On the Automated Assessment of Nuclear Reactor Systems Code Accuracy

Robert F. Kunz, Gerald F. Kasmala, John H. Mahaffy, Christopher J. Murray
Applied Research Laboratory, The Pennsylvania State University, University Park, PA, 16804, USA, Tel.: 814-865-2144, Fax: 814-865-8896, e-mail: rfk102@psu.edu

1. ABSTRACT

An automated code assessment program (ACAP) has been developed to provide quantitative comparisons between nuclear reactor systems (NRS) code results and experimental measurements. The tool provides a suite of metrics for quality of fit to specific data sets, and the means to produce one or more figures of merit (FOM) for a code, based on weighted averages of results from the batch execution of a large number of code-experiment and code-code data comparisons. Accordingly, this tool has the potential to significantly streamline the verification and validation (V&V) processes in NRS code development environments which are characterized by rapidly evolving software, many contributing developers and a large and growing body of validation data.

In this paper, a survey of data conditioning and analysis techniques is summarized which focuses on their relevance to nuclear reactor systems (NRS) code accuracy assessment. A number of methods are considered for their applicability to the automated assessment of the accuracy of NRS code simulations, through direct comparisons with experimental measurements or other simulations. A variety of data types and computational modeling methods are considered from a spectrum of mathematical and engineering disciplines. The goal of the survey was to identify needs, issues and techniques to be considered in the development of an automated code assessment procedure, to be used in United States Nuclear Regulatory Commission (NRC) advanced T/H code consolidation efforts. The ACAP software was designed based in large measure on the findings of this survey. An overview of this tool is summarized and several NRS data applications are provided.

The paper is organized as follows: The motivation for this work is first provided by background discussion that summarizes the relevance of this subject matter to the nuclear reactor industry. Next, the spectrum of NRS data types are classified into categories, in order to provide a basis for assessing individual comparison methods. Then, a summary of the survey is provided, where each of the relevant issues and techniques considered are addressed. Several of the methods have been coded and/or applied to relevant NRS code-data comparisons and these demonstration calculations are included. Next, an overview of the basic design, structure and operational mechanics of ACAP is provided. Then, a summary of the data pre-processing, data analysis and figure-of-merit assembly processing elements of the software is included. Lastly, a number of NRS sample applications are presented which illustrate the functionality of the code and its ability to provide objective accuracy measures.

2. INTRODUCTION

In recent years, the commercial nuclear reactor industry has focused significant attention on nuclear reactor systems (NRS) code accuracy and uncertainty issues. To date, a large amount of work has been carried out worldwide in this area (e.g., Wilson et al., 1985, Ambrosini et al., 1990, D’Auria et al., 1990a, 1995a, 1995c, 1997, 2000, Schultz, 1993), with significant involvement by the NRC. Recently, the NRC has sponsored the present authors to:
1) Survey available data conditioning and analysis techniques, focusing on their appropriateness in NRS code accuracy and uncertainty assessment

2) Develop software to deploy recommended techniques

3) Develop coding and interfaces for the software to enable automated assessment on a large number of data sets so as to facilitate code update efforts and modeling revalidation

This paper documents these efforts. As an outcome of effort 2, the authors have developed a software platform, designated the Automated Code Assessment Program (ACAP). An overview of the design and operation of ACAP is provided, along with several NRS code application examples which illustrate the software’s capabilities. More recently, effort 3 has resulted in the development of a spreadsheet based batch capability for executing ACAP on a large number of data sets. This enables its use in rapidly providing an automated quantitative assessment of the change in the quality of a thermal-hydraulic analysis code from version to version. This aspect of our work is also documented here.

3. SYSTEMS CODE ACCURACY ASSESSMENT ISSUES

The commercial nuclear reactor industry came to focus on code reliability issues significantly (perhaps a decade) earlier than other Computational Fluid Dynamics (CFD) design industries (e.g., aerospace, automotive). There are several reasons for this including the inherent safety (and concomitant code reliability and licensing) concerns associated with nuclear reactors, and fundamental differences between the CFD methods used in reactor systems codes and “standard” multidimensional CFD methods.

Over a decade ago, the NRC initiated an international effort to improve and standardize the assessment of thermal hydraulic (TH) systems codes (Kmetyk et al., 1985, Bessett and Odar, 1986, for example). Prior to that, the assessment of the performance of TH codes had been largely qualitative and subjective, and thereby difficult to use in plant safety certification. In 1984, the NRC organized the International Thermal Hydraulic Code Assessment and Applications Program (ICAP), a major goal of which was the assessment of TH codes using relevant data from a wide range of international experimental facilities. Since that time, a large amount of research has been carried out worldwide in this area. As the NRC has moved towards plant certification based on best estimate methodology, the establishment of V&V guidelines that incorporate quantitative accuracy and uncertainty measures has become even more important.

The issues associated with NRS code accuracy and uncertainty assessment are numerous and complex. They include:

1) Scalability issues:
   - There is a paucity of experimental data taken in full scale hardware, so most code accuracy assessments are made against scaled test facility data since most NRS data is acquired in scaled facilities. Although the common theories of scaling are applied when designing the facilities, the facilities may not provide full dynamic similarity with a full scale reactor. In general, trends and the timing of key events in a NRS transient do scale well but in an effort to determine scaling effects, multiple scales are often utilized to generate the data used in assessment. These scaling complications have been addressed by D’Auria and his coworkers’ UMEA methodology (D’Auria et al., 1995b) and other related methods (Bovalini and D’Auria, 1992, D’Auria et al., 1995a).

2) Discretization and Model Setup Issues:
- NRS codes require nodalization of individual TH components. These discretizations are rarely grid converged in the conventional CFD sense due to the lumped parameter, quasi-1D modeling invoked in these codes. Indeed, it has been widely observed that significant differences in predictions can arise when different nodalizations are applied (Aksan et al., 1992, for example), and this has given rise to the growing practice of “qualifying a nodalization” against steady state data (Bonucelli et al., 1993).
- The selection of a computational time step and duration of a simulation affect the accuracy of a NRS code simulation.
- The specification of boundary conditions in an NRS transient can introduce uncertainty since often a “best” value is difficult to define, and an experienced user may plausibly select from a range of values. Different choices from within this range can yield significantly different predictions of key parameters (D’Auria and Galassi, 1997).

3) User Issues:
- The NRS code user can introduce uncertainty into a simulation, as evidenced by the widely shared observation that different users can easily produce different results using the same code applied to the same transient. Contributing to these differences are: the large number of available physical model specification input options, varying nodalization, time step and boundary condition selection (discussed above), and input errors (Aksan et al., 1992, D’Auria et al., 1990b).

4) Software Reliability
- The possibility of the presence of software/code errors including typographical or logical errors in a NRS code (especially a recently upgraded code being requalified) can introduce uncertainty into analyses, as can:
  - Compiler errors, roundoff errors, machine dependency
Mueller et al. (1982) provide assessment of the role of these “operational” uncertainties in NRS codes.

5) Best estimate vs. conservative criteria:
- In the last decade, NRC has begun to accept licensees’ analyses of best-estimate code results and corresponding uncertainty evaluations as information on which to base licensing decisions and to verify these submittals using best-estimate codes. This contrasts with the historical approach of using models which conform to conservative requirements (spelled out in Appendix K of 10CFR50, 1997). This move engendered “quantification of uncertainty” requirements on best estimate calculations being used for licensing purposes, as embodied within the Code Scaling Applicability and Uncertainty (CSAU) methodology and related approaches.

6) Key parameter selection:
- The assessment of a simulation which models the complete physics of a NR transient can only be assessed if a prioritization is given to some parameters over others. Guidelines have been established (Kmetyk et al., 1985) to identify the “key parameters” for a particular transient and particular reactor design. As a result, the code has to be assessed against each of the different sets of these key parameters for each of the identified transients for each of the reactor designs.
- Often NRS transients are characterized by multiple time ranges, each associated with quite different dominant physical mechanisms. Accuracy assessment must accommodate these since certain key parameters are only relevant in certain of these “time windows”. Unambiguous and generally applicable specification of these time windows is also difficult.

7) Richness of Data:
- As detailed in the next section, a wide variety of NRS data types are encountered including: single value key parame-
ters, timing of events tables, scatter plots, 1-D (in space) steady state data, and time record data.
- The latter of these are themselves characterized by a rich array of features.
This variety of relevant data types complicates accuracy evaluation and broadens the scope of automated code assessment procedures.

8) Inconsistency of comparison quantities
- There is, in general, not a one-to-one correspondence between available experimental and computed data. In particular, the same key parameters are not all measured in any given test program.
- There is, in general, not a one-to-one correspondence between measured and computed time and space coordinates. This can be due to stability limitations of the NRS code and/or nodalization choices. Interpolation may then be required for direct comparison of data and analysis which itself introduces uncertainty into the comparison.

9) Subjectivity of analysis – experimental comparison
- Recently, the NRC has used qualitative code-experimental comparison measures such as “excellent, reasonable, minimal and insufficient”. These are well defined (Damerell and Simons, 1993, Schultz, 1993). These measures allow a group of experts to study a set of results and produce some meaningful statement on code applicability for the particular plant and set of transients.
- The process is useful for major releases on a code, but is time consuming, especially for large test matrices.
- Eliminating the inherent subjectivity of this process is important in the NRC code consolidation effort. This would allow for code upgrades to be rapidly reassessed and for quantitatively tracking improvements in the code’s capability.

10) Uncertainty in experimental measurements
- Several investigators (Bessette and Odar, 1986, Coleman and Stern, 1997) have argued that experimental uncertainty must be considered in code-data comparisons, since simulation performance measures can be misleading when comparisons are made directly to reported measured values. Experimental uncertainty should be incorporated in code-data assessments to lessen the magnitudes of such difference measures.

11) Larger test matrices
- In the past, NRS code accuracy problems have been corrected in ways which have adversely impacted the comparisons of other untested transients. This has led the NRC to introduce much larger test matrices.
- This, of course, translates to a significant increase of code reassessment work in a development environment, and therefore itself motivates an automated code assessment process.

12) Lack of suite of assessment tools.
- Automated code assessment tools are not currently available for NRS code-data or code-code comparisons.
These issues collectively motivate the need for automated code assessment, in the NRC’s code consolidation effort and other system code development efforts, as well as in verification and validation and licensing application environments. Ideally, in the future, when NRS code users are involved in licensing calculations of “real” plant transients, a single post-processor would be deployed. Based on all uncertainties involved, this post-processor would return, at a given confidence level, the maximum expected deviation of several key parameters between code prediction and reactor behaviour (Wilson et al., 1985). The methodology embodied in this “ideal” post-processor must address each of the uncertainty
components summarized above. The need for such a methodology, has motivated a vast amount of research in the past decade (see D’Auria et al., 1995c for a review of much of this work). Some progress has been made in all of these areas, however, reliable and general tools to quantify NRS code accuracy are not available today. An important contribution to meeting this ideal would be a universally available assessment tool for the users of NRS codes to post-process results in a way that would return quantitative accuracy measures of code-data comparisons. Such a tool would only address some of the uncertainties in real plant analysis. However, it would be part of a process which validates a code with scaled facility data, contributing an important component to total uncertainty in full scale plant simulations. Also as the NRC pursues consolidation and advancement of a single NRS code, the need for such tools has never been greater, since such a tool would also greatly streamline revalidation against test matrix data.

It has been the overall goal of this research to initiate a software framework to automatically assess several of the NRS code uncertainty issues summarized above. In particular, a software package has been developed to objectively and quantitatively compare NRS simulations with data. This package, designated the Automated Code Assessment Program (ACAP) is described in detail below. Consistent with the observations made above, the code has been designed to:

- Tie into data bases of NRC test data and code results
- Draw upon a mathematical toolkit to quantitatively compare user specified data and analysis suites
- Return unambiguous quantitative figures-of-merit associated with individual and suite comparisons
- Incorporate experimental uncertainty in the assessment
- Accommodate the multiple data types encountered in NRS environments
- Reduce subjectivity of comparisons arising from the “event windowing” process
- Provide a framework for automated, tunable weighting of key parameters in the construction of figures-of-merit for a given test and in the construction of overall figures-of-merit from component code-data comparison measures
- Accommodate inconsistencies between measured and computed independent variables (i.e. different time steps)

So the ACAP development program addresses issues 6-12 summarized above. The scope of this project therefore did not include an attempt to quantify the uncertainties introduced by user training issues, discretization issues or code operational issues. Nor does the present work address quantification of uncertainty associated with physical models being used on a best estimate basis, nor on scaling uncertainties. However, the present investigators feel that with modest modifications ACAP could be applied parametrically to complement uncertainty assessment in each of these other assessment areas.

In summary, our fundamental goal has been to develop a numerical toolkit to analyze discrete computational and experimental NR systems data, and, in particular, to use these data analysis procedures to develop code-data and code-code comparison measures. Discrete data analysis is, of course, an important element in a wide array of technical disciplines. Indeed, data analysis methods are important anywhere experimental data are used. Techniques to analyze data samples or records lie within the scope of the three overlapping fields: probability and statistics, approximation theory, and time-series analysis. Accordingly, much of the information on this subject is embodied in the mathematics literature. Also, the needs of several engineering and scientific communities have motivated the development of data analysis techniques, which although falling within the three general categories mentioned, are characterized by unique or extended fea-
tures of relevance to the present research. In particular, methods developed in atmospheric/geologic sciences, economic forecasting, aerodynamic stability, demographics, digital signal processing, pattern (i.e., speech, optical, character) recognition and other fields have relevance to the analysis of NR systems data. Many of these methods, which are also surveyed here, are directly applicable or could be adapted to construct systems code-data or code-code comparison measures.

4. CATEGORIZATION OF NUCLEAR REACTOR SYSTEMS DATA

NRS data types are classified here into five categories, in order to provide a basis for assessing individual comparison methods. Specifically, scaled NR facilities are instrumented to provide a fairly wide array of key parameter and other data. These include:

I. Key parameters tables (Figure 1a).
II. Timing of events tables (Figure 1b)*.
III. Scatter plots of nominally 0-D data (Figure 1c)†.
IV. 1-D (in space) steady state data (Figure 1d).
V. Time record data (Figure 1e).

Each of these data types is potentially important in any particular NRS code analysis, and thereby must be considered in automated code assessment procedures. Experimental uncertainty bounds are often available for NRS data (see Figures 1c – 1e). The emphasis of this work has been on the latter three. In particular, general comparison measures for single valued key parameters and timing of events tables can be straightforwardly introduced into an automated code assessment system. For this reason, simple techniques to do this are not considered in this review. Somewhat more sophisticated mathematical techniques are required for analysis of data types III and IV, and data type V in particular provides a significant challenge for several reasons:

1) The ubiquitous appearance and relevance of these transient data in NR systems
2) The typically long record (often O(10^5) time steps) nature of these data, complicated significantly by their non-stationarity and diversity in characteristic features (e.g., long time scale damping, local quasi-periodicity, sudden changes due to active or passive phenomena, chatter (often of high amplitude), dependent variable limits (for volume fraction) between 0 and 1)
3) The significant differences that can appear between computed and measured time trace data (see Figure 1e)

The focus of this survey is on methods applicable to type V data, which include, as a subset, statistical and approximation methods that can be brought to bear on data types III and IV as well.

In order to facilitate the discussion of the data analysis methods below, some nomenclature definition is appropriate. Random data can be defined as data which, in the absence of measurement error, will be unique for each observation. Nearly all experimental data satisfy this definition of randomness. Experimental NRS transient data is random data since any time a given facility is run, the response of the system will not be exactly the same (non-deterministic). Experimental NRS transient data is also non-stationary since generally, the measured parameter cannot be described as having a constant
deterministic behavior at a given space-time coordinate.

* These data can be considered a subset of NRS data class I.
† Often these data are rendered “0-D” by collapsing data obtained at multiple space-time coordinates to a single scatter plot.
stant mean or autocorrelation function, that is, adjacent sections of the time trace will have different statistical measures.

It is not practical to repeat experimental transients enough times to generate a statistically significant ensemble. For this reason, there are not many practical techniques available to analyze non-stationary type V data (Bendat and Piersol, 1986), though some which do exist are reviewed below. This paucity of analysis techniques contrasts with the wide range of powerful tools available to analyze stationary random data. Fortunately, many of these techniques may be applied to non-stationary data with some loss of rigor, or through some “pre-processing” of the non-stationary records (to render the data globally or locally closer to stationary), or both.

NRS code data, interestingly, cannot be viewed as random data at all. In particular, multiple runs of a NRS simulation will return identical results each time. However, one can conceptualize performing multiple runs of a NRS code using varying boundary conditions well within the uncertainty bounds that these boundary conditions are known. These runs would produce an ensemble of time records. One can view an available record as a representative of this ensemble in the same fashion that the experimental data is assumed (by necessity) representative of an ensemble, were it available. So hereafter, we consider both experimental and computed NRS data, and the difference between them (hereafter the absolute error) as non-stationary random data.

Distinction is also drawn between dependent (measured physical) variables and independent (space-time coordinate) variables in NRS data. In NRS types IV and V data, the uncertainty associated with the independent variables is much smaller than that associated with the dependent variables. This limits what kind of data modeling approximations that are appropriate (Press et. al., 1994), and simplifies the consideration of experimental uncertainty (Coleman and Stern, 1997).

5. SUMMARY OF SURVEY

Data Analysis Methods

The data analysis methods surveyed herein, are classified into three broad categories: approximation theory based methods, time series data analysis methods, and basic statistical analysis methods.

The primary distinction between these categories of methods is the nature of the data to which they are applicable. These classes of methods are discussed here. For each, a brief overview of member techniques is provided. Several of these techniques have been adapted to NRS code-data comparison by other workers, and that literature is summarized. Discussion of the applicability of all reviewed techniques to NRS code assessment is provided. Several of the techniques are demonstrated through application to sample NRS code-data sets. The detailed mathematical prescription of the methods that have been chosen for incorporation into ACAP is provided in Kunz et. al, 1998b.

Approximation Theory Based Methods

Approximation theory encompasses mathematical techniques which provide useful (i.e. simple in some sense) functional approximations to discrete or continuous data. Approximation theory techniques for discrete data can be useful as quantitative comparison measures for NRS data since they approximate discrete random data using deterministic functions. The parameters (i.e. coefficients) defining the functions that approximate the data and the code results can be compared directly. Alternatively, figures-of-merit could be constructed using the parameters defining an approximation to the absolute error (i.e., its proximity to zero quantified in some way). These approaches are illustrated below.
The fundamental approximation problem for discrete data can be stated: Given a set of m data points \((f_i(x_i), i=1, m)\), find an analytical functional representation whose exact form (i.e., component magnitudes) is determined by minimizing in some sense the differences between this functional representation and the basis data. Here, we limit the scope of approximation theory discussion to single valued discrete functions of a single independent variable, as characterize types IV and V data. Type III data dismiss spatial-temporal dependence by collapsing the independent variable to a single scatter plot. Accordingly, these data cannot be interpreted as single valued (Methods related to the approximation theory techniques discussed here, but applicable to type III data, are treated in the Basic Statistical Analysis section below). Two subcategories of discrete approximation methods are best approximation methods and interpolation methods. The discussion here is limited to linear methods, that is methods based on linear combinations of basis functions.

The best approximation problem is characterized by an overdetermined system. Specifically, a functional approximation basis will have fewer degrees of freedom (say coefficients of an nth order polynomial) than the number of data points defining the discrete function to be approximated. The problem is then closed by minimizing an appropriate norm of the difference between the discrete data and approximating function. So the best approximation process involves: 1) Specification of a basis family of functions (e.g. polynomial, exponential), 2) Selection of appropriate norm(s) for assessing the accuracy of the representation and 3) Determination of functional coefficients which minimize the selected norms. It is important that both the basis functions and the norm selected in steps 1 and 2 be chosen with careful consideration for what the approximation is to be used for. In particular, basis functions should be selected that retain the important features of the data while ignoring the “noise” or unimportant features of the data.

By far the most employed norm in best approximation methods is the L_2 norm. Best approximation methods which employ minimization of an L_2 norm are termed least-square methods and are characterized by a minimum “energy” of total error, and overall efficiency of the method when orthonormal basis functions are used. If the chosen basis functions are linearly independent, and an L_2 norm is selected for minimization, the approximation problem involves the solution of the normal equations, an \(n \times n\) linear system, where \(n\) is the number of points defining the discrete function.

Other than L_2, norms often used for best approximation are the L_\(\infty\) and L_1 norms. The L_\(\infty\) norm has been widely used for discrete data approximation, with minmax or Chebychev basis polynomials. These polynomials have the desirable feature of the smallest (or nearly so in the case of Chebychev) maximum deviation (for a given polynomial order) from the approximated discrete function.

The L_1 norm minimizes the average absolute value of a functional approximation to discrete data and therefore can be a desirable minimization norm when a small percentage of the data can be deemed erroneous, as characterized by obvious deviation from trends set by the remainder of the data. This is because the effective weight given these points is smaller in the L_1 norm than L_2 and L_\(\infty\) norms.

In order to demonstrate the relative merits of these various norms, an NRS data example is provided here. In particular, a time segment of an OSU SBLOCA test from Lee and Rhee (1997) is considered. Figure 2a shows a plot of measured and RELAP5 predicted vessel pressure vs. time for the NRC12 case. In Figure 2b, the absolute error is plotted vs. a

\[\hat{.}\] Also termed regression methods.
normalized time coordinate. A quadratic fit was selected to represent the absolute error, and \( L_1, L_2 \) and \( L_\infty \) norms were used for minimization. The norm features described above are observable. In particular, the \( L_\infty \) norm fit responds to the very large spikes in error and thereby gives rise to an obviously poor fit. \( L_1 \) and \( L_2 \) norm fits are similar with \( L_1 \) responding less to the large spikes in absolute error early in the time segment, as expected.

In summary, best approximation methods define a subspace (basis) of possible approximations and the best approximation from this space is determined by minimization of an appropriate norm. Another approach to approximating a discrete function is to exactly fit a basis with \( n \) degrees of freedom to the \( n \) data points. This defines the interpolation methods subset of approximation theory. The most common of these is polynomial interpolation, where an \((n-1)\)th order polynomial is fit to \( n \) data points. Interpolation is obviously not appropriate for type III data since both variables in these sets are independent (and functional relationships between them are therefore not single valued). Interpolation can be relevant to automated code assessment for type IV data.

Polynomial interpolation can yield unrealistic variation between discrete data points (Runge phenomenon), especially when a large number of data are being fit (large \( n \)) and the interpolated variable spacing is uniform. This is often the case for types IV and V NRS data (Figures 1d, 1e) where \( \Delta x \) and \( \Delta t \) are typically constant or near constant, and records can be long (often \( O(10^5) \)). In general, polynomial interpolation is not a good choice for data characterized by sharp rises surrounded by weakly stretched curves, as can describe some types IV and V data. Also, for large \( n \), the polynomial interpolation problem can be computationally intensive.

Though many discrete functions cannot be adequately approximated using a single polynomial applied across its range, locally applied polynomial fits can effectively represent discrete data. Cubic splines are by far the most common of these methods. The compact support offered by cubic splines, and other related splines (some classes of B-splines, exponential splines) ameliorate the Runge phenomenon, and thus often return far more realistic function distributions between data pairs.

Figures 3a and 3b illustrate some of the above interpolation techniques for sample type IV NRS data. In particular, MIT-Siddique test data, digitized from Shumway, 1995, is approximated. In Figure 3a, the failings of a seventh order polynomial interpolated to the eight data pairs are observed. Unrealistic variations between pairs are observed for experimental, RELAP5 and absolute error. A standard cubic spline is applied in Figure 3b and this interpolation procedure is seen to provide a far more realistic distribution of the measured and computed quantities and the absolute error.

There appears to have been no direct application of approximation theory methods to types IV and V NRS data in the literature (though best approximation analysis has been used for type III data as discussed in the Basic Statistical Analysis section below). As just discussed and illustrated, there is a significant opportunity to usefully bring elements of approximation theory into NRS code-data and code-code comparisons. For example, low order polynomial best approximation with \( L_1 \) and/or \( L_2 \) minimization can be used to smooth and integrate NRS data type V absolute error. Also spline fits can be used to approximate type IV data. If applied to absolute error, such fits could also be integrated yielding figures-of-merit.

**Time-Series Data Analysis Methods**

Time-series data analysis techniques are designed to estimate properties of a measured or computed process from
a time series of repeated successive observations which are not necessarily independent. Time series data analysis techni ques are considered here for NRS type V data.

In general, data which are amenable to time-series analysis are those which can be modeled as stochastic processes, that is, processes which can be described using probabilistic laws. Time series methods are themselves broadly sub-classified between probabilistic methods and spectral methods. Both are considered here.

Probabilistic methods model processes based on assumptions concerning the nature of the process being studied, and using basic statistical measures. Most of these techniques are formulated for stationary processes, though a number of methods are available to transform data sets so as to render stationary techniques applicable (at least locally). These transformation approaches are discussed below. Assuming stationary data for now, the first step in the application of a probabilistic time series data analysis technique is the determination of an appropriate model of the process under consideration. Such models include purely random processes, moving average (MA) processes, autoregressive (AR) processes, random walk processes and more general combinations or extensions of these (e.g., ARMA, ARIMA). Particular classes of data are well described by particular process models. For example, economic data is often well suited to moving average process modeling.

Once a particular class of process model is selected, the model is “fit” to the data. Standard statistical measures (mean, variance, autocovariance) and other model coefficients are determined which define the fit. “Goodness of fit” measures are deployed (residual analysis) which can provide quantitative measure of how good the model has performed, and how reliable forecasting based on the model is.

The potential usefulness of probabilistic time series data analysis techniques to NRS data is demonstrated in Figure 4, where a “nearly stationary” segment of the OSU SBLOCA test introduced above is analyzed. Figure 4a shows the measured and RELAP5 predicted results between 10000 and 14000s for this case. In Figure 4b, the autocorrelation function of measured and computed pressure traces are plotted vs. time lag for the experimental data and RELAP5 simulation. Also appearing there is an approximate MA process fit to the RELAP-5 simulation. This fit models the data at a given time step as a weighted linear combination of the data values at some number of previous time steps. Two pieces of information are clearly accessible from the autocorrelation plots. First, variations in the experimental measurements are far more random in nature than the RELAP5 results in this region. The computed results show significant autocorrelation out to a lag of more than ten time steps. Second, it is observed that a MA process can do a good job of modeling this feature of the predicted transient.

For stationary or weakly non-stationary data, code-data comparisons of autocorrelation function can be made. In particular, the magnitude of autocorrelation function at a given time lag, or the integral of the autocorrelation or autocovariance to a given time lag can be compared. Alternatively, MA and other probabilistic time series data analysis models can be used to directly compare the computed and measured time histories through direct comparison of the coefficients of the process fitting procedure.

The other class of time series data analysis techniques is spectral techniques. In these methods, the time series is assumed to be composed of sin and cosine waves at different frequencies, that is, a process is modeled through assumed spectral characteristics as opposed to probabilistic characteristics. The most common spectral time series data analysis
methods are discrete Fourier transform techniques. These can be viewed as best approximation procedures using trigonometric basis functions (which form an orthonormal set) and employing $L_2$ minimization. Such techniques are ubiquitously applied in experimental methods, functional analysis and numerous other fields.

The discrete Fourier transform has been used in the NR community for automated code assessment by D’Auria and his coworkers (Ambrosini et al., 1990, D’Auria et al., 2000, for example). In their approach, the discrete Fourier transform of the measured and computed time trace is obtained. From the amplitudes of the component frequencies, two characteristic quantities are computed, the *average amplitude*, $AA$, and the *weighted frequency*, $WF$. The $AA$ sums the difference between experimental and code discrete Fourier transform amplitudes at each frequency. The $WF$ weights each frequency difference in the summation appearing in the $AA$ with the frequency itself. Each measure is non-dimensionalized. The $AA$ clearly provides a measure of the absolute amplitude error for a simulation, and $WF$ provides an indication of where the frequency errors are largest.

To illustrate this method, “artificial” data sets used by D’Auria and his colleagues have been reproduced in Figure 5a. Here an “experimental” transient and six “code” results, digitized from Ambrosini et al., 1990 are reproduced. The code results were originally selected to characterize a variety of code-data discrepancy features. In Figure 5b, the present authors have computed the $AA$ and $WF$ quantities for the six cases and these results closely correspond to those previously published, as expected.

In automated code assessment, the D’Auria FFT approach can be used to quantify code accuracy in a number of ways. For example, threshold “contours of acceptability” can be defined in the $AA$-$WF$ plane, each simulation then returning a single figure-of-merit which quantifies proximity to the origin. This is discussed further below.

Rigorous application of both probabilistic and spectral time series data analysis methods to automated code assessment is limited to stationary periodic data. In addition, spectral approaches which employ global transforms (such as the discrete Fourier transform) are well known to give poor representations of signals characterized by local phenomena. Indeed, square waves, reminiscent of the artificial experimental data in Figure 5a are often used to illustrate this (i.e., Gibb’s phenomena).

Despite such potential concerns, D’Auria’s discrete Fourier transform method has been effectively applied in obtaining information on code accuracy by several researchers in the literature. Accordingly, the present investigators have incorporated this method in ACAP.

**Basic Statistical Analysis Methods**

The two classes of methods considered so far encompass data analysis procedures that are inherently applicable to successive data. As such, the approximation and time series analysis methods model data in a fashion which describes discrete functional behavior with respect to time or space, making them more appropriate for types IV and V NRS data. Basic statistical analysis methods can also be brought to bear in analyzing NRS data. The field of statistics can be broadly defined to incorporate approximation theory and time series data analysis methods. Basic statistical methods are here distinguished as methods that describe random data in a fashion that is unconcerned with the spatial or temporal ordering of the data. Data are treated as a sample of $k$ observations of one or more variables. Index $k$ designates a running index over individual realizations in this data set. An example of data ideally suited to basic statistical description and analysis would
be the test scores and IQs \((x_k, y_k)\) for a sample of \(k\) students.

Single random variables are of fundamental concern in statistics. Here, a single variable, \(x_k\), say test scores, are sampled, and then standard descriptive measures of the sample are computed. Such measures include the mean, variance, median, skewness and other more arcane measures. For automated code assessment, these descriptive measures can be applied to the absolute error, and as such have been termed statistical \textit{difference} measures, and been widely used in the atmospheric sciences community (Fox, 1981, 1984, Wilmott, 1982, Rao, 1987, for example).

Also, multiple random variables can be identified with individual realizations (e.g., \(x_k = \text{test score}, y_k = \text{IQ}\)) and the relationships between these can be studied using correlation and regression procedures. Again, in concert with designations adopted by the atmospheric sciences community (ibid.), these methods are here termed statistical \textit{correlation} measures when applied to code-data comparisons. Predicted value and measured value are treated as paired random variables in these automated code assessment applications. Both statistical difference measures and statistical correlation measures are discussed here.

Straightforward application of basic statistical analysis methods, as just defined, dismiss spatial and temporal localization information. Data are considered from a basic statistical viewpoint as samples comprising one or two random variables (experimental value and/or computed value) with a priori notion of an independent variable ignored. Accordingly, if there are significant spatial or temporal trends in the data (as is the norm in NRS data), quantities like mean, standard deviation and correlation coefficient can be misleading and/or useless. However, if time trends can be removed, or if statistics measures are applied locally (in time), these techniques can provide, if not rigorous, at least useful information. Measures that preprocess the data so as to improve the stationarity assumption are discussed below. If time (or space) localization information is eliminated a priori (as is the case with NRS type III data), basic statistical measures can also be usefully applied.

A number of statistical difference measures have been applied in the NR community (Kmetyk et al., 1985, Wilson et al., 1985, Ambrosini et al., 1990, D’Auria, 1995a, for example) and in the atmospheric sciences community (Fox, 1981, 1984, Wilmott, 1982, Ku et al., 1987, for example). These include: 1) Mean error (or average absolute error), ME, 2) Variance of error (square of standard deviation), VE, 3) Mean square error, MSE, 4) Mean error magnitude, MEM, 5) Mean relative error, MRE. Measures 2 and 4 are closely associated with the \(L_2\) and \(L_1\) norms discussed above, respectively. Relative error measures normalize the absolute error by the local magnitude of the data (measured and/or computed). In addition to these basic difference measures, NR and atmospheric sciences workers have deployed other derived difference measures including: 6) Index of agreement (Wilmott, 1982), IA, 7) Systematic and unsystematic mean square error (Wilmott, 1982), \(\text{MSE}_{\text{S}}, \text{MSE}_{\text{U}}\), and 8) Mean fractional error (Ku et al., 1987), MFE.

These latter three non-standard statistical difference measures have some potentially appealing features for automated code assessment. In particular, the index of agreement distinguishes between the predicted and measured quantity in its definition, and has been defined as the “measure of the degree to which the observed [quantity] is accurately measured by the simulated [quantity]” (Ku et al., 1987). The index of agreement is non-dimensional. Systematic and unsystematic mean square errors measure, for the observed and predicted data respectively, difference from a linear least squares fit of their correlation. By introducing these two measures, and comparing their magnitudes to the mean square
error, one can determine how close the predictions are to “as good as possible”. This is illustrated below. The mean fractional error was defined in an attempt to reduce the bias afforded larger magnitude data by statistical measures based on absolute error, as well as the bias afforded smaller magnitude data by relative error based measures.

To illustrate the utility of these measures, they are each applied to a sample type III NRS data set. Figures 6a and 6b show sample data adapted from Shumway, 1995. These plots show comparisons of RELAP5 simulations of UCB wall condensation tests (separate effects tests that simulate PCCS conditions). For this demonstration calculation, these data were digitized directly from the printed reference and then analyzed. The descriptive measures introduced above were computed and are given in Table 1 for two RELAP simulations (which represented code runs that implemented default and “improved” diffusion models respectively).

### Table 1. Descriptive Statistical Measures for Type III Data.

<table>
<thead>
<tr>
<th>Descriptive Statistical Measure</th>
<th>RELAP5 – default diffusion</th>
<th>RELAP5 – new diffusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean error, ME (average absolute error)</td>
<td>-0.143 x 10^{-1}</td>
<td>0.755 x 10^{-2}</td>
</tr>
<tr>
<td>Variance of error, VE</td>
<td>0.131 x 10^{-2}</td>
<td>0.483 x 10^{-3}</td>
</tr>
<tr>
<td>Mean square error, MSE</td>
<td>0.130 x 10^{-2}</td>
<td>0.480 x 10^{-3}</td>
</tr>
<tr>
<td>Mean error magnitude, MEM</td>
<td>0.271 x 10^{-1}</td>
<td>0.159 x 10^{-1}</td>
</tr>
<tr>
<td>Mean relative error, MRE</td>
<td>-0.982 x 10^{-1}</td>
<td>0.953 x 10^{-1}</td>
</tr>
<tr>
<td>Index of agreement, IA</td>
<td>0.847 x 10^{0}</td>
<td>0.916 x 10^{0}</td>
</tr>
<tr>
<td>Systematic mean square error, MSE_{S}</td>
<td>0.236 x 10^{-3}</td>
<td>0.128 x 10^{-3}</td>
</tr>
<tr>
<td>Unsystematic mean square error, MSE_{U}</td>
<td>0.106 x 10^{-2}</td>
<td>0.352 x 10^{-2}</td>
</tr>
<tr>
<td>MSE_{S}/MSE</td>
<td>0.181 x 10^{0}</td>
<td>0.267 x 10^{0}</td>
</tr>
<tr>
<td>MSE_{U}/MSE</td>
<td>0.819 x 10^{0}</td>
<td>0.733 x 10^{0}</td>
</tr>
<tr>
<td>Mean fractional error, MFE</td>
<td>-0.378 x 10^{-1}</td>
<td>0.260 x 10^{-1}</td>
</tr>
</tbody>
</table>

These statistics consistently confirm the superiority of the new model. Several observations apply:

1) The ME, VE and MEM are significantly smaller for the new model.
2) VE and MSE are nearly identical owing to the small values of ME.
3) The ME and MRE indicate the degree of bias in the predictions. The tabulated values of ME suggest a significant average underprediction of the data for the original model and a small average overprediction for the newer model. The MRE is similar in magnitude for the two runs. This is a manifestation of the favoritism afforded the cluster of lower magnitude data for the original model. This is observable in Figures 6c and 6d which plots the UCB data absolute error. As discussed above, the MFE is a more consistent measure of bias. The ratio of MFE between the two models (1.45) lies between the ratio of ME (1.89) and MRE (1.03).

4) The IA is significantly better (i.e. closer to the perfect agreement value of 1.0) for the new model.

5) The new model predictive improvements quantified by the above measures are accompanied by an increase in the systematic component of the variance (increased MSE_{S}/MSE). This suggests that further improvements to the new diffusion model would likely be possible.
Statistical correlation measures can also provide quantitative descriptions of the correspondence between the data. In particular, the magnitude of correlation coefficient and a “goodness” measure for a polynomial fit could be used to quantitatively provide a figure-of-merit for code-data comparisons. For example, four linear statistical correlation measures were computed for the same UCB data set appearing in Figure 6: 1) the correlation coefficient, $\rho_{xy}$, 2) the $L_2$ norm of a linear least squares fit to the data, $L_2$-standard, 3) the $L_2$ norm of a linear least squares fit to the data constrained to pass through the origin, $L_2$-constrained and 4) the $L_2$ norm of the difference between the data and the “perfect agreement line” defined by $q^{\text{EXPT}} = q^{\text{RELAP}}$ $L_2$-deviation. This is a measure of absolute error. The calculated values of these measures appear in Table 2.

<table>
<thead>
<tr>
<th>Statistical Correlation Measure</th>
<th>RELAP5 – default diffusion</th>
<th>RELAP5 – new diffusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho_{xy}$</td>
<td>0.782</td>
<td>0.847</td>
</tr>
<tr>
<td>$L_2$-standard</td>
<td>0.421</td>
<td>0.227</td>
</tr>
<tr>
<td>$L_2$-constrained</td>
<td>0.421</td>
<td>0.235</td>
</tr>
<tr>
<td>$L_2$-deviation</td>
<td>0.465</td>
<td>0.265</td>
</tr>
</tbody>
</table>

The correlation coefficient shows modest improvement for the new model (i.e., it is somewhat closer to the perfect correlation value of 1.0), whereas all $L_2$ norms are significantly lower (indicating much better agreement). (The lines corresponding to the two least squares analyses performed and the “perfect fit” $q^{\text{EXPT}} = q^{\text{RELAP}}$ line appear in Figures 6a and 6b).

Basic statistical analysis measures should be employed with care for assessing type III data when, as in the above example, the mean value does not represent of an average of random process realizations. This is because data at various spatial locations are included, and significant spatial trends exist. In this circumstance, the mean and the variance cannot be deemed “good statistical estimators” (Bendat and Piersol, 1986) since the mean does not necessarily represent an expected value of absolute error. Similar serious difficulties arise for types IV and V data. For these, application of basic statistical analysis techniques again dismiss the temporal or spatial nature of the data. They are treated as random samples. This is appropriate only if the data is stationary. Otherwise, as above, even the basic statistical measures of mean and variance are of questionable merit.

Even if a particular NRS data set were stationary, and reasonable mean and variance values could be determined, it is not appropriate to assume any distribution of the absolute error about its mean. Accordingly, the most powerful aspect of basic statistical analysis techniques, statistical inference, cannot be deployed. Specifically, because we have no knowledge whatsoever regarding the probability density function of the absolute error about its mean, we cannot make assumptions on its form (say Gaussian, Student t, Chi-square). Therefore, we cannot establish uncertainty or confidence interval bounds on the absolute error. This makes it difficult to determine whether differences in basic statistical analysis measures are statistically significant enough to draw meaningful conclusions. The foregoing arguments are illustrated in Figures 6e and 6f. There, the probability density function (PDF) for the absolute error of the two UCB “samples” are seen to be nei-
ther consistent with one another nor with any well defined form. Nevertheless, assuming a normal distribution of the absolute error about its mean, a 95.4 (or $2\sigma$) confidence interval can be easily determined: $-0.087 \leq ME_{\text{default diffusion}} \leq 0.058$, $0.036 \leq ME_{\text{new diffusion}} \leq 0.052$.

Wilson et al., 1985 also assumed a normal distribution of absolute error about the mean of a locally near-stationary NRS data set, and then proceeded to construct a 95% confidence limit on the mean absolute error. They, of course, recognized the limitations discussed above and presented their results as “reasonable confidence limits [that] would be at the 95% level if [the absolute error was normally distributed stationary data].” Fox, 1980 also constructed confidence intervals for average absolute error in atmospheric analysis-data comparisons, but similarly noted, “If the assumptions concerning the use of the distribution upon which the interval construction is based are seriously violated, the interval statement itself will be inaccurate”. The present authors believe that the definitive unavailability of a known distribution function for absolute error in NRS code-data comparisons renders such statistic inference approaches inappropriate. Constructing statistical estimators for NRS data, as above, can provide a useful indication of code accuracy, but in the authors’ opinion, rigorous statistical inference measures should not be computed and used to assess code-data uncertainty. This position seems consistent with Wilmott’s, 1982 position: “Confidence bands and tests of statistical significance are not nearly as illuminating as an informed scientific evaluation of the summary and difference measures.”

In summary, the authors incorporated each of the statistical difference and correlation measures summarized above in ACAP. As discussed below, a code accuracy figure-of-merit is constructed based on some subset of these measures, but statistical inference, including the construction of significance measures, is not implemented.

**Trend Removal and Time Windowing**

As discussed above, a common characteristic in the application of time series data analysis and basic statistical analysis class methods to type V NRS data is that non-stationarity of NRS data renders many of the powerful methods within these classes less useful or inapplicable. Nevertheless, in the discussion and demonstration computations above, the application of several of these methods to “raw” NRS data was provided, and was seen to be of some use in automated code assessment. Two alternatives to this “apply-it-anyway” approach are:

1) Pre-processing of the NRS data, rendering it amenable to more rigorous application of time series data analysis and basic statistical analysis methods.

2) Application of methods expressly designed for non-stationary data.

Techniques in the first category include trend removal and time-windowing, and these are discussed here. Several non-stationary analysis methods are considered below.

There are a number of methods available for transforming data to more closely satisfy the stationary process assumption. Such trend removal techniques can therefore, in principle, increase the usefulness of time series data analysis and basic statistical analysis methods. These techniques are best applied when the non-stationarity is not of principal interest, as in the removal of “drift” from an experimental data set. Examples of techniques used in trend removal include simple curve fitting (i.e. best approximation methods from approximation theory discussed above), smoothing, high pass filtering, running averages, and others.

In trend removal, the modeled trend (or its deviation from its mean) is subtracted or filtered from the raw data. If
the removed non-stationarity is not of principal interest, then this information is discarded. However, if the trend itself is of importance (as is usually the case in NRS data), the “removed information” should be retained for concomitant analysis. In this light, the trend removal can be considered as a linear decomposition. For the present automated code assessment application, the authors believe that it is more general, and often appropriate, to assume that both stationary and non-stationary components of NRS data sets are important from an accuracy standpoint. Accordingly, separate data analysis techniques can be brought to bear on the stationary and non-stationary components of the time series. If the underlying assumptions of these separate analyses are not violated, they can be considered together in constructing a code-data comparison measure. This should yield a more rigorous, more robust (less susceptible to pathological exceptions) and more accurate comparison measure.

An approach to automated code assessment which accommodates this view is as follows. Trend removal is first performed on both experimental and systems code data. Time series data analysis and/or basic statistical analysis methods are then applied to the two residuals (raw data - trend) which should be “closer” to stationary than the absolute error, and at least have a mean much closer to zero than the absolute error. Approximation theory and/or basic statistical analysis comparison measures are then applied to experimental and computed trends. This approach is demonstrated here.

Figure 7a shows a comparison of the predicted and measured rod temperature in the core heatup and reflood stages of a FLECHT SBLOCA test vs. a TRAC-B simulation (Paige, 1998). A particular axial location in the core has been selected. For reference, the absolute error is plotted in Figure 7b. This data is clearly nonstationary, and though the basic trend of the data is well captured by the simulation, there is some underprediction of the peak rod temperature and some oscillatory features in the simulation, both of which should be captured quantitatively using an automated code assessment procedure.

The procedure outlined above was applied. A running average (of 80 time steps) was performed on the raw experimental and TRAC-B data sets. Figure 7c shows the computed trends arising from this process. The residuals of these two data sets (raw data - trend) are shown in Figure 7d. Inspection of Figures 7c and 7d clearly suggest the utility of trend removal in isolating two classes of discrepancy. The absolute error of the trend defines the overall peak rod temperature error level, and can be quantified using several of the approximation theory and basic statistical analysis methods described above. The residuals and their absolute error are plotted in Figure 7d are clearly more amenable to time series data analysis and basic statistical analysis than the raw absolute error plotted in Figure 7b. These methods provide figures-of-merit quantifying the significant oscillation appearing in the absolute error.

In this example, the trend removal process selected was not ideal for capturing the global trend associated with reflood. In particular, the running average smoothes the steep temperature drop features in both the measured and computed traces, as seen in Figure 7c, and this manifests itself in a transfer of error content from trend to residual in this region (t > 230 s). It is likely that an alternative global trend removal process could do a better job at capturing this local feature (e.g. smaller range running average or high pass filter). However, the time scale of this sharp descent feature (which has been chosen to be part of the global trend) is commensurate with the time scale of the oscillation feature in the absolute error (which has been designated part of the residual.) Global trend removal processes cannot therefore completely distinguish the two. This difficulty motivates the next topic, time windowing.
Time windowing, that is separating regions of the time trace prior to data analysis, can ameliorate some of the ambiguities associated with global approximation theory, time series data analysis or basic statistical analysis methods. Indeed, when the techniques defined so far, including trend removal, are successively applied to a few suitable, predefined time windows, more meaningful and robust comparison measures can be constructed. This is illustrated here.

Figure 7e shows the FLECHT data with two defined time windows associated with transitional and reflood segments of the SBLOCA. For the transition window, the same running average trend removal deployed above was used, but a smaller running average range (6 time steps) was used for the reflood window. The desired original trend in now well captured, as seen in Figure 7f. Also, Figure 7g illustrates that undesirable transfer of error content to the residuals has been mitigated. For this case, a reasonable choice for trend figure-of-merit is MRE. Reasonable choices for residual figures-of-merit are ME, AA and WF. These five figures-of-merit are given in Table 3.

Table 3. Figure-of-Merit for FLECHT Data.

<table>
<thead>
<tr>
<th>Figure-of-Merit</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRE$_{trend}$</td>
<td>.054</td>
</tr>
<tr>
<td>ME$_{residual}$</td>
<td>-0.2 °K</td>
</tr>
<tr>
<td>$\sigma = (VE_{residual})^{1/2}$</td>
<td>8.2 °K</td>
</tr>
<tr>
<td>AA$_{residual}$</td>
<td>1.1</td>
</tr>
<tr>
<td>WF$_{residual}$</td>
<td>29.5</td>
</tr>
</tbody>
</table>

The MRE$_{trend}$ captures the 5% average underprediction in trend in the transition region. The proximity of ME$_{residual}$ to 0 provides that the trend removal process was effective. The residual standard deviation of 8 °K is primarily due to the low frequency oscillation in absolute error. The AA value of greater than 1 indicates a discrepancy in average amplitude between the data and prediction larger than the average amplitude of the data residual itself. This discrepancy is contained in lower wave numbers and identified by the nondimensional WF of 29.5 (which indicates centering of AA near mode 30. There are 260 modes in the discrete Fourier transform).

Unfortunately, predefining time windows introduces, by definition, some subjectivity into the automated code assessment process. This issue has been treated extensively in the NR automated code assessment literature, and several investigators have concluded that time windowing is required (Kmetyk et al., 1985, D’Auria et al., 1995). In the view of the present authors, time windowing can be incorporated definitively in an automated code assessment process. Specifically, each experimental set in the automated code assessment data base can have associated with it predefined time windows. These ranges will be agreed upon for each test matrix trace prior to incorporation within the automated code assessment data base. The process for defining and achieving consensus on these is not treated here, but would presumably become part of a formal process in augmenting the automated code assessment data base. Once in the data base, these time windows become fixed. This approach eliminates subjectivity in the process. ACAP incorporates both trend removal and time windowing options, as summarized in Kunz et. al., 1998a, 1998b, 2000a.
Other Methods

Time-Frequency Methods

NRS data is characterized by non-stationarity, and this limits the applicability and power of many of the techniques introduced above. As discussed above, trend removal and time windowing can be effective in rendering the data closer to stationary. Alternatively, time-frequency methods are directly applicable to non-stationary data, and are therefore of interest here. Two principal techniques in this class are the short time Fourier transform and the wavelet transform.

The short time Fourier transform defines a time window which slides along the time trace. At each timestep, a discrete Fourier transform is applied to this local time window. From this, local frequency and phase content of the signal is obtained. If the sample size within the window is large and the trace within the window is near stationary, the short time Fourier transform will capture accurate, time localized spectral information of the signal. Such a time-frequency method ameliorates the problems associated with global transforms applied to data with local features (discussed in time series data analysis section above). This, of course, comes at the expense of increased dimensionality in the problem.

The short time Fourier transform is characterized as having a “fixed resolution” over the entire time-frequency domain. This manifests itself by limiting the short time Fourier transform analysis to having good temporal resolution or good frequency resolution (depending on choice of window size) but not both. This limitation is overcome using wavelet transforms. Wavelet transforms differ from short time Fourier transform methods in that their choice of basis functions are not necessarily sinusoidal, and their resolution effectively varies in the time-frequency plane. This allows for more accurate representation of features in a time trace than short time Fourier transform methods, especially when important features appear at widely varying times and/or frequencies **. These basic ideas are illustrated in Figure 8. In Figure 8a a segment of the same OSU SBLOCA data plotted in Figure 2a appears (experimental data only here). A short time Fourier transform and wavelet transform of this data appear in Figures 8b and 8c. These plots are spectrograms of the transforms, that is, contour plots of the square modulus of the transform coefficients in the time-frequency plane. Both transforms capture the higher energy associated with the oscillatory feature in the time trace, in a time localized fashion. The wavelet transform is seen to provide a better resolved representation of the feature.

There are a wide array of discrete and continuous wavelet transforms available, and the proper choice depends primarily on the nature of the features being extracted (Morlet continuous wavelet transforms were used for the results presented here). Some of these target features include local periodicity, local minima and maxima and, importantly, their variation in time. Accordingly, it is likely that a suite of wavelet transform tools could be used effectively in automated code assessment for NR applications (currently, the Morlet transform is available in ACAP).

The question arises as to what to do with the large amount of data that is generated by a wavelet analysis. One approach, devised here, and installed in ACAP, hybrids D’Auria’s method within a time-frequency approach. A wavelet transform of the artificial D’Auria data shown in Figure 5a was taken. Parameters analogous to D’Auria’s average amplitude and weighted frequency were constructed at each time step, and the locus of these points are plotted for each simulation in Figure 8d. This plot illustrates that accuracy can vary widely with time in the AA-WF plane. A scalar figure-of-

**.It is this feature that motivates the principal use of wavelets, data compression.
merit can be defined as the percentage of points that line on the origin side of a prespecified acceptability threshold contour. Clearly the choice of an appropriate “acceptance region” needs to be specified by the ACAP user and is dependent on the wavelet transform used, the nature of the data and the particular features that the user is interested in assessing for NRS code accuracy. A linear AA-WF plane acceptance boundary is defined in Figure 8d. Table 4 lists corresponding “percent acceptable” figures-of-merit.

Table 4. Wavelet Based Figure-of-Merit for D’Auria Data.

<table>
<thead>
<tr>
<th>Case</th>
<th>Percent Acceptable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>98</td>
</tr>
<tr>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>18</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
</tr>
</tbody>
</table>

So time-frequency and in particular wavelet analysis offers the possibility of mitigating some of the limitations of other methods considered above, including applicability to non-stationary data and improved capturing of time local features. These two benefits warranted the incorporation of wavelet transform techniques into ACAP.

Pattern Recognition and Multi-Variate Analysis Methods

Pattern recognition and multi-variate analysis methods can also, in principle, be usefully employed in NRS data accuracy quantification. Details of these methods and their possible adaptation for this use are provided in Kunz et al., 1998a.

Experimental Uncertainty

The NRC has emphasized for over a decade (Kmetyk et al., 1985, Bessette and Odar, 1986) the importance of including the contribution of experimental data uncertainty when assessing the accuracy associated with NRS code simulations. Indeed, the current NRC hierarchal quality definitions for code-data comparisons are specified with reference to the experimental uncertainty: “[for excellent agreement], the code will, with few exceptions, lie within the uncertainty bands of the data. Whereas for reasonable agreement, quantitative differences between code and data are generally observed to be greater than the experimental uncertainty,” (Damerell and Simons, 1993, Schultz, 1993). This issue has also received attention recently from the industrial CFD community at large (Coleman and Stern, 1997).

Experimental uncertainty is usually reported with a 95% confidence interval, that is, the true value of a quantity is expected to lie within ± the reported uncertainty of the reported data value 95% of the time. If the experimental uncertainty band is large, consideration of experimental uncertainty in the construction of code-data comparison measures lessens the significance and/or magnitudes of these measures. To illustrate this, consider an example adapted from Coleman and Stern, 1997 applied to the OSU SBLOCA data used in the examples above. Figure 9a shows predicted and measured integrated mass flow through an Automatic Depressurization System (ADS) for the NRC12 case. As can be seen in the figure, an experimental uncertainty bound is known for this quantity. If the absolute error is plotted vs. time and consid-
ered with the experimental uncertainty for this case, the absolute error is a less meaningful quantity. This is illustrated in Figures 9a and 9b. In particular, an artificial code solution was constructed which, as seen in Figure 9a, clearly exhibits significant differences from both RELAP and measured values (a straight line was taken for the artificial data). However, Figure 9b illustrates that the artificial solution cannot be deemed much less accurate than the RELAP simulation if taken in light of the experimental uncertainty.

This simple example serves to motivate the incorporation of experimental uncertainty in automated code assessment metrics. Building such measures into basic statistical analysis techniques is straightforward. Two possibilities are: 1) reporting mean error magnitude with the experimental uncertainty (MEMEU) or 2) constructing a “percent validated” (PV) metric defined as the percentage of the computed data that lies within the experimental uncertainty (0 ≤ PV ≤ 100). These definitions are consistent with the criteria used for code validation given by Coleman and Stern, 1997. For the data shown in Figure 9, these metrics are given in Table 5.

Table 5. Automated Code Assessment Measures Incorporating Experimental Uncertainty

<table>
<thead>
<tr>
<th>Statistic</th>
<th>RELAP5</th>
<th>Artificial Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEMEU</td>
<td>187 ± 250 kg</td>
<td>313 ± 250 kg</td>
</tr>
<tr>
<td>PV</td>
<td>73 %</td>
<td>66 %</td>
</tr>
</tbody>
</table>

These measures declare the RELAP simulation slightly superior to the artificial data, though both are seen to remain mostly within the experimental uncertainty. The MEMEU and PV metrics are less distinguishing than MEM alone, as desired.

For more sophisticated automated code assessment tools, incorporation of experimental uncertainty is not as straightforward. This is because for more refined error measures, the component contributions to experimental uncertainty must be individually ascertained in order to properly incorporate it within an automated code assessment measure. Consider an example where a zero drift experimental error gives rise to an experimental uncertainty that is of the same order of magnitude as well defined periodic features within the signal. This is illustrated in Figure 10a. Here, an experimental uncertainty of ± .3 is associated with the drift in the measured data. The two analyses shown return similar MEMEU and PV metrics (MEMEU₁ = .36 ± .3, MEMEU₂ = .37 ± .3, PV₁ = 46%, PV₂ = 50%). However, simulation 2 is clearly the superior one if taken in light of its capturing the dominant oscillatory feature of the data.

One approach to resolving this issue is to ascribe the experimental uncertainty to the trend in the data. Now, as indicated above, the automated code assessment data base must include, in addition to the raw experimental data, the experimental uncertainty. Also, trend removal and/or time windowing information may be included with each set. Therefore, including more detailed information on the experimental uncertainty (such as which component in the trend decomposition process it is associated with) is reasonable.

This proposal is illustrated in Figures 10b and 10c. There, trend removal has been performed on each of the signals appearing in Figure 10a. If the experimental uncertainty is ascribed entirely to the trend, MEMEU and PV measures for the two code-data trends are, as before, inconclusive (MEMEU₁ = .30 ± .3, MEMEU₂ = .32 ± .3, PV₁ = 50%, PV₂ = 42%). This says that the figures-of-merit associated with the trends do not establish significant superiority of either simu-
lation. However, examination of the residuals clearly establishes the superiority of simulation 2 over simulation 1. This is obvious upon inspection of Figure 10c. In an attempt to capture this quantitatively, several automated code assessment measures were brought to bear on this residual data. These are presented in Table 6.

### Table 6. Automated Code Assessment Measures Applied to Residuals in Figure 10c

<table>
<thead>
<tr>
<th>Figure-of-merit</th>
<th>Simulation 1</th>
<th>Simulation 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>ME</td>
<td>-0.0005</td>
<td>0.0039</td>
</tr>
<tr>
<td>MEM</td>
<td>0.19</td>
<td>0.24</td>
</tr>
<tr>
<td>$\sigma = (VE)^{1/2}$</td>
<td>0.21</td>
<td>0.30</td>
</tr>
<tr>
<td>AA</td>
<td>1.00</td>
<td>1.30</td>
</tr>
<tr>
<td>WF</td>
<td>15.3</td>
<td>13.5</td>
</tr>
<tr>
<td>$\rho_{xy}$</td>
<td>$0.4 \times 10^{-9}$</td>
<td>$-0.6 \times 10^{-3}$</td>
</tr>
</tbody>
</table>

Several interesting findings apply. First, the ME, MEM and VE measures are very poor indicators of the level of agreement. Indeed, simulation 2 exhibits apparently worse agreement with data if these measures were to be considered. This observed behavior is a manifestation of the slightly different frequency of the simulation 2 residual, which yields large absolute errors where the traces are locally out of phase. This again highlights the care which must be taken in deploying basic statistical analysis methods for automated code assessment. D'Auria's approach fares as poorly here. The AA for simulation 2 is larger than simulation 1. The explanation for this can be gleaned from Figure 10d, where discrete Fourier transforms for the three traces are shown. Again, the slight difference between measured and predicted frequency for simulation 2 gives rise to an AA which is larger than the AA for simulation 1. The difference in measured and predicted amplitudes, designated |absolute error| is included in the figure. The WF parameter captures the not-very-useful fact that the error for simulation 2 is centered at a slightly lower frequency than simulation 1. The final parameter appearing in Table 5 is the correlation coefficient. This parameter does a good job in illustrating the superiority of simulation 2.

In summary, it is important to incorporate experimental uncertainty in automated code assessment, and this motivated the authors to include the PV strategy (as well as time windowing and trend removal) in ACAP. Several general shortcomings with basic statistical analysis and spectral time series data analysis methods have been observed here, but selection of an appropriate metric (in this case correlation coefficient) brought to light the essential features in the comparison.

### 6. IMPLEMENTATION ISSUES

**Inconsistency of Comparison Quantities**

In the background section, two inconsistencies between computed and measured data were mentioned, which introduce some uncertainty into code-data comparisons. The first of these is that there is often not a one-to-one correspondence between available measured and computed dependent variables. This arises because all key parameters are not necessarily always measured in a test program. The second is that time and/or space coordinates in the NRS simulation may not be the same as those in the experiments.

From an automated code assessment standpoint, code and experimental data must both be available for a com-
parison. If not, a contributing figure-of-merit associated with the unavailable parameter cannot be constructed. This issue complicates the determination of the relative performance of a single code version applied to two similar facilities/tests if the same data is not available for the two. However, the main role of automated code assessment is to compare the relative performance of two similar versions of a code against a single facility/test or against each other. So the same figure-of-merit can always be consistently built for a given test.

The second code-data consistency issue mentioned above is within the scope of the present work. Specifically, data analysis modules of the automated code assessment procedure must accommodate, where needed, the differences between code and data space-time coordinates. This issue is relevant, of course, only to types IV and V data where discretization choices and/or numerical stability issues will generally return NRS predictions of dependent variables at different locations in space-time than where the data was taken.

Of the techniques analyzed, approximation theory and all of the basic statistical analysis methods considered except correlation measures, do not inherently require that experiment and computation have coincident independent variables. Such consistency is however required for basic statistical analysis correlation measures, probabilistic time series data analysis methods and the multivariate methods considered. Also, valid application of some trend removal processes including running averages require independent variable consistency. Discrete Fourier transform and time-frequency methods can be more accurately deployed if samples are taken at the same time steps, but this is not a requirement of their implementation.

ACAP incorporates a “resampling”/interpolation pre-processor for bringing experimental and computed data to the same independent variable basis. In particular, what we’ve loosely define here as “resampling” simply involves interpolating the systems code solution data onto the same set of time steps or spatial locations where experimental measurements are available.

Subjectivity Removal – Automated Construction of Figures-of-Merit

So far, a number of techniques for quantifying code-data or code-code comparisons have been summarized. Most of these techniques have been installed within a “toolkit” of assessment modules in ACAP. The input to each of these modules is the data to be compared; the output from each of them is one or more figures-of-merit.

ACAP constructs one (or at most a few) overall figures-of-merit defining the fidelity of a suite of NRS code runs applied to the automated code assessment data base. As mentioned in the previous sections, defining the best way to construct overall figures-of-merit is beyond the scope of the present work. Rather, we focused on providing a general software framework for doing so.

In application, an automated code assessment run will involve extraction of multiple code-data or code-code “raw” data sets. For each a number of data will be available, most generally several from each of types I-V. For each of these x vs. y sets, one or more comparison measures could be deployed, each returning a “local” figure-of-merit.

Willmott, 1982 and Fox, 1984 have both recommended that multiple difference based statistical accuracy indices should be presented when reporting model (i.e., simulation) performance. Within the scope of the present work, the authors have accommodated this philosophy by implementing a general figure-of-merit weighting construct in ACAP. Specifically, a single figure-of-merit for a given simulation can be constructed from an arbitrarily (i.e., user specified or
“canned”) weighted sum of several statistical accuracy measures. Removal of subjectivity is achieved once this figure-of-merit construction is frozen.

The overall figures-of-merit constructed in this process are to be interpreted as relative performance measures. These can then form the basis of acceptance/rejection tests in code revalidation. As relative measures, they must accommodate the basic requirement that superior solutions yield superior figures-of-merit. It has been observed above that a given figure-of-merit may or may not satisfy this basic “sanity” check depending on the application. The authors anticipate that a good deal of the effort involved in developing a robust figure-of-merit assembly procedure will be focused on satisfying this requirement.

7. ACAP PROGRAM DESCRIPTION AND MECHANICS

ACAP is a PC and UNIX station based application which can be run interactively on PCs running WINDOWS 95/98/NT or in batch mode on PCs as a WINDOWS console application or in batch mode on UNIX stations as a command line executable. The interactive and batch PC versions can be modified and recompiled from a WINDOWS “folder” under the Microsoft Visual C++ environment. The batch UNIX version can be modified and recompiled using any C++ compiler which conforms to the C++ draft standard (including the freely available g++/gcc compilers).

Interactive Mode Execution

A brief summary of the operation of ACAP is provided here. Figure 11 shows a schematic overview of the structure of the code. Experimental and computational NRS data are input through ACAP data files, which, in their simplest form, contain a table of x-y data and a few data descriptor keywords. The user specifies, either interactively or through front end script files, a suite of data conditioning and data analysis methods to be deployed in quantifying the correspondence between the measurements (if available) and the (one-or-more) simulation data sets. This suite of methods is termed the ACAP configuration, which can be saved in a file for later use on the current or other data sets. In interactive mode, ACAP displays the data sets with a modest but reasonably versatile embedded plotting package, and provides standard windows environment interfaces to select and adapt the mathematical methods to be deployed. The code then executes specified data conditioning processes and data comparison measures. Lastly, with user selected weighting, also part of the configuration, an overall figure-of-merit is constructed quantifying the accuracy of the individual code runs. The results of the ACAP session, including a summary of all selections made, and the component and overall figures-of-merit are output to screen and file. Figure 12 illustrates several elements of the interactive ACAP interface for an application of the software to the "D'Auria" data (Ambrosini et al., 1990). Complete documentation for the ACAP software is available in the ACAP User’s Manual (Appendix A of Kunz et al., 1998c).

Batch Mode Execution

As discussed above, users must take care in assembling robust ACAP configurations, in order that returned FOMs reliably quantify the improvements or degradation in model upgrade/code version. The interactive mode for ACAP is preferred in constructing these configurations on a "new" test matrix data set. This is because the ACAP GUI allows one to effectively visualize and interact with the systems code and experimental data (i.e. try different conditioning and comparison strategies) until a satisfactory configuration is established. Once a configuration is established for a given test suite entry, it becomes, in principle, frozen in time. Subsequent reassessments of code versions are then more efficiently
carried out in batch mode.

ACAP has the potential to significantly streamline NRS code development efforts. The development environment for such software is characterized by rapidly evolving software (i.e. frequent updates), many contributing developers and a large (and growing) body of validation data. As each new version of a NRS code is proposed for release, it is important that a revalidation process is undertaken, on some level, to ensure that new modifications have not “broken” some of the required application capabilities of the code. Such revalidation is a major element in the configuration control of the NRC’s consolidated code, and allows for quantitative tracking of improvements in the code’s capability. ACAP, running in batch mode, has great potential to further expand this role in the development process.

Accordingly, batch execution of ACAP is provided as an option within the auto-validation tool (Auto-DA), currently in use at NRC. As also illustrated schematically in Figure 11, this tool automatically runs systems code simulations for a sequence of test cases and generates a prespecified series of plots, using xmgr5 (Anon., 1999), which include experimental measurements and the results of the multiple simulation runs. The Auto-DA utility, which is comprised of two PERL scripts, has been extended to optionally execute ACAP for each test case, so that attendant to each test case and plot is the figure-of-merit output of the ACAP session.

This Auto-DA/ACAP batch capability is illustrated here with an example application. Elements of this example are provided in Figure 13. Figure 13a shows the Auto-DA "path" spreadsheet page (Microsoft Excel shown here), which points to two TRAC-M executable version file paths and the ACAP executable file path. The Auto-DA "cases" spreadsheet page is shown in Figure 13b. There, the "CaseIDs" are identified, Demo1, Demo2, Demo3. Two rows and associated TRAC-M version are included for each case, indicating that both the baseline and a "new" version of TRAC-M are to be executed for this case. The "Base" column indicates with an "X" which run is to be deemed the base case against which all other runs for that case (only one here) are to be compared using ACAP. (Often an experimental data set will be the "base" and one or more NRS code runs will be compared to it in ACAP). The "ACAP" spreadsheet page illustrated in Figure 13c includes, for each of the cases, a sample fully configured ACAP session. Specifically, all information necessary to run ACAP once for each case (i.e., three times here [Demo1, Demo2, Demo3]) is provided including data conditioning, data comparison and FOM assembly elements.

Once these spreadsheet pages are assembled, they are converted to text files and Auto-DA is executed. For those cases that an ACAP configuration has been built, Auto-DA generates the two necessary input files to ACAP. One is a data file which contains x-y values for the data sets to be assessed. The other is an ACAP script file which names the data file and specifies the ACAP configuration (processing parameters such as preprocessing to be performed, metric selection and parameters as required, and metric weighting). The Auto-DA script generates these two files for the specified NRS code solutions and experimental data as well as a file which causes ACAP to be launched in batch mode with the appropriate ACAP script for execution external to the Auto-DA script. ACAP then executes once for each case, writing computed component and overall figures-of-merit to a file. Once the input spreadsheets are assembled for a given suite of test cases, the overall process outlined above becomes quite streamlined. In particular, to revalidate a new version of the code one simply edits the "path" and "cases" pages to point to the new code version and reexecutes Auto-DA and its output ACAP script.
8. ACAP METHODS

The data conditioning and data comparison utilities available in ACAP are summarized in Table 7. The detailed mathematical prescription of these methods is provided in Kunz et. al, 1998b.

The available data conditioning utilities include particular choices of resampling, trend removal and time windowing methods. In ACAP, the user may specify up to six time windows. For each window, a fully configured ACAP session is specified. Individual figures-of-merit are computed for each window and a global figure-of-merit is constructed based on a weighted sum of these contributions.

As discussed above, resampling of the computed data traces is often required for types IV and V data assessments where discretization choices and/or numerical stability issues will generally give rise to NRS predictions of dependent variables at different locations in space-time than where the data was taken. Such consistency is required for all data comparison utilities except methods 1, 8, 15 and 16, though the DFFT and CWT methods can be more accurately deployed if samples are taken at the same time steps. Also, valid application of some trend removal processes, including running averages, require independent variable consistency. ACAP provides a palette of resampling options to perform this task.

As also discussed above, trend removal techniques can be useful in analyzing non-stationary NRS data. A running-average smoother is installed in ACAP for trend definition and the mechanics are available to separately analyze both differences in the trend itself and in the more nearly stationary “low-pass-filtered” traces.

Among the data comparison utilities is the FFT method of D’Auria, 2000. Also available are a number of baseline statistical techniques (methods 2-5, 11-14), Willmott’s, 1982, Index of Agreement (method 7), and several adapted statistical methods utilized by the atmospheric sciences community (methods 6, 8-10, see Ku et al., 1987 for example). Experimental uncertainty is incorporated in a fashion consistent with recent computational fluid dynamic (CFD) code validation work undertaken by Coleman and Stern, 1997, where a “Percent Validated” metric (method 16) is defined from the fraction of simulation data in a trace which falls within the uncertainty bands of the measurements.

In the authors’ view, a particularly attractive comparison tool for NRS code accuracy assessment is the continuous wavelet transform (CWT) measure installed in ACAP (method 15), and some further discussion of this method is provided here. Wavelet transforms are time-frequency techniques which are directly applicable to non-stationary data. As such, if applied consistently, they can provide more accurate representation of local features in a time trace than global transforms (such as the FFT), especially when important features appear at widely varying time scales (as is characteristic of NRS data traces, e.g., see Figure 1e). Also, a variety of CWTs are available, each targeting particular features in a signal (the Morlet wavelet is implemented in the first ACAP release.)

<table>
<thead>
<tr>
<th>Method</th>
<th>Utility Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>D’Auria FFT (DFFT)</td>
<td>Data Comparison Utility</td>
</tr>
<tr>
<td>Mean Error (ME)</td>
<td>Data Comparison Utility</td>
</tr>
<tr>
<td>Variance of Error (VE)</td>
<td>Data Comparison Utility</td>
</tr>
<tr>
<td>Mean Square Error (MSE)</td>
<td>Data Comparison Utility</td>
</tr>
<tr>
<td>Mean Error Magnitude (MEM)</td>
<td>Data Comparison Utility</td>
</tr>
</tbody>
</table>

Table 7. ACAP Methods
Another issue related to the baseline ACAP methods, mentioned above, is the widely varying range and dimensionality of the various data comparison measures. This complicates the definition of an overall figure-of-merit, and thereby motivated normalization and range limit scaling in constructing component figures of merit. Specifically, each individual comparison measure was redefined to range from 0 to 1, corresponding to worst possible and best possible agreement between a given computed trace and experiment. The process implemented to do so comprised two steps. First, all dimensional figures-of-merit are non-dimensionalized with respect to the experimental dependent variable range $|O_{\text{max}} - O_{\text{min}}|$. This “sizes” the different metrics such that $O(10^0)$ errors (i.e. order of 100 % errors) between traces will give rise to $O(10^0)$ metric values. The second step is to, where necessary, modify these “sized” metric definitions so that they independently return figures of merit between 0 and 1. Several of the comparison metrics have ranges between 0 and $\infty$ or $-\infty$ and $\infty$. For all of these except the DFFT and CWT measures, a method for achieving the desired range of $[0,1]$ is implemented, somewhat arbitrarily, as $\text{FOM} = 1/(|\eta|+1)$, where $\eta$ is the non-dimensionalized metric. The DFFT, CWT, MRE and $\rho_{\text{xy}}$ metrics require somewhat different treatment, the form of which is available in Kunz et al., 1998b. Since chosen normalization and range limit scaling of the data comparison utilities in ACAP are somewhat arbitrary, ACAP users may wish to invoke alternate definitions or simply consider the “raw” metrics returned by the baseline methods. This latter option is available in the code, the former would require some modest C++ code modifications.

Significantly more detail on the specific methods employed in ACAP are available in Kunz et al., 1998a-d, 2000a, b. The mathematical specification of the methods is available in Kunz et al., 1998b.

9. ACAP APPLICATIONS

To date, ACAP has been employed for a large number of test cases, as we evolve the capabilities of, and our own experience with, the tool. Several example applications are presented in this section. The purpose of these demonstration cases is to illustrate the functionality of the code and its ability to provide objective accuracy measures.

<table>
<thead>
<tr>
<th>Method Utility Class</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Method</strong></td>
</tr>
<tr>
<td>Mean Relative Error (MRE)</td>
</tr>
<tr>
<td>Index of Agreement (IA)</td>
</tr>
<tr>
<td>Systematic Mean Square Error (SMSE)</td>
</tr>
<tr>
<td>Unsystematic Mean Square Error (UMSE)</td>
</tr>
<tr>
<td>Mean Fractional Error (MFE)</td>
</tr>
<tr>
<td>Correlation Coefficient ($\rho_{\text{xy}}$)</td>
</tr>
<tr>
<td>Standard Linear Regression (L2-standard)</td>
</tr>
<tr>
<td>Origin Constrained Linear Regression (L2-constrained)</td>
</tr>
<tr>
<td>Perfect Agreement Norm (L2-perfect agreement)</td>
</tr>
<tr>
<td>Continuous Wavelet Transform (CWT)</td>
</tr>
<tr>
<td>Percent Validated (PV)</td>
</tr>
<tr>
<td>Resampling</td>
</tr>
<tr>
<td>Trend Removal</td>
</tr>
<tr>
<td>Time-Windowing</td>
</tr>
</tbody>
</table>
D’Auria Sample Experimental and Calculated Time Traces

The functionality of ACAP is demonstrated using the D’Auria “sample” data introduced above. The data was input to ACAP and displayed graphically as reproduced in Figure 12. Four component figures-of-merit were chosen: DFFT, MSE, $\rho_{xy}$ and CWT. These were selected and each given a weight of 0.25 in the Figure of Merit Configuration dialog box, as also shown in the figure. The assessment analysis was then run and the results displayed below the data plot. For the rather arbitrary selections made here, ACAP returns consistently superior component and overall figures-of-merit for sample trace 1. The CWT measure is illustrated in Figure 14, where the locus of points generated by the CWT for each time trace is plotted in the AA-1/WF plane. The percentage of points within the illustrated acceptance boundary defines the figure-of-merit.

Type III Data Assessment

In order to illustrate the use of ACAP for producing figures-of-merit for type III data, use is made of an, as yet, unpublished two-phase pressure drop analysis performed at Penn State. Several different popular empirical correlations were used to predict the two-phase pressure drop for water flowing upwards through a heated tube at 1000 psia. Comparisons were made against experimental data from Matzner et al., 1965. Figure 15a shows a predicted vs. measured scatter plot comparison of the experimental data against Martinelli-Nelson correlation predictions. Figure 15b shows a similar comparison using results from the Freidel empirical correlation.

Visual inspection of the data illustrates that the Freidel model is clearly more accurate over the entire range of pressures analyzed. The issue here is whether this behavior can be captured quantitatively through some figure-of-merit strategy using ACAP. After importing the relevant data into the code, the ACAP session was configured to make use of the metrics that may reasonably be applied to type III data. No data preconditioning was necessary because the data were already synchronized before being imported into the code. Table 8 provides a summary of the individual figures-of-merit returned by ACAP for each metric, the weighting factors used, and an overall assessment value, for each pressure drop correlation.

Table 8. Comparison of ACAP Results for Presented Type III Data

<table>
<thead>
<tr>
<th>Method</th>
<th>M-N Model</th>
<th>Freidel Model</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Error</td>
<td>0.948</td>
<td>0.995</td>
<td>0.077</td>
</tr>
<tr>
<td>Variance of Error</td>
<td>0.996</td>
<td>0.999</td>
<td>0.077</td>
</tr>
<tr>
<td>Mean Square Error</td>
<td>0.994</td>
<td>0.999</td>
<td>0.077</td>
</tr>
<tr>
<td>Mean Error Magnitude</td>
<td>0.947</td>
<td>0.980</td>
<td>0.077</td>
</tr>
<tr>
<td>Mean Relative Error</td>
<td>0.923</td>
<td>0.984</td>
<td>0.077</td>
</tr>
<tr>
<td>Index of Agreement</td>
<td>0.965</td>
<td>0.993</td>
<td>0.077</td>
</tr>
<tr>
<td>Systematic Mean Square Error</td>
<td>0.956</td>
<td>0.999</td>
<td>0.077</td>
</tr>
<tr>
<td>Unsystematic Mean Square Error</td>
<td>0.974</td>
<td>0.999</td>
<td>0.077</td>
</tr>
<tr>
<td>Mean Fractional Error</td>
<td>0.484</td>
<td>0.882</td>
<td>0.077</td>
</tr>
<tr>
<td>Correlation Coefficient</td>
<td>0.990</td>
<td>0.989</td>
<td>0.077</td>
</tr>
<tr>
<td>Standard Linear Regression</td>
<td>0.984</td>
<td>0.997</td>
<td>0.077</td>
</tr>
<tr>
<td>Origin Constrained Linear Regression</td>
<td>0.996</td>
<td>0.997</td>
<td>0.077</td>
</tr>
<tr>
<td>Perfect Agreement Norm</td>
<td>0.992</td>
<td>0.997</td>
<td>0.077</td>
</tr>
</tbody>
</table>
For each method, except one, the figures-of-merit are seen to be closer to unity for the Freidel case, indicating better agreement to the experimental data. The exact sensitivity of a particular metric to changes in the pressure drop correlation is seen to vary significantly. In some cases, the figures-of-merit only differ in the third decimal place while for others, the differences occur in the first or second decimal place. While it is not the purpose of this paper to present a detailed discussion of this behavior, these differences in sensitivity derive, in part, from the ability, or lack thereof, of a particular metric to capture a particular trait of the data set. For example, the closely corresponding values of the $\rho_{xy}$ and $L_2$-constrained metrics suggest that both models correlate quite well with data. Taken with the significant differences in MFE, which is a good measure of bias in the predictions, one can conclude that the shortcomings of the Martinelli-Nelson model are principally due to a consistent over-prediction of the pressure drop. These observations, and other which can be drawn from the results in Table 8, illustrate the utility of selecting multiple figures-of-merit to capture different features in NRS code comparisons. Here, equal weighting was arbitrarily given to each method in constructing the overall merit value. In general, when constructing figure-of-merit configurations, the user would need to analyze the data, identify the traits which need to be captured, and make appropriate decisions as to which metrics ought to be used and how they should be weighted.

**Type V Data Assessment**

The next example illustrates the use of ACAP with type V data. Figure 16 shows a comparison between the predicted and measured rod surface temperature at a particular axial level during the core heatup and reflood stages of a FLECHT SEASET vs. TRAC-B simulation (Paige, 1998). Two different reflood heat transfer models within TRAC-B were employed - the original and a newer model.

Again, after importing the relevant data into the code, methods were selected to construct a figure-of-merit. In particular, most of the metrics used in the previous example were retained and the DFFT and CWT methods were also implemented. Because the data were not synchronized before being imported, ACAP’s resampling feature was used to linearly interpolate between each respective model’s data points, and subsequently generate new “predicted” data points which correspond in time to those in the experimental data set. Furthermore, because the original reflood model results only ran out to about 330 seconds, the data used to construct the figures-of-merit was limited to $t < 330$ using ACAP’s time-windowing feature. After these configuration steps were performed, ACAP generated the figures-of-merit for each simulation and these are summarized in Table 9.

For each case, the merit values do generally get better for the modified reflood model simulation, as expected. The overall figure of merit for the new reflood model is 0.880 while for the original code case, it is only 0.553.

<table>
<thead>
<tr>
<th>Method</th>
<th>Original Model</th>
<th>New Model</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>D’Auria FFT</td>
<td>0.035</td>
<td>0.141</td>
<td>0.077</td>
</tr>
</tbody>
</table>

Table 9. Comparison of ACAP Results for Presented Type V Data
As a final demonstration example, a type V data assessment is performed for a case which has an experimental uncertainty available. Integrated mass flow through an automatic depressurization system for an OSU SBLOCA case is considered (NRC12). An experimental uncertainty is known for this quantity. Figure 17a shows the ACAP interface display of the experimental data with a RELAP5 solution and an artificial systems code solution (here simply a straight line). The percent validated metric, defined above, is utilized to assess the relative accuracy of the two simulations. Though the artificial code solution exhibits significant differences from both RELAP5 and measured values, the RELAP5 and artificial simulations return PV values of 0.42 and 0.40 respectively. Therefore the artificial data cannot be deemed much less accurate than the RELAP5 simulation if taken in light of the experimental uncertainty. This is further illustrated in Figure 17b where the absolute error of the two simulations is plotted with the experimental uncertainty bands.

10. CONCLUSION AND SUMMARY RECOMMENDATIONS

A number of mathematical data analysis methods have been surveyed for their applicability in the construction of NRS code-data and code-code comparison measures. The goal of the survey was to identify issues and techniques to be considered in the development of an automated code assessment procedure, ACAP, to be brought to bear in NRC advanced T/H code consolidation efforts. Techniques from the overlapping fields of approximation theory, time-series data analysis, basic statistical analysis, as well as several other methods have been considered. Several techniques were demonstrated using example NRS code-data sets.

A number of conclusions apply:

1) Most of the methods considered can be applied to provide useful quantitative measures of accuracy for at least a subset of NRS data types III, IV and V.
2) Inappropriate use of some methods can yield incorrect results, that is, return figures-of-merit that are worse for more accurate simulations. This motivates:

   - Definition of a robust comparison measure or suite of measures as one that reliably returns better figures-of-merit for

---

### Table 9. Comparison of ACAP Results for Presented Type V Data

<table>
<thead>
<tr>
<th>Method</th>
<th>Original Model</th>
<th>New Model</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Error</td>
<td>0.555</td>
<td>0.969</td>
<td>0.077</td>
</tr>
<tr>
<td>Variance of Error</td>
<td>0.779</td>
<td>0.996</td>
<td>0.077</td>
</tr>
<tr>
<td>Mean Square Error</td>
<td>0.519</td>
<td>0.995</td>
<td>0.077</td>
</tr>
<tr>
<td>Mean Error Magnitude</td>
<td>0.555</td>
<td>0.967</td>
<td>0.077</td>
</tr>
<tr>
<td>Mean Relative Error</td>
<td>0.689</td>
<td>0.985</td>
<td>0.077</td>
</tr>
<tr>
<td>Index of Agreement</td>
<td>0.421</td>
<td>0.992</td>
<td>0.077</td>
</tr>
<tr>
<td>Mean Fractional Error</td>
<td>0.052</td>
<td>0.556</td>
<td>0.077</td>
</tr>
<tr>
<td>Correlation Coefficient</td>
<td>0.037</td>
<td>0.988</td>
<td>0.077</td>
</tr>
<tr>
<td>Standard Linear Regression (L2 Norm)</td>
<td>0.926</td>
<td>0.998</td>
<td>0.077</td>
</tr>
<tr>
<td>Origin Constrained Linear Regression (L2 Norm)</td>
<td>0.981</td>
<td>0.999</td>
<td>0.077</td>
</tr>
<tr>
<td>Perfect Agreement Norm (L2 Norm)</td>
<td>0.979</td>
<td>0.998</td>
<td>0.077</td>
</tr>
<tr>
<td>Continuous Wavelet Transform</td>
<td>0.665</td>
<td>0.864</td>
<td>0.077</td>
</tr>
<tr>
<td>Combined Figure-of-Merit</td>
<td>0.553</td>
<td>0.880</td>
<td></td>
</tr>
</tbody>
</table>
superior comparisons and worse figures-of-merit for inferior comparisons

- That great care be taken in the selection of the suite of analysis tools chosen for each particular comparison

3) The inherent limitations to stationary data of most available methods render straightforward application to NRS type V data less than rigorous. Trend removal techniques can be brought to bear to preprocess the data, thereby yielding more robust comparison measures, especially when deployed in concert with time-windowing.

4) Experimental uncertainty can be effectively incorporated in code-data accuracy assessment within the framework of the “toolkit” of analysis procedures considered. Experimental uncertainty should be included with the “raw” experimental data in the code reassessment test matrix.

5) Inconsistency between the computed and measured independent variable range and basis (i.e. different time steps) motivated the incorporation of resampling and range trimming conditioners within ACAP. Such “synchronization” is required for most comparisons.

6) For type V data, techniques that are intrinsically appropriate for non-stationary data analysis can be utilized in the construction of comparison measures. These include best approximation fits and, most promising in the view of the present investigators, time-frequency techniques.

7) There is a fundamental lack of rigor in applying basic statistical analysis procedures to most NR systems data. This arises due to non-stationarity of the data and the unavailability of a known distribution of error about its mean. This renders the construction of statistical inference measures suspect at best. Basic statistical difference and correlation measures can be deployed to construct useful figures-of-merit, but uncertainty bounds should not be inappropriately constructed.

8) As indicated in conclusion 2 above, great care must be taken in employing comparison measures. In particular, for each experimental data set, a demonstrably robust assessment strategy must be developed. The present investigators feel that this requirement motivates a process whereby expert assessors “calibrate” and document a suite of robust data analyses for each experimental data set in the code reassessment matrix. This assessment configuration will in general include preconditioning strategies, data comparison measures, figure-of-merit weighting assembly factors, and should be included with the “raw” experimental data in the reassessment matrix. Such configured assessments will then be used to define ACAP sessions in future code re-assessments.

In concert with the method assessment findings presented, a set of baseline techniques for code-data (or code-code) comparisons, data preconditioning, figure-of-merit-assembly and incorporation of experimental uncertainty were selected and implemented in ACAP. An overview of the code mechanics with example applications was provided. The authors believe that the ACAP tool can play an important role in code quality assessment in the NRC’s consolidated code development and other NRS code development and application environments. Details of these and a more general overview of the software are available in Kunz et. al., 1998a-d, 2000a, b.

11. ACKNOWLEDGEMENT

This work was performed under United States Nuclear Regulatory Commission Contract NRC-04-97-046, Task Order #3, with contract monitor Dr. Jennifer Uhle.

12. REFERENCES

CSNI Specialist Meeting on Transient Two-Phase Flow, Aix-en-Provence, France.
Appendix K of 10CFR50, 1/1/97 Edition.


Lee and Rhee 1997 Data provided by S. Smith of NRC.


Figure 1a) Sample NRS data type I. Key parameters tables (from Bessette and Odar [1986]).

Figure 1b) Sample NRS data type II. Timing of events table (from Jo and Connell [1985]).

Figure 1c) Sample NRS data type III. Scatter plot of nominally 0-D data (from Shumway [1995]).

Figure 1d) Sample NRS data type IV. 1-D (in space) steady state data (from Shumway [1995]).

Figure 1e) Sample NRS data type V. Time record data (from Lee and Rhee [1997]).
Figure 2a) Comparison of measured and RELAP5 predicted vessel pressure vs. time for the NRC12 case (from Lee and Rhee [1997]).

Figure 2b) Comparison of several best approximation fits to the absolute error associated with data in Figure 2a.

Figure 3a) Polynomial fit to measured (closed circles) and two RELAP5 predicted (open symbols) heat flux distributions for MIT-Siddique test data (digitized from Shumway [1995]).

Figure 3b) Cubic spline fit to measured (closed circles) and two RELAP5 predicted (open symbols) heat flux distributions for MIT-Siddique test data (digitized from Shumway [1995]).
Figure 4a) Segment of measured and RELAP5 predicted vessel pressure vs. time for the NRC12 case (from Lee and Rhee [1997]). b) Autocorrelation of experimental and computed time series, and approximate MA model of computed time trace.

Figure 5a) D’Auria artificial code assessment data, digitized from Ambrosini et al. [1990].

Figure 5b) D’Auria figure-of-merit computed from data appearing in Figure 5a.
Figure 6a) UCB wall condensation test data from Shumway [1995]. Computed vs. measured wall heat flux. Default RELAP5 diffusion model. Lines correspond to L2-standard (solid), L2-constrained (dotted) and Perfect agreement (dashed).

Figure 6b) UCB wall condensation test data from Shumway [1995]. Computed vs. measured wall heat flux. New RELAP5 diffusion model. Lines correspond to L2-standard (solid), L2-constrained (dotted) and Perfect agreement (dashed).

Figure 6c) UCB wall condensation test data from Shumway [1995]. Absolute error vs. measured wall heat flux. Default RELAP5 diffusion model.

Figure 6d) UCB wall condensation test data from Shumway [1995]. Absolute error vs. measured wall heat flux. New RELAP5 diffusion model.

Figure 6e) UCB wall condensation test data from Shumway [1995]. PDF of absolute error. Default RELAP5 diffusion model.

Figure 6f) UCB wall condensation test data from Shumway [1995]. PDF of absolute error. New RELAP5 diffusion model.
Figure 7a) Segment of measured and TRAC-B predicted rod temperature vs. time for FLECHT SBLOCA test (Paige [1998]). b) Absolute error associated with data in a). c) Running average trends and the absolute error associated with the data in a). d) Residuals and the absolute error associated with data in a), c). e) Time windows defined for data in a). f) Running average trends and the absolute error associated with the data in a) with separate running averages applied to transition and reflood time windows. g) Residuals and the absolute error associated with data in a), f).
Figure 8a) Measured vessel pressure vs. time for the NRC12 case (from Lee and Rhee [1997]). b) Short time Fourier transform spectrogram of data in Figure 8a. c) Morlet continuous wavelet transform spectrogram of data in Figure 8a.
Figure 8d) Continuous wavelet representation of solution accuracy applied to D'Auria's artificial data appearing in Figure 5a.
Figure 9a) RELAP5 predicted and measured integrated mass flow through an ADS vs. time for NRC12 case (from Lee and Rhee [1997]). Linear “artificial” data and experimental uncertainty band also plotted. b) absolute error associated with data in Figure 9a.
Figure 10a) Artificial data, simulations and experimental uncertainty band.
Figure 10b) Trends for artificial data traces in Figure 10a. c) Residuals for artificial data traces in Figure 10a.
Figure 10d) Discrete Fourier transform of artificial data in Figure 10a and the absolute error of the amplitudes.
Figure 11) Schematic overview of the structure of ACAP and the Auto-DA tool.
Figure 12) Elements of interactive mode ACAP interface. a) "D'Auria" data [5] displayed in ACAP main window with results of comparison assessment for sample “code” results. b) Resampling dialog. c) Figure-of-merit configuration dialog.
Figure 13) Elements of batch mode Auto-DA/ACAP spreadsheet interface. a) Auto-DA "path" spreadsheet page, b) "cases" page, c) "ACAP" page
Figure 14) Display of continuous wavelet transform applied to D'Auria data, illustrating locus of points in AA-1/WF plane and acceptance boundary.
Figure 15) Sample Type III data comparisons. Predicted vs. measured scatter plot comparison of pressure drop. Experimental data (Matzner et al. [1965]) against a) Martinelli-Nelson correlation predictions and b) Freidel empirical correlation.
Figure 16) Sample type V data comparisons. Predicted vs. measured rod surface temperatures during heatup and reflood of a FLECHT SEASET transient.

Figure 17) Sample type V data comparisons. Predicted vs. measured integrated mass flow through an ADS vs. time for NRC12 case. a) ACAP display of data and assessment output. b) Plot of absolute error for two simulations with experimental uncertainty.