SANDIA REPORT

SAND2003-2752
Unlimited Release
Printed August 2003

On the Role of Code Comparisons in Verification and Validation

Timothy G. Trucano, Martin Pilch, and William L. Oberkampf

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

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of
Energy under Contract DE-AC04-94AL85000

Sandia National Laboratories

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

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

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

Telephone: (865)576-8401
Facsimile: (865)576-5782
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

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

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/ordering.htm

SAND2003-2752
Unlimited Release
Printed August 2003

On the Role of Code Comparisons in Verification and Validation

Timothy G. Trucano
Optimization and Uncertainty Estimation

Martin Pilch and William L. Oberkampf
Validation and Uncertainty Quantification

Sandia National Laboratories
P. O. Box 5800
Albuquerque, New Mexico 87185-0819

Abstract

This report presents a perspective on the role of code comparison activities in verification and validation. We formally define the act of code comparison as the Code Comparison Principle (CCP) and investigate its application in both verification and validation. One of our primary conclusions is that the use of code comparisons for validation is improper and dangerous. We also conclude that while code comparisons may be argued to provide a beneficial component in code verification activities, there are higher quality code verification tasks that should take precedence. Finally, we provide a process for application of the CCP that we believe is minimal for achieving benefit in verification processes.

Acknowledgements

The authors thank David Peercy, Steve Lott and Mark Christon of Sandia National Laboratories for reviewing a draft of this report.

Contents

Acknowledgements 4
Section 1 Introduction 9
Section 2 Formality of Code Comparisons 11
Section 3 Verification Using the Code Comparison Principle 13
3.1 Code Verification 14
3.2 Solution Error Estimation and Accuracy Verification 16
Section 4 Validation Using the Code Comparison Principle 19
Section 5 A Process for the Code Comparison Principle 21
Section 6 Potentially Useful Code Comparison Activities 27
Section 7 Conclusions 31
References 33

(Page Left Blank)

Figures

Figure 5.1 Key elements of a defensible code comparison process. 22
Figure 5.2 The multiplicity of challenges in using the CCP. 25

(Page Left Blank)

Section 1
Introduction

We present a perspective on the use of code-to-code comparisons, called ***code comparisons*** for short, in verification and validation (V&V) activities in this report. A "code" in our present usage is shorthand for the software implementation of a computational physics model, such as finite difference or finite element solution of the conservation laws of continuum mechanics. We will restrict the scope of our discussion to the case of comparison of two distinct and substantively different codes, denoted Code1 and Code2. We always assume that Code2 is the benchmark, or reference, code in a sense we will make more precise below. Code1 is always the subject of the code comparison exercise; it is intended that something be learned about Code1 through some kind of comparison with Code2. We recognize, however, this is not always the manner in which code comparisons are conducted. For example, in some cases no participating code is considered to be a benchmark. All of the conclusions in this report hold even more strongly for such a code comparison exercise. The strongest potential value for verification and validation relies upon an identified benchmark in the code comparison, and that is the underlying assumption that we apply.

When the "same physics" and "same algorithms" that are implemented in Code2 (phrases we often hear) are also implemented in Code1, we still consider these codes to be distinct. We thus include the "same physics" and "same algorithms" situation to be in the scope of this report. We note two special cases of code comparisons that are excluded from our discussion below. In the first case, it is a realistic possibility that "Code1" and "Code2" could be different dimensional options of the same specific code (e.g. 1-D vs 2-D, 2-D vs 3-D). In the second case, "Code1" and "Code2" could represent different meshing versions of the same specific code (e.g. hexahedral vs tetrahedral, or Eulerian vs Lagrangian). In each of these special cases, we believe that code comparisons, when performed carefully, are not only useful, but probably essential. All we will say about these cases in this report is that the process for performing code comparisons that we define in Section 5 below is clearly applicable to these situations and should, indeed, be applied.

When Code1 and Code2 are distinctly different codes, however, our view of the value of code comparisons is very different. It is our belief that in the absence of other, more carefully formulated V&V activities, code comparisons are dangerous and have little real value. When the reasoning for this viewpoint is explained and understood, we believe that it is logical and inevitable that code comparisons will be de-emphasized in formal V&V activities.

Stated even more explicitly, we claim the following:

- Code comparisons are not, strictly speaking, verification activities. They should not be used to replace one or more verification elements in a properly formulated verification plan.

- Code comparisons are not validation activities in any circumstances.

If code comparisons are used as part of a particular V&V activity, our recommendation is that they should be precisely defined and applied only as verification activities and that they should be performed along the lines of the process that we suggest in Section 5 below. Our general position on the issue of code comparisons is that code comparisons would be performed only as part of a larger program of independent verification tasks, such as application of software quality engineering (SQE) methodologies, algorithm testing procedures, verification test suites, and comparison with analytical solutions (see Oberkampf and Trucano, 2002; Oberkampf, Trucano and Hirsch, 2002). The Accelerated Strategic Computing Initiative (ASCI) V&V program at Sandia has elaborated concepts for appropriate and needed verification activities that support a formal validation process (Trucano, Pilch and Oberkampf (2003). For verification, code comparisons represent the analog of phenomena discovery experiments for validation. Since we have argued that phenomena discovery experiments provide little real value to rigorous validation goals and have analyzed this statement in a previous report (Trucano, Pilch and Oberkampf, 2002), we believe that the same conclusion is true for code comparisons in the verification arena.

We vigorously oppose any attempts to use code comparisons as substitutes for authentic validation tasks, as we will explain below.

Section 2
Formality of Code Comparisons

Use of comparison of the Code2 benchmark with the subject Code1 in verification activities rests upon the following trivial but formal rule:

Code Comparison Principle (CCP)

$$\left\|\mathrm{Code1}-\mathrm{Truth}\right\| \le \left\|\mathrm{Code1}-\mathrm{Code2}\right\| + \left\|\mathrm{Code2}-\mathrm{Truth}\right\|$$

Here, by “Code1” and “Code2” we mean any solutions output variables or functions of such variables that are compared mathematically and the difference quantified. This inequality is derived from the triangle inequality for norms. We emphasize norms (“metrics”) in the statement of the CCP because of the weight we have given to the rigor of comparison that should be applied in verification and validation in our previous writing (Oberkampf and Trucano, 2002; Oberkampf, Trucano, and Hirsch, 2002). We could also have used an equivalence relation (Simmons, 1963) instead, without changing the meaning of the CCP; and an equivalence relation might capture even better the logic underlying the usual application of code comparisons. Here, an example of an appropriate equivalence relation is “suitably accurate,” as measured by solutions to a set of test problems. “Truth” (see below) is then the correct solution to the problems. Code2 is equivalent to “Truth” if its solutions to the test problems are sufficiently close to the correct solutions (as defined by a verification metric, for example; Trucano, Pilch, and Oberkampf, 2003). Code1 is equivalent to Code2 if its solutions are sufficiently close to the solutions of Code2. If this is the case, it then follows that Code1 is equivalent to “Truth.” The alternative formalism that results in this case is:

$$\underline{\mathrm{Code2} \sim \mathrm{Truth}} \;\; \text{and} \;\; \underline{\mathrm{Code1} \sim \mathrm{Code2}} \;\; \text{implies} \;\; \underline{\mathrm{Code1} \sim \mathrm{Truth}}$$

We have used the word ***Truth*** in the CCP because we wish to concisely emphasize the distinction between a decisive benchmark and less appropriate information. It is perfectly appropriate to replace “Truth” with the phrase “Benchmark Information” or “Acceptable” or any other suitable word that the springs to mind. It is in this way that Code2 epitomizes its role as a benchmark. We fully recognize that no such thing as “Truth” exists in these matters, and nowhere in this argument is the notion of some kind of absolute truth required. “Truth” simply represents the information captured by Code2 that supports the belief that Code2 can be used as a verification benchmark in the application of the CCP. The notion of a logical equivalence relation captures this understanding more appropriately, but the norm formalism in our direct definition of the CCP more accurately captures the specific manner in which the CCP is applied in real code comparison activities.

Specifically in verification of Sandia ASCI codes, the meaning of “Truth” to us is “correct solution of the partial differential equations and the specified initial and boundary conditions.” The norm in the definition of the CCP then denotes any formal mathematical comparison the reader might wish to apply, especially as given by the principles detailed in Trucano, Pilch, and Oberkampf (2003) – but not qualitative comparisons such as the viewgraph norm (Trucano, Pilch, and Oberkampf, 2002).

To sum up, the entire focus of the CCP is to then argue that the left side of the CCP is small by arguing that the right side is small. Much of the time this argument is expressed in the following operational way: first, that it is “evident” or “well-understood” or “well accepted” that $\|\text{Code2} - \text{Truth}\|$ is small. Then, second, demonstrating that $\|\text{Code1} - \text{Truth}\|$ is small *mainly only requires* demonstrating that $\|\text{Code1} - \text{Code2}\|$ is small. We will now discuss this approach separately for both verification and validation.

Section 3
Verification Using the Code Comparison Principle

Oberkampf and Trucano (2002) discuss the proper elements of verification. It is convenient to use a classification of these elements that is introduced in that reference. Verification, according to the thinking of Oberkampf and Trucano, naturally falls into asking and answering two questions. First, is the software system that implements the algorithms intended to accurately numerically solve the partial differential equations defining a computational science conceptual model free of errors? This element is called ***code verification*** (published use of this term by others, including Roache, 1998, is discussed in Oberkampf and Trucano and we do not repeat that discussion here). The element of code verification encompasses two general classes of underlying activities. Oberkampf and Trucano define these classes as “Numerical Algorithm Verification” and “Software Quality Assurance.”

The second question that verification must address is whether a particular calculation of a specified problem is “correct.” More to the point as a matter of practicality, the question that must be addressed is really whether a particular calculation of a specified computational problem is “accurate enough.” The full resolution of this question for discrete algorithms which purport to solve systems of partial differential equations requires the activity of accuracy assessment on specified grids as well as evidence that the accuracy will improve as the discretization is refined (demonstration of convergence, for example). In the past we (and Roache, 1998) have referred to this element as ***calculation verification***. More recently (Oberkampf, Trucano and Hirsch, 2002) we have emphasized the intent by referring to this element as ***numerical error estimation***. For purposes of this document, we refer to this element as numerical error estimation.

Clearly, code verification and numerical error estimation are coupled. For example, our ultimate belief in assessment of accuracy for a particular calculation requires belief that the software (code) is verified. Otherwise, there is no basis for arguing that an accurate calculation, if such is the case, did not result from mutually canceling errors in the software implementation, such as an inadequate algorithm incorrectly implemented. On the other hand, a code that has a great deal of code verification evidence, such as might lead optimistic individuals to proclaim that the code was “verified,” has no guarantee of producing an accurate calculation in any specific circumstances. Accurate calculations depend on software fidelity and resolution. For example, because of computer resource limitations, a “verified” code may have to be applied to calculations with meshes that are too under-resolved to yield accurate answers. How one develops and trusts numerical error estimation for applications of computational science codes is the heart of the matter.

It is then fair to ask how the CCP may help resolve the questions of code verification and numerical error estimation.

3.1 Code Verification

First, consider the problem of code verification. Can the CCP be used to provide realistic evidence of algorithmic verification, software quality assurance, or both?

Software quality assurance (SQA) is virtually never the objective of code comparisons. Rather, SQA is centered on software engineering techniques that have no natural expression in terms of the CCP. Software reuse is an example that is a rather common practice. Suppose that a module is directly extracted from Code2 and implemented in Code1. (This is the most direct example of algorithm reuse, which is often very important in constructing new codes based on old codes.) Given such reuse of a module originating in Code 2, it always requires independent software engineering procedures to assess its implementation in Code 1 (for example, does it compile?). These procedures should be the same ones used to test the original implementation in Code 2. Simply comparing the two codes on one or more problems loses the power of the procedures that were originally applied to establish the benchmark quality of the Code 2 implementation. Such comparisons will also increase the amount of work performed.

For example, unit testing is a typical software engineering technique for testing the implementation of modules. Unit tests are chosen because their correct solution is independently known. If the module implementation in Code 2 is indeed an appropriate benchmark, and if unit testing was applied as part of the assessment of the module implementation in Code2, then what would be the point of applying the CCP to each unit test? Or, how would one decide which of a subset of unit tests to apply the CCP to? In fact, we claim that the last thing anybody should do is to compare Code1 results with Code2 results on unit tests. The appropriate SQA technique is to directly apply the unit tests to the Code1 implementation and skip the intermediate and less forceful step of some kind of code comparison.

This argument holds for the wide spectrum of software engineering based testing discussed in greater detail in Oberkampf and Trucano (2002). From another point of view, inferring code reliability from software testing should also involve probabilistic inference (see Singpurwalla and Wilson, 1999). Thus, given this point of view, code comparisons applied to test suites addressing SQA should also encompass statistical software reliability ideas. We have never seen this approach applied in any computational physics and engineering code comparison activity, either in the design of the activity or in the analysis of its results.

The CCP doesn't even make sense for other SQA techniques, such as complexity analysis or other static assessment procedures discussed, for example, by Hatton (1997).

The dominant role of the CCP for code verification is, in fact, algorithm verification. In this role, the CCP serves to define additional tests that populate the Verification Test Suite (VERTS) for Code1 (see Pilch, et al., 2001). For this purpose, Code2 must successfully assume the role of a trusted benchmark for the test problem defining the comparison. We emphasize that it is expected that the problem being solved does not have an analytic solution, or is solvable otherwise than through a code calculation. It is a complex problem by definition because it requires a Code2 calculation to define the benchmark. If the opposite were the case, Code1 would be directly compared with the analytic solution rather than with the Code2 solution.

The entire effort of comparing a Code1 calculation with a Code2 calculation as an element of the Code1 VERTS makes sense in direct proportion to the degree that we believe that Code2 is indeed a benchmark. This is not a matter of proclaiming Code2 to be a benchmark by definition. Rather, the fact of the matter is that a *lot of work* is required to declare Code2 to be a benchmark, especially for the purpose of some kind of code verification. This work must include significant effort to specify and document the resulting evidence of the correct implementation and functioning of Code2. Our position as stated in the Sandia V&V program has been that evidence that is not clearly described and documented is of little or no value (Trucano, Pilch and Oberkampf, 2002). Code2 is an appropriate benchmark for code verification as a VERTS element for Code1 only when we have accumulated and documented a scientifically defensible body of convincing evidence that Code2 has undergone independent code verification and is functioning properly on carefully designed test suites.

Unfortunately, it appears to often be the case that code comparisons are intended to short circuit the painstaking and labor-intensive accumulation of sound verification evidence for Code1. The CCP in reality offers the illusion of a labor- or budget-saving code verification procedure by focusing on the seemingly more constrained problem of estimating $\|\text{Code1} - \text{Code2}\|$. This approach is not acceptable for formal verification activities.

The fact remains that if a scientifically defensible code verification process has been applied to Code2, the same process should be directly applied to Code1 as well. The reason that the CCP may be chosen instead is either from the desire to reduce resource expenditures or because the verification process for Code2 may not be particularly well done or documented. We argue that code verification is a subject where you likely get what you pay for. Applying the CCP (mainly) because it saves time or money or both is neither compelling nor fulfilling.

We strongly believe that the CCP would be a less attractive option for code verification if visible evidence of the existence and results of the Code2 verification process exists. We presume that this evidence provides understanding and support for the belief that Code2 is indeed a benchmark. The accumulation of the same evidence for Code1 then seems to be demanded. As it is, applying the CCP because direct verification evidence is not visible invites the perverse belief that the attraction of code comparisons for code verification is in direct proportion to the *lack* of scientifically defensible evidence that $\|\mathrm{Code2} - \mathrm{Truth}\|$ is small. On the other hand, if substantial evidence exists that $\|\mathrm{Code2} - \mathrm{Truth}\|$ is small and if similar evidence is accumulated for estimating $\|\mathrm{Code1} - \mathrm{Truth}\|$, then estimating $\|\mathrm{Code1} - \mathrm{Code2}\|$ becomes simply extra and unneeded work and should not be done.

Our conclusion is that without additional systematic verification tasks, it is unlikely that the use of the CCP will provide credible evidence of code verification of Code1.

3.2 Solution Error Estimation and Accuracy Verification

Now consider the element of numerical error estimation, which focuses on the accuracy of specific calculations. For ASCI codes, numerical error estimation is typically achieved by demonstrating to a lesser or greater degree that the code converges to an answer as the grid is refined (Oberkampf and Trucano, 2002). Can we thus demonstrate that a specific calculation of Code1 is accurate (enough) through the use of the CCP?

Verification of numerical accuracy through the estimation of numerical error is easy to perform if we know what the exact solution of a problem is. Complex problems don't have the luxury of mathematically rigorous exact solutions. One needs codes to solve these problems. This leads to great practical difficulties associated with determining the accuracy of specific calculations for these applications. While there are techniques for attempting to characterize and estimate numerical accuracy in some generality, such as formal convergence analysis and *a posteriori* error estimation, it is true that some understanding of numerical error must also depend upon studies of specific complex test problems. Because complex test problems do not have analytic solutions, this is the area where use of the CCP is believed to have significant power. The reasoning is roughly as follows:

- For a given comparison problem, which could have been a previous application of Code2, Code2 defines the benchmark, in particular it is a numerical accuracy benchmark.

- Comparison of Code1 with Code2 for this problem then allows quantitative error assessment for Code1.

- Because the comparison problem is believed to be "relevant" or otherwise associated with a class of applications for Code1, the understanding of errors that results from the CCP is extrapolated to the class of applications and constitutes a statement of evidence about accuracy verification for Code1 for that class of applications.

We really face a conundrum. Our best chance for understanding numerical error is for test problems that are too simple to convincingly extrapolate to real applications. Complex test problems provide a much more convincing basis for extrapolation, but seemingly provide far riskier information about numerical errors. As described above, the use of the CCP seems to provide exactly what we need to break this conundrum. However, application of the CCP as argued above also begins to look like a self-fulfilling prophecy on these kinds of problems, because Code2 essentially is used ***as if by definition*** it specifies the "correct" solution (or solution with sufficiently small numerical error) of the problem. But does it?

For increasingly complex problems the bitter fact remains that it becomes increasingly difficult to show that $\|\text{Code2} - \text{Truth}\|$ is small. This undermines the basis for the reasoning detailed in the above bullets. By an extension of our arguments above, however, attempting to reduce the amount of work in verification leads to an even greater application of the fiat argument in this case. Those who support code comparisons for the purpose of calculation verification will argue that it is *self-evident* that Code2 is computing the problem correctly, or that Code2 at least establishes a relevant benchmark based on the "history" of its use. This argument is often made without presenting the critical and necessary evidence that Code2 has "converged" to the "correct" solution to begin with; or, since we don't know what the correct solution is but are using Code2 to define it, to at least demonstrate evidence of small numerical error. As we have emphasized in our recent writing on this topic (Oberkampf and Trucano, 2002; Oberkampf, Trucano, and Hirsch, 2002) Code2 numerical error estimation for the chosen comparison can ***only*** be based upon empirical demonstration of accuracy, not code developers' claims or informal legacy history.

In the absence of convincing accumulation of empirical verification evidence, this logic is too murky to hold up to rigorous scrutiny. One piece of evidence that suggests the appeal underlying code comparisons as elements of accuracy verification of complex test problem calculations is that most of these comparisons are simply code "bake offs" or beauty contests. A somewhat quantitative example (at least one doesn't have to look at side-by-side color shaded plots when one reads the paper) chosen at random is found in Rose (2001). This article addresses a specific code calculation of a difficult opacity benchmark problem and compares results with nine other codes (in the role of Code2), with no attendant discussion at all of calculation numerical accuracy. What is one supposed to make of this? That the author assumes that the nine Code2's are verified? That verification of the nine Code2's isn't worth discussing because that is self-evident?

Even given the philosophical limitations that we have stressed, benefits achieved from the use of the CCP for verification of complex problem numerical accuracy would likely increase if a rational methodology was consistently applied. In Section 5 of this report, we suggest an appropriate methodology to apply to code comparisons if, indeed, one must perform them despite the warnings we voice in this report. It should be of little surprise to the reader that our proposed methodology is directly taken from the experimental validation methodology that we have recently defined and published (Trucano, Pilch and Oberkampf, 2002).

For numerical error estimation of calculations, we will repeat the broad argument we made above in slightly different language. The crux of the matter for use of the CCP on complex calculations is that $\|\mathrm{Code2} - \mathrm{Truth}\|$ is shown to be small, *not* that $\|\mathrm{Code1} - \mathrm{Code2}\|$ is shown to be small. Establishing this "fact" requires a chain of logic and accompanying set of evidence, which we write as $\{\mathrm{Evidence}_1, \mathrm{Evidence}_2, \ldots, \mathrm{Evidence}_N\}$. If this was in fact the case and such a chain of logic and evidence existed, then the same chain of logic and procedures could be and should be applied to developing the same set of evidence for Code1. There would be no need to execute the CCP, nor would there be a perception that such a comparison would provide real value. However, when a fiat argument is used to "prove" that $\|\mathrm{Code2} - \mathrm{Truth}\|$ is small, then the CCP becomes attractive because investigation of how small $\|\mathrm{Code1} - \mathrm{Code2}\|$ is a simpler problem that requires fewer resources and less time.

Once again, the perverse fact remains that the CCP is more likely to be applied in solution accuracy assessment when the most critical information that the CCP relies on for scientific credibility, that $\|\mathrm{Code2} - \mathrm{Truth}\|$ is small, is missing.

There is one other danger associated with the use of the CCP when insufficient evidence exists that $\|\mathrm{Code2} - \mathrm{Truth}\|$ is small, especially when focusing on calculation accuracy. If it turns out that $\|\mathrm{Code1} - \mathrm{Code2}\|$ is large for a given problem then it is quite clearly dangerous to conclude that Code1 is wrong if one has not adequately demonstrated that $\|\mathrm{Code2} - \mathrm{Truth}\|$ is small. Exactly the opposite conclusion could be true instead. Code1 could have implemented an algorithmic correction that was neglected in Code2 that causes divergence in the results of the two codes. We believe that this problem is widespread, and leads to real difficulties in successfully concluding verification tasks that are CCP-centric. Especially when algorithms are substantially different between Code1 and Code2 the result of a divergence of their results seems to be never ending debate about which code is "correct." We argue that this question shouldn't even be asked in such a context. Only solid verification evidence that $\|\mathrm{Code2} - \mathrm{Truth}\|$ is small convincingly avoids this problem of drawing false conclusions from code comparisons.

We emphasize our fundamental point one more time. One of the biggest challenges that we face in verification is the understanding of just what aggregation of evidence is *sufficient* to claim that a code is verified and specific calculations are accurate. If Code2 is in fact claimed to be "verified" for justifiable reasons – in other words, because of an accumulation and documentation of a rigorous body of evidence – whatever approach led to this conclusion for Code2 is too valuable to not be applied to Code1. The CCP simply blurs the clarity and rigor of the process successfully used on Code2.

Section 4
Validation Using the Code Comparison Principle

When the norm at issue in the CCP is a validation metric (see Trucano, et al. 2001), the same general criticisms that we have presented above for verification can also be applied in exactly the same way. We will therefore not repeat the above arguments in a way that is specific to validation. But, we have a more grave criticism of the appropriateness of the CCP for validation that is different than the arguments used above for verification, and which is therefore worth emphasizing.

From its inception in 1999 the position of the ASCI V&V program at Sandia has been that ***validation is only accomplished through the confrontation of calculations with experimental data***. Experimental uncertainty characterization is a key component in performing high quality validation. Using Code2 as a benchmark for a CCP procedure in validation eliminates explicit attention on experimental uncertainty and is thus unacceptable. In fact, it may be the case that one reason that code comparisons are preferred in particular cases is because Code2 may so effectively *obscure* experimental uncertainty and provide a *fictitious* level of filtering of the data for benchmark purposes. Avoiding the need to deal with "dirty" experimental data may be desirable from certain perspectives, but it is completely inappropriate from a rigorous validation perspective.

Most of the time it is a severe mistake to believe that Code2 represents a significant interpolation or extrapolation of experimental data for complex problems. The ultimate form of a mistake along these lines is summarized by the pompous claim "Code2 is better than the experimental data." Such perceptions, if honestly held, usually arise from confusing ***calibration*** with ***validation***. Dealing with experimental uncertainty estimation, whether it is for calibration or for validation, is indeed difficult and it opens new and complex issues. However, the history of science has learned that experimental uncertainty must be dealt with. We believe that there is ultimately little logical basis for such fallacious claims, although we would not deny some potential for a limited version of them in the future in very specific circumstances. At best, just as for verification, a carefully constructed chain of evidence may have led to a rational basis for believing in Code2-based interpolation or extrapolation of experimental data. If this is the case, the process that accumulates this evidence should be applied directly to Code1.

A comparison with Code2 may serve as the basis for believing that Code1 is not modeling physics correctly. However, as is the case in verification discussed above, in the absence of carefully assembled and documented understanding of why Code2 is an appropriate validation benchmark, the attendant danger of applying the CCP is exactly as stated for verification. The truth may be that Code2 may be wrong while Code1 turns out to be correct.

Section 5
A Process for the Code Comparison Principle

Despite our analysis above, we recognize that it is unlikely that people will avoid the use of the CCP. Therefore, if code comparisons are going to continue to be performed at least there should be minimal expectations concerning the manner in which code comparisons are performed and results are presented. We believe that code comparisons should only be performed as a structured part of a spectrum of verification tasks, so that there is a significant body of evidence for verification in addition to only having code comparisons. We firmly believe that code comparisons should not be performed for validation. Finally, when code comparisons are performed we believe that their execution should mirror elements in a methodology that we have recently advocated for performing *experimental validation* (Trucano, Pilch and Oberkampf, 2002). The purpose of this section is to discuss this final point in greater detail.

Well-established scientific principles for *experimental validation* require:

- Experimental data of sufficient quality to perform the role of a benchmark.
- Logically defensible methods of comparing calculations with the benchmark experimental data.
- Logically defensible methods of drawing conclusions from the comparison of calculations with experimental data.

Similarly, *code comparisons* require the same approach, with an appropriate transcription of the basic meaning of the concepts. Thus, we argue that code comparisons require:

- Code2 has been thoroughly tested, documented and shown to be a benchmark. This means that $\|\text{Code2} - \text{Truth}\|$ has been systematically analyzed and evaluated using a wide range of procedures.
- Logically defensible methods of comparing Code1 calculations with Code2 calculations.
- Logically defensible methods of drawing conclusions from the comparison of Code1 calculations with the Code2 calculations.

Anything less cannot be scientifically defended and should not be undertaken.

Trucano, Pilch, and Oberkampf (2002) define the main elements of a methodology that addresses these concerns with regard to experimental validation. We have transcribed the validation emphasis of this methodology in their original report to an emphasis on code comparisons below. Figure 5.1 transcribes their fundamental diagram modified specifically to emphasize code comparisons. While other processes for performing code comparisons may be usable, we believe this process emphasizes elements that are important for performing a rational code comparison.

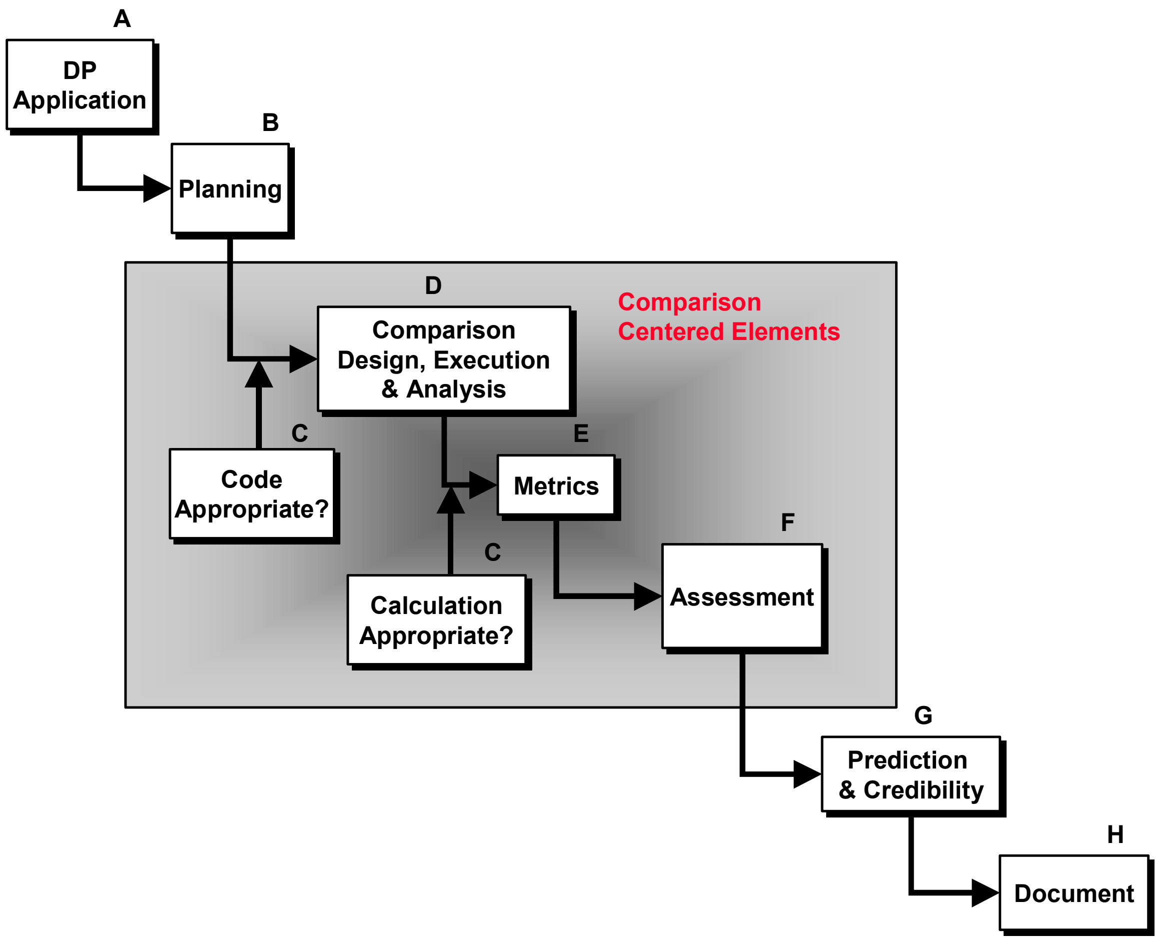

Figure 5.1 Key elements of a defensible code comparison process.

A. Defense Programs application requirements

All code comparison activities should have the goal of assessing credibility of a code for a given Defense Programs (DP) application. The constraints and requirements that emanate from the specification of the DP application influence the code comparison activity in exactly the same way that they influence experimental validation activities.

B. Planning

All code comparison activities require planning that is influenced by the intended DP application. All code comparison activities and planning should therefore be integrated in the overall V&V plan(s) through element G below for the given DP application. All code comparison activities should have specific technical plans associated with them, integrating means and ends and establishing priorities.

C. Appropriateness

Appropriateness plays the same role in code comparisons that code and solution verification do in experimental validation. Evidence of the appropriateness of Code1for undergoing a code comparison with Code2 should be accumulated and documented; this also involves asking this question about specific Code1 calculations too. In addition, evidence for the appropriateness of Code2 must be presented. This particularly centers on evidence of verification of Code2 and its benchmark calculations. We will discuss this issue further below.

D. Comparison design, execution, and analysis

The "experiment" is now the specific planned code comparison activity. The comparison should be designed, executed and analyzed in a scientifically defensible manner. A point of particular concern for this element is to quantify the uncertainty in the benchmark Code2 or, more specifically, the computational error in its particular calculations.

E. Metrics

Viewgraph norms are as unacceptable for code comparisons as they are for experimental validation. Code comparison metrics should be quantitatively precise and scientifically defensible as a means for comparing codes. We argue that rigorous metrics are more important for code comparisons given the likely difficulty in suitably quantifying the uncertainty in the benchmark.

F. Assessment

All code comparison metrics of code comparisons should be assessed using scientifically defensible means. Assessment especially must define quantitative measures of agreement for specific system response quantities. Assessment must also emphasize that precise and logical conclusions be drawn from the exercise of comparison, and whether the comparison is acceptable or unacceptable for the DP requirements. The whole point of a code comparison should be an underlying notion of precision. If one can't define precise assessment criteria for code comparisons then just what is the purpose of the activity?

G. Prediction and Credibility

The goal of code comparisons is to improve the credibility of the subject code for the stated DP application. The results of code comparison activities should therefore be cast in this light, i.e. the code comparison activities should clearly and directly relate to system response functions stated in the DP application. For example, use of the CCP in this case should be expected to contribute to our understanding of elements that influence predictive use of the code in interpolation and extrapolation, such us uncertainty quantification.

H. Documentation

Details of code comparisons should be traceable, reproducible, and fully documented. The consequences, and the means by which those consequences were determined, should be traceable and reproducible. Detailed documentation is essential for achieving traceable and reproducible code comparison, including, for example, input files and geometry specifications.

It is worth discussing more about the issue of "appropriateness" when one decides to apply the CCP. Figure 5.2 illustrates the resulting logical options in code comparisons that result from appropriateness or lack thereof. ***Appropriate*** in Figure 5.2 means that there is substantial evidence that the code is suitable for use in its defined role in the code comparison exercise. ***Inappropriate*** means that there is evidence that the code is not suitable for use in the code comparison exercise. A couple of simple but effective examples will make this clear. Appropriateness of Code2 means there must be a weight of evidence that it is a suitable benchmark. Inappropriateness of Code2 means that there is little or no evidence. What people think and "legacy" history is not evidence; evidence is documented and quantified.

More specifically, one could argue that there must be evidence that there are not software bugs in Code2 that will degrade the accuracy of its benchmark calculations in order for it to be appropriate for a code comparison exercise. This is indeed a complex problem to solve for the elaborate computational physics and engineering codes that are often most involved in code comparison exercises.

For example, Code1 is inappropriate for the comparison if bugs in the code prevent achieving the objective of the comparison. Suppose the purpose of the comparison is to compare a new algorithm in Code1 with an old algorithm in Code 2. Suppose further that Code1 has a data structure error that corrupts a database used in either the calculation or the post-processing of its results. When Code1 and Code2 are then compared, whatever the result is it is not relevant to the objective of comparing two algorithms because of the corruption of the comparison by the Code1 database error.

Code1 (or Code2 for that matter) could be inappropriate because of user errors in construction of input files. This would be comic if it did not happen so frequently, to be discovered only after intense effort to understand why the codes either agreed or disagreed.

There are a huge number of practical experiences that could be used to detail what we mean by "appropriateness" for the comparison. The present discussion is sufficient to make the point. It should be clear from Figure 5.2 that only two out of eight logical cases dealing with the issue of appropriateness, those cases where both codes are "appropriate," turn out to produce results that are defensible. In our opinion, this suggests that the odds are against code comparisons being fruitful for just this reason alone. Needless to say, confirming the appropriateness of the participating codes for the comparison activity is *not a result* of the CCP, it is *a necessary condition* for applying the CCP.

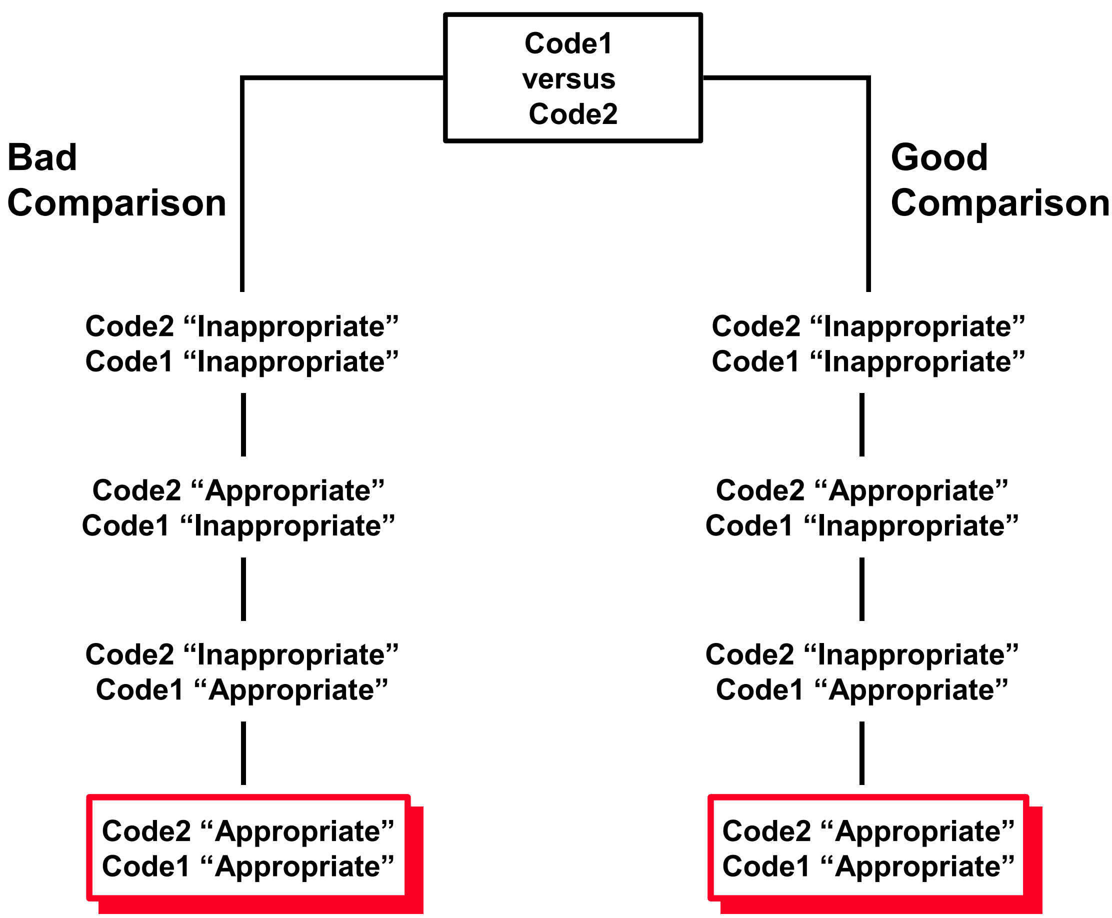

Figure 5.2 The multiplicity of challenges in using the CCP.

(Page Left Blank)

Section 6
Potentially Useful Code Comparison Activities

There are several areas where code comparisons *may* be a useful tool for achieving specific goals (not V&V!). Examples of such potential uses in our minds include:

1. Code1 calibration based on Code2 performance.

   This use performs calibration of Code1 parameters to achieve agreement with the results of Code2 on one or more problems. This is valuable in proportion to the degree that Code2 has been rigorously demonstrated to be a benchmark for the defined problems.

2. Code1 anomaly (unexpected gross failure) identification via comparison with Code2 under controlled conditions.

   This use is a testing technique, primarily aimed at identifying (probable) errors in Code1. A typical illustration of qualitative use of this technique is to execute Code1 for the same problem, with the same computational grid, as Code2. "Gross failure" means that Code1 doesn't even run the problem, while Code2 does, suggesting the presence of bugs in Code1 to be identified and removed. Note that we do not assume that Code2 runs the calculation correctly, which is why we do not claim that this is a verification activity. To the degree that it helps debug Code1, though, it might be a useful software development activity.

3. Use of multiple codes in a manner analogous to multiple "experimental facilities."

   Multiple codes and a careful code comparison exercise can be used to investigate possible bias errors due to widely varying models of physical phenomena. Investigations such as code comparison studies can be thought of as attempts to understand and quantify epistemic uncertainty (lack of knowledge uncertainty). Epistemic uncertainty is of particular concern and difficulty and the CCP may provide insight into its nature for particular simulation problems if carefully applied.

Code comparisons are also used in certain segments of the computational science community to understand ***uncertainty***, or potential uncertainty bounds, in complex modeling endeavors. It is easy to uncover published evidence of this approach to understanding complex systems, both physical and human. Astrophysics, for example, is a field that is dominated by speculative numerical modeling at its theoretical frontiers, simply because of the difficulty of performing controlled experiments, and the sparseness and intrinsic complexity of astrophysical data. Corresponding numerical model comparison activities clearly probe the level of model uncertainty (an example of epistemic uncertainty; Helton, 1997) that is present in fundamental astrophysics research.

A canonical example of the use of code comparisons in theoretical opacity models is found in Serduke et al. (2000). This paper briefly documents the latest in a series of opacity model comparison workshops that Serduke has organized for years. The clear intent of these workshops is to explore *uncertainty bounds* on opacity modeling, which is of importance in bounding stellar evolution models, supernova modeling, star formation, and so on. This activity is directly in line with the item #3 above. This paper is also revealing about why we would consider this activity to be an uncertainty estimation endeavor, and not a V&V task. While some control is exerted over the form of the model comparisons, there is in fact *no* benchmark identified (because there is none). Therefore, there are no formal means of drawing conclusions from the stated comparisons. The closest thing to a stated comparison benchmark is in fact a summary benchmark – the reported closeness of agreement of the various models, defined in specific ways. It suffices to quote the authors: "How close an agreement is good enough? Unfortunately, the answer depends closely upon the application." (Serduke et al., 2000, p. 532).

Some statistical analysis of these code comparisons is performed, which is certainly an improvement over other code comparison practice that we have observed over the years. But the published comparisons of one of the most important quantities (iron X-ray transmission in a temperature range that may be accessible to National Ignition Facility experiments; Lindl, 1998) in this paper are qualitative and difficult to definitively apply for purposes of V&V (Trucano, Pilch, and Oberkampf, 2002). As far as the real relevance to V&V goes, the authors appear to understand the core issue. Again we quote: "…Further development of experimental techniques and their application to a wide range of opacity problems is not only welcomed but ***essential*** [our emphasis] for continuing progress in the field." (Serduke et al., 2000, p. 540) In other words, the code comparison exercise has emphasized the need for useful and applicable experimental data. The real value of this published model comparison exercise is now obvious. By engaging in formal, controlled model comparisons, the resulting improved understanding of the *epistemic uncertainty* in current opacity models allows better prioritization and targeting of future experimental efforts. In this regard, this study is a useful example of a *helpful* code comparison exercise.

Earlier in this report we pointedly discussed the dangers of using the CCP for verification and validation per se. We also provide a specific warning regarding the use of the CCP for "code qualification." ***Code qualification*** is essentially a technical and management decision that a code is appropriate to use for a specific application. Such a decision can be based on many factors, depending on the approach chosen to make the decision. In our view, qualification can be and should be based on verification and validation; it is also true that the absence of appropriate verification and validation evidence could be neglected in a qualification decision. Instead, it might be the preference of the people who have to make this decision to base it on the conduct and results of code comparisons, or at least to make the CCP an important factor in qualification decisions. Because of the logical and operational weaknesses associated with the CCP that we have detailed above, we must emphasize that we disagree with such a basis for qualification and believe it to be dangerous.

(Page Left Blank)

Section 7
Conclusions

Code comparisons do not provide substantive evidence that software is functioning correctly (code verification). Instead, carefully planned, executed, and measured software verification procedures are required. If these procedures have been applied to Code2, the benchmark, they should also be directly applied to Code1. If this is the case, comparing Code2 with Code1 becomes extra, unnecessary work. If these procedures have *not* been applied to Code2, then comparing Code2 with Code1 is inconclusive, and probably dangerous, because there is insufficient scientifically credible evidence that Code2 is an appropriate benchmark.

Assessment of the numerical accuracy of calculations (for calculations that do not have analytic solutions) via comparison of Code1 with Code2 does not provide substantive evidence that Code1 calculations are accurate. Instead, a careful assessment of numerical error, relying upon convergence studies and empirical error estimation, is required. If this assessment has been performed for Code2 calculations, it should be performed for Code1 calculations. If numerical error estimation has not been performed for Code2, then there is insufficient scientifically credible evidence that Code2 is an appropriate benchmark.

The myth that we must recognize is that verification of Code1 software as well as verification of the accuracy of Code1 calculations can be placed on some kind of "resource discount plan" through the operation of the CCP. This myth rests firmly on the fiat argument that Code2 is believed to be a sufficient benchmark because of the vast amount of experience accumulated over the years using Code2, not because Code2 has been subjected to a stressing scientific verification process. Accumulated experience is an untrustworthy basis for drawing this conclusion because this "experience" is neither formally aggregated, nor quantified, nor documented. The proof of this lies simply in the fact that some users of Code2 are more trustworthy than others. This kind of "accumulated experience" is little better than a medieval guild. The real logic of the CCP in this circumstance is "I think Code2 works well, therefore I will use it as a benchmark for Code1."

In reality, well-designed code comparison procedures will *at most* produce evidence that Code1 is *not* functioning properly on specific calculations. In the absence of a credible basis for giving Code2 the status of a benchmark, we may compound our problems disastrously if we act as if Code1 is *wrong* simply because it produces a calculation that does not agree with Code2. However, the exact opposite could be the case – Code2 is *wrong* while CODE1 is *right*.

A second myth that is ever present is the belief that Code2 embodies wide physical modeling experience and understanding, thus allowing Code1 *validation* to be placed on the same kind of discount plan through operation of the CCP. A comparison with Code2 may indeed serve as the basis for believing that Code1 is not modeling physics correctly. In the absence of carefully assembled and documented understanding of why Code2 is an appropriate validation benchmark, the attendant danger of applying the CCP is exactly as stated for verification, especially in the case where Code2 may be wrong while Code1 turns out to be correct. The more complex the physics is, the weaker the argument for the CCP with regard to validation.

Finally, we think that it is wise to recall and stress Bill Rider's (of Los Alamos National Laboratory) Seven Deadly Sins of Verification (Kamm, 2002) when one considers applying the CCP:

The Seven Deadly Sins of Verification

Assume the code is correct.

Qualitative comparison.

Use of problem-specific settings.

Code-to-code comparisons only.

Computing on one mesh only.

Show only results that make the code "look good."

Don't differentiate between accuracy and robustness.

References

1. L. Hatton (1997), "The T Experiments: Errors in Scientific Software," IEEE Computational Science & Engineering, Vol. 4, No. 2, 1997; 27-38.

2. J. C. Helton (1997), "Uncertainty and Sensitivity Analysis in the Presence of Stochastic and Subjective Uncertainty," Journal of Statistical Computation and Simulation, Vol. 57, 1997; 3-76.

3. J. Kamm (2002), "Summary of the ASCI/NNSA Verification Workshop 2001," unpublished report, Los Alamos National Laboratory.

4. J. D. Lindl (1998), Inertial Confinement Fusion, Springer-Verlag, New York.

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

6. W. K. Oberkampf, T. G. Trucano and C. Hirsch (2002), "Verification, Validation and Predictive Capability in Computational Engineering and Physics," invited contribution to the Workshop "Foundations for Verification and Validation in the 21st Century," Johns Hopkins University and the Applied Physics Laboratory, Laurel, MD, October 22-23, 2002.

7. M. Pilch, T. Trucano, J. Moya, G. Froehlich, A. Hodges, D. Peercy (2001), "Guidelines for Sandia ASCI Verification and Validation Plans –Content and Format: Version 2.0," SAND2000-3101, Sandia National Laboratories, Albuquerque, New Mexico, Printed January, 2001.

8. P. J. Roache (1998), Verification and Validation in Computational Science and Engineering, Hermosa Publishers, Albuquerque, New Mexico.

9. S. J. Rose (2001), "The Radiative Opacity of the Sun Centre – A Code Comparison Study," Journal of Quantitative Spectroscopy & Radiative Transfer, Vol. 71, 635-638.

10. F. J. D. Serduke, E. Minguez, S. J. Davidson, and C. A. Iglesias (2000), "WorkOp-IV Summary: Lessons From Iron Opacities," Journal of Quantitative Spectroscopy & Radiative Transfer, Vol. 65, 527-541.

11. G. F. Simmons (1963), Introduction to Topology and Modern Analysis, McGraw-Hill Book Company, New York.

12. N. D. Singpurwalla and S. P. Wilson (1999), Statistical Methods in Software Engineering: Reliability and Risk, Springer-Verlag, New York.

13. T. G. Trucano, R. G. Easterling, K. J. Dowding, T. L. Paez, A. Urbina, V. J. Romero, B. M. Rutherford, and R. G. Hills (2001), "Description of the Sandia Validation Metrics Project," SAND2001-1339, Sandia National Laboratories, Albuquerque, New Mexico, Printed August 2001.

14. T. G. Trucano, M. Pilch, and W. L. Oberkampf (2002), "General Concepts for Experimental Validation of ASCI Code Applications," SAND2002-0341, Sandia National Laboratories, Albuquerque, New Mexico, Printed March 2002.

15. T. G. Trucano, M. Pilch, and W. L. Oberkampf (2003), “Verification Concepts Supporting Sandia National Laboratories Validation Activities,” to be published.

Distribution

EXTERNAL DISTRIBUTION

M. A. Adams
Jet Propulsion Laboratory
4800 Oak Grove Drive, MS 97
Pasadena, CA 91109

M. Aivazis
Center for Advanced Computing Research
California Institute of Technology
1200 E. California Blvd./MS 158-79
Pasadena, CA 91125

Charles E. Anderson
Southwest Research Institute
P. O. Drawer 28510
San Antonio, TX 78284-0510

Bilal Ayyub (2)
Department of Civil Engineering
University of Maryland
College Park, MD 20742-3021

Ivo Babuska
TICAM
Mail Code C0200
University of Texas at Austin
Austin, TX 78712-1085

Osman Balci
Department of Computer Science
Virginia Tech
Blacksburg, VA 24061

S. L. Barson
Boeing Company
Rocketdyne Propulsion & Power
MS IB-39
P. O. Box 7922
6633 Canoga Avenue
Canoga Park, CA 91309-7922

Steven Batill (2)
Dept. of Aerospace & Mechanical Engr.
University of Notre Dame
Notre Dame, IN 46556

Ted Belytschko (2)
Department of Mechanical Engineering
Northwestern University
2145 Sheridan Road
Evanston, IL 60208

James Berger
Inst. of Statistics and Decision Science
Duke University
Box 90251
Durham, NC 27708-0251

(2)
Laboratory for Computational Physics and Fluid Dynamics
Naval Research Laboratory
Code 6400
4555 Overlook Ave, SW
Washington, DC 20375-5344

Pavel A. Bouzinov
ADINA R&D, Inc.
71 Elton Avenue
Watertown, MA 02472

John A. Cafeo
General Motors R&D Center
Mail Code 480-106-256
30500 Mound Road
Box 9055
Warren, MI 48090-9055

James C. Cavendish
General Motors R&D Center
Mail Code 480-106-359
30500 Mound Road
Box 9055
Warren, MI 48090-9055

Chun-Hung Chen (2)
Department of Systems Engineering & Operations Research
George Mason University
4400 University Drive, MS 4A6
Fairfax, VA 22030

Wei Chen
Department of Mechanical Engineering
Northwestern University
2145 Sheridan Road, Tech B224
Evanston, IL 60208-3111

Kyeongjae Cho (2)
Dept. of Mechanical Engineering
MC 4040
Stanford University
Stanford, CA 94305-4040

Harry Clark
Rocket Test Operations
AEDC
1103 Avenue B
Arnold AFB, TN 37389-1400

Hugh Coleman
Department of Mechanical &
Aero. Engineering
University of Alabama/Huntsville
Huntsville, AL 35899

Raymond Cosner (2)
Boeing-Phantom Works
MC S106-7126
P. O. Box 516
St. Louis, MO 63166-0516

Thomas A. Cruse
398 Shadow Place
Pagosa Springs, CO 81147-7610

Phillip Cuniff
U.S. Army Soldier Systems Center
Kansas Street
Natick, MA 01750-5019

Department of Energy (4)
Attn: Kevin Greenaugh, NA-115
B. Pate, DD-14
William Reed, DP-141
Jamileh Soudah, NA-114
1000 Independence Ave., SW
Washington, DC 20585

Prof. Urmila Diwekar (2)
University of Illinois at Chicago
Chemical Engineering Dept.
810 S. Clinton St.
209 CHB, M/C 110
Chicago, IL 60607

David Dolling
Department of Aerospace Engineering
& Engineering Mechanics
University of Texas at Austin
Austin, TX 78712-1085

Robert G. Easterling
51 Avenida Del Sol
Cedar Crest, NM 87008

Isaac Elishakoff
Dept. of Mechanical Engineering
Florida Atlantic University
777 Glades Road
Boca Raton, FL 33431-0991

Ashley Emery
Dept. of Mechanical Engineering
Box 352600
University of Washingtion
Seattle, WA 98195-2600

Scott Ferson
Applied Biomathematics
100 North Country Road
Setauket, New York 11733-1345

Joseph E. Flaherty (2)
Dept. of Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12181

John Fortna
ANSYS, Inc.
275 Technology Drive
Canonsburg, PA 15317

Marc Garbey
Dept. of Computer Science
Univ. of Houston
501 Philipp G. Hoffman Hall
Houston, Texas 77204-3010

Roger Ghanem
Dept. of Civil Engineering
Johns Hopkins University
Baltimore, MD 21218

Mike Giltrud
Defense Threat Reduction Agency
DTRA/CPWS
6801 Telegraph Road
Alexandria, VA 22310-3398

James Glimm (2)
Dept. of Applied Math & Statistics
P138A
State University of New York
Stony Brook, NY 11794-3600

James Gran
SRI International
Poulter Laboratory AH253
333 Ravenswood Avenue
Menlo Park, CA 94025

Bernard Grossman (2)
Dept. of Aerospace &
Ocean Engineering
Mail Stop 0203
215 Randolph Hall
Blacksburg, VA 24061

Sami Habchi
CFD Research Corp.
Cummings Research Park
215 Wynn Drive
Huntsville, AL 35805

Raphael Haftka (2)
Dept. of Aerospace and Mechanical
Engineering and Engr. Science
P. O. Box 116250
University of Florida
Gainesville, FL 32611-6250

Achintya Haldar (2)
Dept. of Civil Engineering
& Engineering Mechanics
University of Arizona
Tucson, AZ 85721

Tim Hasselman
ACTA
2790 Skypark Dr., Suite 310
Torrance, CA 90505-5345

G. L. Havskjold
Boeing - Rocketdyne Propulsion & Power
MS GB-09
P. O. Box 7922
6633 Canoga Avenue
Canoga Park, CA 91309-7922

George Hazelrigg
Division of Design, Manufacturing
& Innovation
Room 508N
4201 Wilson Blvd.
Arlington, VA 22230

David Higdon
Inst. of Statistics and Decision Science
Duke University
Box 90251
Durham, NC 27708-0251

Richard Hills (2)
Mechanical Engineering Dept.
New Mexico State University
P. O. Box 30001/Dept. 3450
Las Cruces, NM 88003-8001

F. Owen Hoffman (2)
SENES
102 Donner Drive
Oak Ridge, TN 37830

Luc Huyse
Southwest Research Institute
6220 Culebra Road
P. O. Drawer 28510
San Antonio, TX 78284-0510

George Ivy
Northrop Grumman Information Technology
222 West Sixth St.
P.O. Box 471
San Pedro, CA 90733-0471

Rima Izem
Sience and Technology Policy Intern
Board of Mathematical Sciences and their Applications
500 5th Street, NW
Washington, DC 20001

Ralph Jones (2)
Sverdrup Tech. Inc./AEDC Group
1099 Avenue C
Arnold AFB, TN 37389-9013

Leo Kadanoff (2)
Research Institutes Building
University of Chicago
5640 South Ellis Ave.
Chicago, IL 60637

George Karniadakis (2)
Division of Applied Mathematics
Brown University
192 George St., Box F
Providence, RI 02912

Alan Karr
Inst. of Statistics and Decision Science
Duke University
Box 90251
Durham, NC 27708-0251

J. J. Keremes
Boeing Company
Rocketdyne Propulsion & Power
MS AC-15
P. O. Box 7922
6633 Canoga Avenue
Canoga Park, CA 91309-7922

K. D. Kimsey
U.S. Army Research Laboratory
Weapons & Materials Research
Directorate
AMSRL-WM-TC 309 120A
Aberdeen Proving Gd, MD 21005-5066

B. A. Kovac
Boeing - Rocketdyne Propulsion & Power
MS AC-15
P. O. Box 7922
6633 Canoga Avenue
Canoga Park, CA 91309-7922

Chris Layne
AEDC
Mail Stop 6200
760 Fourth Street
Arnold AFB, TN 37389-6200

W. K. Liu (2)
Northwestern University
Dept. of Mechanical Engineering
2145 Sheridan Road
Evanston, IL 60108-3111

Robert Lust
General Motors, R&D and Planning
MC 480-106-256
30500 Mound Road
Warren, MI 48090-9055

Sankaran Mahadevan (2)
Dept. of Civil &
Environmental Engineering
Vanderbilt University
Box 6077, Station B
Nashville, TN 37235

Hans Mair
Institute for Defense Analysis
Operational Evaluation Division
4850 Mark Center Drive
Alexandria VA 22311-1882

W. McDonald
NDM Solutions
1420 Aldenham Lane
Reston, VA 20190-3901

Gregory McRae (2)
Dept. of Chemical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139

Michael Mendenhall (2)
Nielsen Engineering & Research, Inc.
510 Clyde Ave.
Mountain View, CA 94043

Dr. John G. Michopoulos
Naval Research Laboratory,
Special Projects Group, Code 6303
Computational Mutliphysics Systems Lab
Washington DC 20375, USA

Sue Minkoff (2)
Dept. of Mathematics and Statistics
University of Maryland
1000 Hilltop Circle
Baltimore, MD 21250

Max Morris (2)
Department of Statistics
Iowa State University
304A Snedecor-Hall
Ames, IW 50011-1210

R. Namburu
U.S. Army Research Laboratory
AMSRL-CI-H
Aberdeen Proving Gd, MD 21005-5067

NASA/Ames Research Center (2)
Attn: Unmeel Mehta, MS 229-3
David Thompson, MS 269-1
Moffett Field, CA 94035-1000

NASA/Glen Research Center (2)
Attn: John Slater, MS 86-7
Chris Steffen, MS 5-11
21000 Brookpark Road
Cleveland, OH 44135

NASA/Langley Research Center (7)
Attn: Dick DeLoach, MS 236
Michael Hemsch, MS 280
Tianshu Liu, MS 238
Jim Luckring, MS 280
Joe Morrison, MS 128
Ahmed Noor, MS 369
Sharon Padula, MS 159
Hampton, VA 23681-0001

C. Needham
Applied Research Associates, Inc.
4300 San Mateo Blvd., Suite A-220
Albuquerque, NM 87110

A. Needleman
Division of Engineering, Box D
Brown University
Providence, RI 02912

Robert Nelson
Dept. of Aerospace & Mechanical Engr.
University of Notre Dame
Notre Dame, IN 46556

Dick Neumann
8311 SE Millihanna Rd.
Olalla, WA 98359

Efstratios Nikolaidis (2)
MIME Dept.
4035 Nitschke Hall
University of Toledo
Toledo, OH 43606-3390

D. L. O'Connor
Boeing Company
Rocketdyne Propulsion & Power
MS AC-15
P. O. Box 7922
6633 Canoga Avenue
Canoga Park, CA 91309-7922

Tinsley Oden (2)
TICAM
Mail Code C0200
University of Texas at Austin
Austin, TX 78712-1085

Michael Ortiz (2)
Graduate Aeronautical Laboratories
California Institute of Technology
1200 E. California Blvd./MS 105-50
Pasadena, CA 91125

Dale Pace
Applied Physics Laboratory
Johns Hopkins University
111000 Johns Hopkins Road
Laurel, MD 20723-6099

Alex Pang
Computer Science Department
University of California
Santa Cruz, CA 95064

Allan Pifko
2 George Court
Melville, NY 11747

Cary Presser (2)
Process Measurements Div.
National Institute of Standards
and Technology
Bldg. 221, Room B312
Gaithersburg, MD 20899

Thomas A. Pucik
Pucik Consulting Services
13243 Warren Avenue
Los Angles, CA 90066-1750

P. Radovitzky
Graduate Aeronautical Laboratories
California Institute of Technology
1200 E. California Blvd./MS 105-50
Pasadena, CA 91125

W. Rafaniello
DOW Chemical Company
1776 Building
Midland, MI 48674

Chris Rahaim
1793 WestMeade Drive
Chesterfield, MO 63017

Pradeep Raj (2)
Computational Fluid Dynamics
Lockheed Martin Aeronautical Sys.
86 South Cobb Drive
Marietta, GA 30063-0685

J. N. Reddy
Dept. of Mechanical Engineering
Texas A&M University
ENPH Building, Room 210
College Station, TX 77843-3123

John Renaud (2)
Dept. of Aerospace & Mechanical Engr.
University of Notre Dame
Notre Dame, IN 46556

E. Repetto
Graduate Aeronautical Laboratories
California Institute of Technology
1200 E. California Blvd./MS 105-50
Pasadena, CA 91125

Patrick J. Roache
1215 Apache Drive
Socorro, NM 87801

A. J. Rosakis
Graduate Aeronautical Laboratories
California Institute of Technology
1200 E. California Blvd./MS 105-50
Pasadena, CA 91125

Tim Ross (2)
Dept. of Civil Engineering
University of New Mexico
Albuquerque, NM 87131

J. Sacks
Inst. of Statistics and Decision Science
Duke University
Box 90251
Durham, NC 27708-0251

Sunil Saigal (2)
Carnegie Mellon University
Department of Civil and
Environmental Engineering
Pittsburgh, PA 15213

Larry Sanders
DTRA/ASC
8725 John J. Kingman Rd
MS 6201
Ft. Belvoir, VA 22060-6201

Len Schwer
Schwer Engineering & Consulting
6122 Aaron Court
Windsor, CA 95492

Paul Senseny
Factory Mutual Research Corporation
1151 Boston-Providence Turnpike
P.O. Box 9102
Norwood, MA 02062

E. Sevin
Logicon RDA, Inc.
1782 Kenton Circle
Lyndhurst, OH 44124

Mark Shephard (2)
Rensselaer Polytechnic Institute
Scientific Computation Research Center
Troy, NY 12180-3950

Tom I-P. Shih
Dept. of Mechanical Engineering
2452 Engineering Building
East Lansing, MI 48824-1226

T. P. Shivananda
Bldg. SB2/Rm. 1011
TRW/Ballistic Missiles Division
P. O. Box 1310
San Bernardino, CA 92402-1310

Y.-C. Shu
Graduate Aeronautical Laboratories
California Institute of Technology
1200 E. California Blvd./MS 105-50
Pasadena, CA 91125

Don Simons
Northrop Grumman Information Tech.
222 W. Sixth St.
P.O. Box 471
San Pedro, CA 90733-0471

Munir M. Sindir
Boeing - Rocketdyne Propulsion & Power
MS GB-11
P. O. Box 7922
6633 Canoga Avenue
Canoga Park, CA 91309-7922

Ashok Singhal (2)
CFD Research Corp.
Cummings Research Park
215 Wynn Drive
Huntsville, AL 35805

R. Singleton
Engineering Sciences Directorate
Army Research Office
4300 S. Miami Blvd.
P.O. Box 1221
Research Triangle Park, NC 27709-2211

W. E. Snowden
DARPA
7120 Laketree Drive
Fairfax Station, VA 22039

Bill Spencer (2)
Dept. of Civil Engineering
and Geological Sciences
University of Notre Dame
Notre Dame, IN 46556-0767

Fred Stern
Professor Mechanical Engineering
Iowa Institute of Hydraulic Research
The University of Iowa
Iowa City Iowa 52242

D. E. Stevenson (2)
Computer Science Department
Clemson University
442 Edwards Hall, Box 341906
Clemson, SC 29631-1906

Tim Swafford
Sverdrup Tech. Inc./AEDC Group
1099 Avenue C
Arnold AFB, TN 37389-9013

Kenneth Tatum
Sverdrup Tech. Inc./AEDC Group
740 Fourth Ave.
Arnold AFB, TN 37389-6001

Ben Thacker
Southwest Research Institute
6220 Culebra Road
P. O. Drawer 28510
San Antonio, TX 78284-0510

Fulvio Tonon (2)
Geology and Geophysics Dept.
East Room 719
University of Utah
135 South 1460
Salt Lake City, UT 84112

Robert W. Walters (2)
Aerospace and Ocean Engineering
Virginia Tech
215 Randolph Hall, MS 203
Blacksburg, VA 24061-0203

Leonard Wesley
Intellex Inc.
5932 Killarney Circle
San Jose, CA 95138

Justin Y-T Wu
8540 Colonnade Center Drive, Ste 301
Raleigh, NC 27615

Ren-Jye Yang
Ford Research Laboratory
MD2115-SRL
P.O.Box 2053
Dearborn, MI 4812

Simone Youngblood (2)
DOD/DMSO
Technical Director for VV&A
1901 N. Beauregard St., Suite 504
Alexandria, VA 22311

M. A. Zikry
North Carolina State University
Mechanical & Aerospace Engineering
2412 Broughton Hall, Box 7910
Raleigh, NC 27695

FOREIGN DISTRIBUTION:

Yakov Ben-Haim (2)
Department of Mechanical Engineering
Technion-Israel Institute of Technology
Haifa 32000
ISRAEL

Gert de Cooman (2)
Universiteit Gent
Onderzoeksgroep, SYSTeMS
Technologiepark - Zwijnaarde 9
9052 Zwijnaarde
BELGIUM

Graham de Vahl Davis
CFD Research Laboratory
University of NSW
Sydney, NSW 2052
AUSTRALIA

Luis Eca (2)
Instituto Superior Tecnico
Department of Mechanical Engineering
Av. Rovisco Pais
1096 Lisboa CODEX
PORTUGAL

Charles Hirsch (2)
Department of Fluid Mechanics
Vrije Universiteit Brussel
Pleinlaan, 2
B-1050 Brussels
BELGIUM

Igor Kozin (2)
Systems Analysis Department
Riso National Laboratory
P. O. Box 49
DK-4000 Roskilde
DENMARK

K. Papoulia
Inst. Eng. Seismology & Earthquake
Engineering
P.O. Box 53, Finikas GR-55105
Thessaloniki
GREECE

Dominique Pelletier
Genie Mecanique
Ecole Polytechnique de Montreal
C.P. 6079, Succursale Centre-ville
Montreal, H3C 3A7
CANADA

Vincent Sacksteder
Via Eurialo 28, Int. 13
00181 Rome
Italy

Lev Utkin
Institute of Statistics
Munich University
Ludwigstr. 33
80539, Munich
GERMANY

Malcolm Wallace
Computational Dynamics Ltd.
200 Shepherds Bush Road
London W6 7NY
UNITED KINGDOM

Department of Energy Laboratories

Los Alamos National Laboratory (53)
Mail Station 5000
P.O. Box 1663
Los Alamos, NM 87545
Attn: Peter Adams, MS B220
Mark C. Anderson, MS D411
Robert Benjamin, MS P940
Jane M. Booker, MS P946
Terrence Bott, MS K557
Jerry S. Brock, MS D413
D. Cagliostro, MS F645
Katherine Campbell, MS F600
David L. Crane, MS P946
John F. Davis, MS B295
Helen S. Deaven, MS B295
Barbara DeVolder, MS B259
Scott Doebling, MS P946
S. Eisenhawer, MS K557
Dawn Flicker, MS F664
George T. Gray, MS G755
Ken Hanson, MS B250
Alexandra Heath, MS F663
R. Henninger, MS D413
Brad Holian, MS B268
Kathleen Holian, MS B295
Darryl Holm, MS B284
James Hyman, MS B284
Valen Johnson, MS F600
Cliff Joslyn, MS B265
James Kamm, MS D413
S. Keller-McNulty, MS F600
Joseph Kindel, MS B259
Ken Koch, MS F652
Douglas Kothe, MS B250
Jeanette Lagrange, MS D445
Len Margolin, MS D413
Harry Martz, MS F600
Mike McKay, MS F600
Kelly McLenithan, MS F664
Mark P. Miller, MS P946
John D. Morrison, MS F602
Karen I. Pao, MS B256
James Peery, MS F652
M. Peterson-Schnell, MS B295
Douglas Post, MS F661 X-DO
William Rider, MS D413
Tom Seed, MS F663
Kari Sentz, MS B265
David Sharp, MS B213
Richard N. Silver, MS D429
Ronald E. Smith, MS J576
Christine Treml, MS H851
David Tubbs, MS B220
Daniel Weeks, MS B295
Morgan White, MS F663
Alyson G. Wilson, MS F600

Lawrence Livermore National Laboratory (21)
7000 East Ave.
P.O. Box 808
Livermore, CA 94550
Attn: Thomas F. Adams, MS L-095
Steven Ashby, MS L-561
John Bolstad, MS L-023
Peter N. Brown, MS L-561
T. Scott Carman, MS L-031
R. Christensen, MS L-160
Evi Dube, MS L-095
Henry Hsieh, MS L-229
Richard Klein, MS L-023
Roger Logan, MS L-125
C. F. McMillan, MS L-098
C. Mailhiot, MS L-055
J. F. McEnerney, MS L-023
M. J. Murphy, MS L-282
Daniel Nikkel, MS L-342
Cynthia Nitta, MS L-096
Peter Raboin, MS L-125
Kambiz Salari, MS L-228
Peter Terrill, MS L-125
Charles Tong, MS L-560
Carol Woodward, MS L-561

Argonne National Laboratory
Attn: Paul Hovland
Mike Minkoff
MCS Division
Bldg. 221, Rm. C-236
9700 S. Cass Ave.
Argonne, IL 60439

SANDIA INTERNAL

1	MS 1152	1642	M. L. Kiefer
1	MS 1186	1674	R. J. Lawrence
1	MS 0525	1734	P. V. Plunkett
1	MS 0525	1734	R. B. Heath
1	MS 0525	1734	S. D. Wix
1	MS 0429	2100	J. S. Rottler
1	MS 0429	2100	R. C. Hartwig
1	MS 0447	2111	P. Davis
1	MS 0447	2111	P. D. Hoover
1	MS 0479	2113	J. O. Harrison
1	MS 0487	2115	P. A. Sena
1	MS 0453	2130	H. J. Abeyta
1	MS 0482	2131	K. D. Meeks
1	MS 0482	2131	R. S. Baty

1	MS 0481	2132	M. A. Rosenthal
1	MS 0427	2134	R. A. Paulsen
1	MS 0509	2300	M. W. Callahan
1	MS 0645	2912	D. R. Olson
1	MS 0634	2951	K. V. Chavez
1	MS 0769	5800	D. S. Miyoshi
1	MS 0735	6115	S. C. James
1	MS 0751	6117	L. S. Costin
1	MS 0708	6214	P. S. Veers
1	MS 0490	6252	J. A. Cooper
1	MS 0736	6400	T. E. Blejwas
1	MS 0744	6400	D. A. Powers
1	MS 0747	6410	A. L. Camp
1	MS 0747	6410	G. D. Wyss
1	MS 0748	6413	D. G. Robinson
1	MS 0748	6413	R. D. Waters
1	MS 0576	6536	L. M. Claussen
1	MS 1137	6536	G. K. Froehlich
1	MS 1137	6536	A. L. Hodges
1	MS 1138	6536	M. T. McCornack
1	MS 1137	6536	S. V. Romero
1	MS 1137	6544	S. M. DeLand
1	MS 1137	6545	L. J. Lehoucq
1	MS 1137	6545	G. D. Valdez
1	MS 0720	6804	P. G. Kaplan
1	MS 1395	6820	M. J. Chavez
1	MS 1395	6821	M. K. Knowles
1	MS 1395	6821	J. W. Garner
1	MS 1395	6821	E. R. Giambalvo
1	MS 1395	6821	J. S. Stein
1	MS 0779	6840	M. G. Marietta
1	MS 0779	6840	P. Vaughn
1	MS 0779	6849	J. C. Helton
1	MS 0779	6849	L. C. Sanchez
1	MS 0778	6851	G. E. Barr
1	MS 0778	6851	R. J. MacKinnon
1	MS 0778	6851	P. N. Swift
1	MS 0776	6852	B. W. Arnold
1	MS 0776	6852	T. Hadgu
1	MS 0776	6852	R. P. Rechard
1	MS 9001	8000	J. L. Handrock
1	MS 9007	8200	D. R. Henson
1	MS 9202	8205	R. M. Zurn
1	MS 9005	8240	E. T. Cull, Jr.
1	MS 9051	8351	C. A. Kennedy
1	MS 9405	8700	R. H. Stulen
1	MS 9404	8725	J. R. Garcia
1	MS 9404	8725	W. A. Kawahara
1	MS 9161	8726	E. P. Chen
1	MS 9405	8726	R. E. Jones
1	MS 9161	8726	P. A. Klein
1	MS 9405	8726	R. A. Regueiro
1	MS 9042	8727	J. J. Dike
1	MS 9042	8727	A. R. Ortega
1	MS 9042	8728	C. D. Moen
1	MS 9003	8900	K. E. Washington
1	MS 9003	8940	C. M. Hartwig
1	MS 9217	8962	P. D. Hough
1	MS 9217	8962	K. R. Long
1	MS 9217	8962	M. L. Martinez-Canales
1	MS 9217	8962	J. C. Meza
1	MS 9012	8964	P. E. Nielan
1	MS 0841	9100	T. C. Bickel
1	MS 0841	9100	C. W. Peterson
1	MS 0826	9100	D. K. Gartling
1	MS 0824	9110	A. C. Ratzel
1	MS 0834	9112	M. R. Prairie
1	MS 0834	9112	S. J. Beresh
1	MS 0835	9113	S. N. Kempka
1	MS 0834	9114	J. E. Johannes
1	MS 0834	9114	K. S. Chen
1	MS 0834	9114	R. R. Rao
1	MS 0834	9114	P. R. Schunk
1	MS 0825	9115	B. Hassan
1	MS 0825	9115	F. G. Blottner
1	MS 0825	9115	D. W. Kuntz
1	MS 0825	9115	M. A. McWherter-Payne
1	MS 0825	9115	J. L. Payne
1	MS 0825	9115	D. L. Potter
1	MS 0825	9115	C. J. Roy
1	MS 0825	9115	W. P. Wolfe
1	MS 0836	9116	E. S. Hertel
1	MS 0836	9116	D. Dobranich
1	MS 0836	9116	R. E. Hogan
1	MS 0836	9116	C. Romero
1	MS 0836	9117	R. O. Griffith
1	MS 0836	9117	R. J. Buss
1	MS 0847	9120	H. S. Morgan
1	MS 0555	9122	M. S. Garrett
1	MS 0893	9123	R. M. Brannon
1	MS 0847	9124	J. M. Redmond
1	MS 0557	9124	T. G. Carne
1	MS 0847	9124	R. V. Field
1	MS 0557	9124	T. Simmermacher
1	MS 0553	9124	D. O. Smallwood
1	MS 0847	9124	S. F. Wojtkiewicz
1	MS 0557	9125	T. J. Baca
1	MS 0557	9125	C. C. O'Gorman
1	MS 0847	9126	R. A. May
1	MS 0847	9126	S. N. Burchett
1	MS 0847	9126	T. D. Hinnerichs
1	MS 0847	9126	K. E. Metzinger
1	MS 0847	9127	J. Jung
1	MS 0824	9130	J. L. Moya
1	MS 1135	9132	L. A. Gritzo
1	MS 1135	9132	J. T. Nakos
1	MS 1135	9132	S. R. Tieszen
20	MS 0828	9133	M. Pilch
1	MS 0828	9133	A. R. Black
1	MS 0828	9133	B. F. Blackwell

1	MS 0828	9133	K. J. Dowding
20	MS 0828	9133	W. L. Oberkampf
1	MS 0557	9133	T. L. Paez
1	MS 0847	9133	J. R. Red-Horse
1	MS 0828	9133	V. J. Romero
1	MS 0828	9133	M. P. Sherman
1	MS 0557	9133	A. Urbina
1	MS 0847	9133	W. R. Witkowski
1	MS 1135	9134	S. Heffelfinger
1	MS 0847	9134	S. W. Attaway
1	MS 0835	9140	J. M. McGlaun
1	MS 0835	9141	E. A. Boucheron
1	MS 0847	9142	K. F. Alvin
1	MS 0847	9142	M. L. Blanford
1	MS 0847	9142	M. W. Heinstein
1	MS 0847	9142	S. W. Key
1	MS 0847	9142	G. M. Reese
1	MS 0826	9143	J. D. Zepper
1	MS 0827	9143	K. M. Aragon
1	MS 0827	9143	H. C. Edwards
1	MS 0847	9143	G. D. Sjaardema
1	MS 0826	9143	J. R. Stewart
1	MS 0321	9200	W. J. Camp
1	MS 0318	9200	G. S. Davidson
1	MS 0847	9211	S. A. Mitchell
1	MS 0847	9211	M. S. Eldred
1	MS 0847	9211	A. A. Giunta
1	MS 1110	9211	A. Johnson
1	MS 1176	9211	L. P. Swiler
20	MS 0819	9211	T. G. Trucano
1	MS 0847	9211	B. G. vanBloemen Waanders
1	MS 0316	9212	S. J. Plimpton
1	MS 1110	9214	D. E. Womble
1	MS 1110	9214	J. DeLaurentis
1	MS 1110	9214	R. B. Lehoucq
1	MS 1111	9215	B. A.Hendrickson
1	MS 1110	9215	R. Carr
1	MS 1110	9215	S. Y. Chakerian
1	MS 1110	9215	W. E. Hart
1	MS 1110	9215	V. J. Leung
1	MS 1110	9215	C. A. Phillips
1	MS 1109	9216	R. J. Pryor
1	MS 0310	9220	R. W. Leland
1	MS 0310	9220	J. A. Ang
1	MS 1110	9223	N. D. Pundit
1	MS 1110	9224	D. W. Doerfler
1	MS 0847	9226	P. Knupp
1	MS 0822	9227	P. D. Heermann
1	MS 0822	9227	C. F. Diegert
1	MS 0318	9230	P. Yarrington
1	MS 0819	9231	R. M. Summers
1	MS 0819	9231	K H. Brown
1	MS 0819	9231	S. P. Burns
1	MS 0819	9231	D. E. Carroll
1	MS 0819	9231	M. A. Christon
1	MS 0819	9231	R. R. Drake
1	MS 0819	9231	A. C. Robinson
1	MS 0819	9231	M. K. Wong
1	MS 0820	9232	P. F. Chavez
1	MS 0820	9232	M. E. Kipp
1	MS 0820	9232	S. A. Silling
1	MS 0820	9232	P. A. Taylor
1	MS 0820	9232	J. R. Weatherby
1	MS 0316	9233	S. S. Dosanjh
1	MS 0316	9233	D. R. Gardner
1	MS 0316	9233	S. A. Hutchinson
1	MS 1111	9233	A. G. Salinger
1	MS 1111	9233	J. N. Shadid
1	MS 0316	9235	J. B. Aidun
1	MS 0316	9235	H. P. Hjalmarson
1	MS 0660	9514	M. A. Ellis
1	MS 0660	9519	D. S. Eaton
1	MS 0421	9800	W. Hermina
1	MS 0139	9900	M. O. Vahle
1	MS 0139	9904	R. K. Thomas
1	MS 0139	9905	S. E. Lott
1	MS 0428	12300	D. D. Carlson
1	MS 0428	12301	V. J. Johnson
1	MS 0638	12316	M. A. Blackledge
1	MS 0638	12316	D. L. Knirk
1	MS 0638	12316	D. E. Peercy
1	MS 0829	12323	W. C. Moffatt
1	MS 0829	12323	J. M. Sjulin
1	MS 0829	12323	B. M. Rutherford
1	MS 0829	12323	F. W. Spencer
1	MS 0405	12333	T. R. Jones
1	MS 0405	12333	S. E. Camp
1	MS 0434	12334	R. J. Breeding
1	MS 0830	12335	K. V. Diegert
1	MS 1030	12870	J. G. Miller
1	MS 1170	15310	R. D. Skocypec
1	MS 1176	15312	R. M. Cranwell
1	MS 1176	15312	D. J. Anderson
1	MS 1176	15312	J. E. Campbell
1	MS 1179	15340	J. R. Lee
1	MS 1179	15341	L. Lorence
1	MS 1164	15400	J. L. McDowell
1	MS 1174	15414	W. H. Rutledge
1	MS 9018	8945-1	Central Technical Files
2	MS 0899	9616	Technical Library