer Simulation Validation: Fundamental Concepts, Methodological Frameworks, and Philosophical Perspectives,* Springer-Verlag, Berlin, 2019.

This 1074 page encyclopedic compilation of 43 chapters by 53 contributors covers not only validation *per se* but prerequisite verification of codes and of solutions, plus related issues such as predictive capability, applications, credibility, stakeholder issues, statistical aspects, and philosophical considerations that are both serious and professional.

Oberkampf, W. L. (2019),"Simulation Accuracy, Uncertainty, and Predictive Capability: A Physical Science Perspective," Ch. 3 in Beisbart and Saam (2019).

Rider, W. J. (2019),"The Foundations of Verification in Modeling and Simulation", Ch. 11 in Beisbart and Saam (2019).

Roache, P. J. (2019), "Validation in Fluid Dynamics and Related Fields", Ch. 27 in Beisbart and Saam (2019).

Beisbart, C. (2019), "Should Validation and Verification be Separated Strictly,"Ch. 42 in Beisbart and Saam (2019).

Eça, L., Vaz, g., and Hoekstra, M. (2020), "A Contribution for the Assessment of Discretization Error Estimators Based on Grid Convergence Studies," *ASME Journal of Verification, Validation and Uncertainty Quantification*, Vol. 3,June, 2018, 021001.

This is an exceptionally thorough work on grid convergence studies using RANS turbulence models.

Eça, L. (2020), "2nd Workshop on the Assessment of Multivariate Metric for Validation at Multiple Set Points," *Proc. of V&V Verification and Validation Symposium, ASME V&V 2020*, May 20-22, 2020, Baltimore, USA.

Eça, K. Dowding, and P.J. Roache (2020), "On the Interpretation and Scope of the V&V20 Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer," *Proc. of V&V Verification and Validation Symposium, ASME V&V 2020*, May 20-22, 2020, Baltimore, USA.

Eça,L. (2020), "Overview of the 2018 Workshop on Iterative Errors in Unsteady Flow Simulations" *Proc. of V&V Verification and Validation Symposium, ASME V&V 2020*, May 20-22, 2020, Baltimore, USA..

Eça,L., G. Vaz, S. L. Toxopeus, M. Hoekstra (2019), "Numerical Errors in Unsteady Flow Simulations," *ASME Journal of Verification, Validation and Uncertainty Quantification*, June 2019, Vol. 4, DOI 021001.

This paper thoroughly considers *all* numerical errors including round-off, statistical, iterative, and time and space discretization errors. The authors make good use of the Method of Manufactured Solutions beyond the purpose of code verifications.

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

This ASME Standard presents a lucid and thorough illustration of V&V concepts applied in computational solid mechanics. Topics include development of a V&V plan,

model development, code verification, calculation verification, and two broadly differing approaches to validation, the first using only opinions of subject matter specialists to estimate uncertainties, the second using actual uncertainty data. Finally, in both approaches, the Validation Area Metric is used to assess the model. Unfortunately, this Area Metric may be shown to be misleading, as in the following manuscript (available from the author at hermosa@sdc.org).

Roache, P. J. (2020), "Critique of the Validation Area Metric and ASME V&V 10.1-2012." In review.