VALIDATION OF GEOTECHNICAL SOFTWARE FOR REPOSITORY PERFORMANCE ASSESSMENT

T. LeGore, J.D. Hoover, R. Khaleel, E.C. Thornton
R.P. Anantatmula, D.C. Lanigan
Westinghouse Hanford Company, P.O. Box 1970,
Richland WA., 99352

ABSTRACT

An important step in the characterization of a high level nuclear waste repository is to demonstrate that geotechnical software, used in performance assessment, correctly models the physical processes existing in a repository environment. This step is generally referred to as model validation. There is another type of validation, called software validation. It is based on meeting the requirements of specifications documents (e.g. IEEE specifications) and does not directly address the correctness of the specifications.

The process of comparing physical experimental results with the predicted results should incorporate an objective measure of the level of confidence regarding correctness. This measure can take several forms, such as explicit confidence levels or tolerance limits on the accuracy of the prediction. These measures generally require a replication or multiple execution of the experiment to obtain statistical information. However, replication is not always possible for physical data from sources such as natural analogues, field investigations, or large scale laboratory experiments. Proper account of the effects of measurement errors, fabrication errors, limited knowledge of the experiment and the propagation of these errors must also be considered in the comparison. This error analysis is confounded by the use of software based on numerical methods that render calculus based methods of error propagation unusable.

A sound methodology has been developed that allows the experimental uncertainties to be explicitly included in the comparison process. The methodology also allows objective confidence levels to be associated with the software. In the event of a poor comparison, the method also lays the foundation for improving the software.

INTRODUCTION:

In a cooperative effort, Ishikawajima-Harima Heavy Industries, Co, Ltd. (IHI), and the Nuclear Waste Department of Westinghouse Electric Corporation (WEC) are studying experiment concepts for obtaining data for the validation of geotechnical software. A general approach for the data comparison process has been developed for use with large scale tests that allows an objective determination to be made of the confidence levels associated with the software.

Validation of software must be dealt within the context of a total software quality assurance (SQA) program. An SQA program guides and controls the software from initial conception through to the final testing, operation, and eventually retirement of the software. Validation represents the most significant aspect of an SQA program. There are two generally accepted meanings to validation:

"Assurance that the completed software conforms to the functional requirements specified for it at the beginning of the software development process"

"Assurance that a model, as embodied in a computer code, is a correct representation of a process or system for which it is intended"

The first definition is the basis for much of the development of SQA standards such as the ANSI/IEEE series, where they are intended for a much broader application than just geotechnical software. Validation against a requirements document is verifiable and traceable, a primary requirement for a QA program. Additionally, software that has nothing to do with modeling reality can be validated according to these standards. For software that does involve modeling of physical systems, these standards do fall short.

The second definition applies to modeling software used in a repository performance assessment. In this context, validation is a major part of the quality assurance process of defining the accuracy and confidence limits of the software.

There are two important types of information that are to be gained from a validation effort 1) the establishment of the limits of the software to make a prediction and 2) the achievement of as narrow as possible limits on the accuracy of the software. It is equally important to quantify the accuracy of the software, regardless of how poor or good it is, as it is to achieve a particular level of accuracy. Without the determination of the accuracy, a credible performance assessment cannot be made.

DATA SOURCES

A primary consideration in performing a validation of software is the source of data against which the performance of the software is compared. There are three primary sources of data for use in validation:

- Literature search
- Field experiment
- Laboratory experiment

Literature search is the simplest to perform. It is also the most limited of the three. Software used for a performance assessment of a candidate repository site will be "state of the art". Data may not be available from the literature if the software incorporates new processes or extended capabilities. Where the data exist, the quality of the experiment may be unsatisfactory. The accuracy of the physical

description of the experiment may be unknown which in turn will affect the estimated accuracy of the results. Statistically sound replication of the data may be absent as well. The conditions of the system for which the data is being reported may not correspond exactly to the system being modelled by the software. All of these factors contribute to a low credibility and questionable usefulness of literature data for final validation of software. Literature data may still be useful in the initial stages of validation and as a further guide to the design of appropriate experiments.

Field experiment allows the closest approach to testing in the actual repository environment, especially if the experiments are done at a candidate repository site. Although the conditions of the testing in the field can represent those in the repository closer than other test conditions, there are also significant drawbacks to field experiments:

Extensive local scale testing of the site is required to adequately characterize the physical system before a prediction of experimental results can be made.

Due to the high cost of such testing the site must be a strong candidate for the final repository site. (This allows the cost to be shared with the site characterization program.) This places the validation of the software late in the repository program, after the software has been used to make major programmatic decisions.

Any candidate site will be difficult to test for the very reason that the lack of a response is a desirable feature. The testing program will be pushed to its limits to achieve measurable responses in the areas of flow and mass transport processes.

The statistical replication of the data will be poor, since the number of testable sites will be low and the cost of each test will be high.

In the event of a poor or unacceptable comparison between the software and the experiment, the parameters of the field site cannot be varied to aid in determining the source of the errors.

For chemical reaction experiments, the necessary data on reaction paths and kinetics may be unavailable.

Demonstration of the software's ability to predict, even with low accuracy, the results of a field experiment at a candidate repository site will contribute highly to the confidence in the utility of the software. Thus the field experiment is most valuable as the final step in the validation process. However, the field experiment is limited in its general ability to provide insight and support for the validation of software.

The laboratory experiment is the most versatile of the three data sources. The generation of data for the comparison process is performed in a controlled environment under known conditions. The processes to be tested, the ranges of parameters to be covered, and the choice of material properties are controlled by the experiment designer. This allows several of the disadvantages of the field experiment to be overcome. Material properties can be characterized before inclusion into the experiment, an adequate number of sampling points can be included, and entire experiments can be replicated. In the event of a poor comparison, the experiment can be modified and individual processes or parameter ranges examined. These advantages place the laboratory experiment in the central position of providing the bulk of the data necessary for validation work.

ERROR ANALYSIS

The accuracy of software is tied directly to the problems of error detection, propagation, and estimation. The two primary means of estimating errors in experimental data are calculus based and statistically based methods. Each of these methods are described in the following discussion together with the problems presented by complex computer codes.

The calculus based methods rely on the existence of a constitutive equation that has a closed form solution ($Y = f(X_1,X_2,...,X_n)$) describing the experimental results. The partial derivatives of the dependent variable with respect to each of the independent variables is formed and the estimate of the error in the dependent variable is approximated as:

$$VAR[Y] = \sum_{i=1}^{N} VAR[X_i] \left[\frac{\partial Y}{\partial X_i}\right]^2 + 2 \sum_{i=1}^{N-1} \sum_{J=i+1}^{N} COV[X_i, X_J] \frac{\partial Y}{\partial X_i} \frac{\partial Y}{\partial X_J} \quad \text{Eq. (1)}$$

where VAR and COV are the variance and covariance of the variables respectively.

The most obvious problem is that the closed form solution must exist before the method can be applied. This is a highly limiting condition for practical problems as evidenced by the proliferation of numerically based (e.g. finite element) codes to perform calculations that are not otherwise possible. Where a closed form solution does exist, it often does not represent a physically realizable experiment. This limits the closed form solution to the role of verification without contributing to the validation of the software. A second problem arises when the statistical data on the input variables is not known and must be estimated based upon professional judgement. This reduces the objectivity of the estimate and hence its credibility.

Statistical methods rely upon the replication of the experimental results to develop explicitly the distribution of errors in the dependent variable. Comparison with the predicted results is then usually based upon the predicted results falling with a pre-specified tolerance band around the mean experimental result. Replication of experiments is a sound method but can be prohibitive when the costs of experimentation are high or, as can be the case in field testing, the sites available to be tested are limited. Additionally, this method does not allow the assignment of error to the various sources. There is little that can be done with the

data to aid in improving the software as the sources of the error are not identified.

A second statistical method that is used when replication is limited is to calculate a correlation coefficient that is used to indicate how closely the predicted results match the limited experimental values. This suffers from the drawback that there is usually no significance or confidence level associated with a particular value of the correlation coefficient. This follows from the fact that the distribution of the correlation coefficient is not known so that even if a significance level is specified, the corresponding value of the correlation coefficient is not specified.

SOURCES OF ERROR

There are four sources of error that need to be considered in performing an error analysis of the accuracy of software:

- Numerical computation errors;
- Measurement and instrumentation errors;
- Fabrication errors;
- Theory errors.

The numerical computation errors are errors in the computation that result from such things as round off, truncation, convergence criteria, and approximations in the numerical methods. Quantifying these errors is the task of a verification and benchmarking program and are not dealt with here.

The remaining three source of errors are logically incorporated in the validation methodology. The measurement and fabrication errors are accounted for directly while the errors of theory are, by definition, those errors not accounted for by the other possible error sources.

VALIDATION METHODOLOGY

A nonparametric validation methodology that has been designed to resolve these problems and allow a sound and defensible comparison between experiments and complex software is shown in Fig. 1. There are three key steps to this method: 1) Characterization of the measurement errors, 2) Characterization of the fabrication errors and 3) Propagation of these errors into the output of the software.

The process of determining the distribution of measurement errors is standard in many industries. Instrument calibration is a standard practice that is performed routinely. When a measurement involves the expertise of the experimenter then further care is required. Statistical process quality control measures are required in order to characterize the effect the experimenter has on the resulting measurement. These procedures are also well established in both literature and industry and so present little problem in implementation.

The characterization of fabrication errors for laboratory experiments also relies upon the field of statistical process quality control. The material properties that are put into an experiment (e.g. the hydraulic conductivity of a porous medium) are subject to variation. The same can be said for the boundary and initial conditions of the experiment (e.g. the initial masses of reactants in a geochemistry experiment). Also any constants used in the analysis, such as reaction rate constants are subject to uncertainty resulting from their own experimental determinations.

The characterizations of 'fabrication' errors for field experiments is not explicitly an error of fabrication, but results from errors of knowledge about the state of the system being tested. Estimation of these errors is possible through the use of geostatistical estimation techniques. Geostatistics can provide a linear unbiased estimate of the physical properties and initial conditions between the site sampling points. Even more important, geostatistics can provide uncertainty estimates of the variation of those data about the estimated values. Once the errors of knowledge are characterized, they can be treated as equivalent to the errors of fabrication for a laboratory experiment.

The propagation of these errors into the output of the software is the third key step in the process. A reference analysis of the experiment is performed using the mean values for all of the software input data. The output of this mean value analysis forms the basis for developing the distributions of the test statistics used later.

The fabrication errors are the starting point in a Monte Carlo process that will develop the distributions of the test statistics. A realization of the physical description of the experiment is created by randomly selecting values for the software input parameters from the predetermined distributions. The realization used as input to the software leads to the software output as a realization of the experimentally measurable variables (temperature, pressure concentration, etc.). A realization of the measurement errors is similarly created and algebraically added to the experimental realization to create a realization of the measured data. The distance, or fit, between the reference analysis (based on best estimate data) and the realization of measured data is then calculated using a test statistic of the analyst's choice. In practice, more than one statistic can and should be used. Replication of this Monte Carlo sequence allows the analyst to build up an empirical discrete distribution for each of the test statistics. These distributions are then used later in the comparison of the experimental data to the reference analysis.

The primary advantage of creating the empirical distributions is that no assumptions need to be made about the form of the distribution. The form is implicit in the discrete data. The analyst is free to choose a distance measure that reflects a particular feature of the system or theory. The distance measure can then be directly tested and assigned a significance level or accepted/rejected based upon a

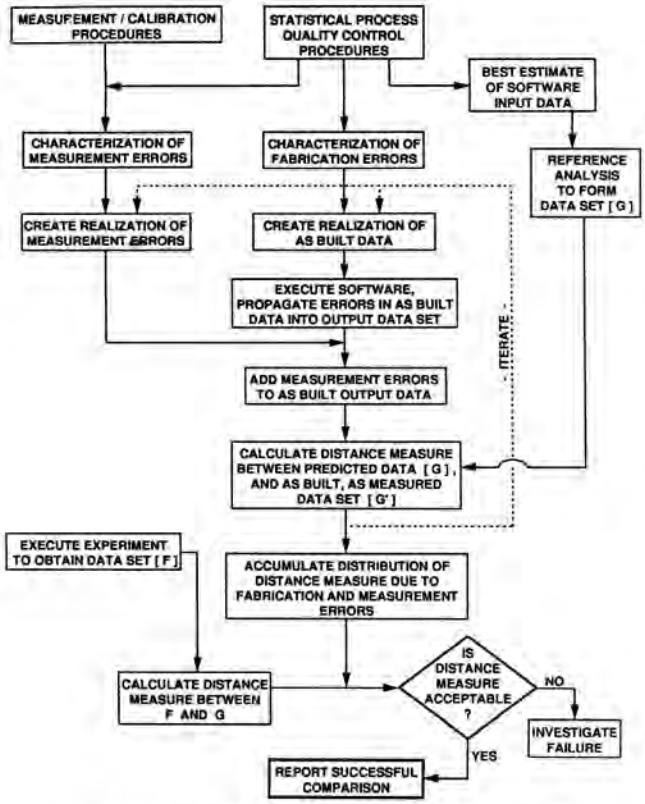

Fig. 1. Nonparametric Validation Methodology.

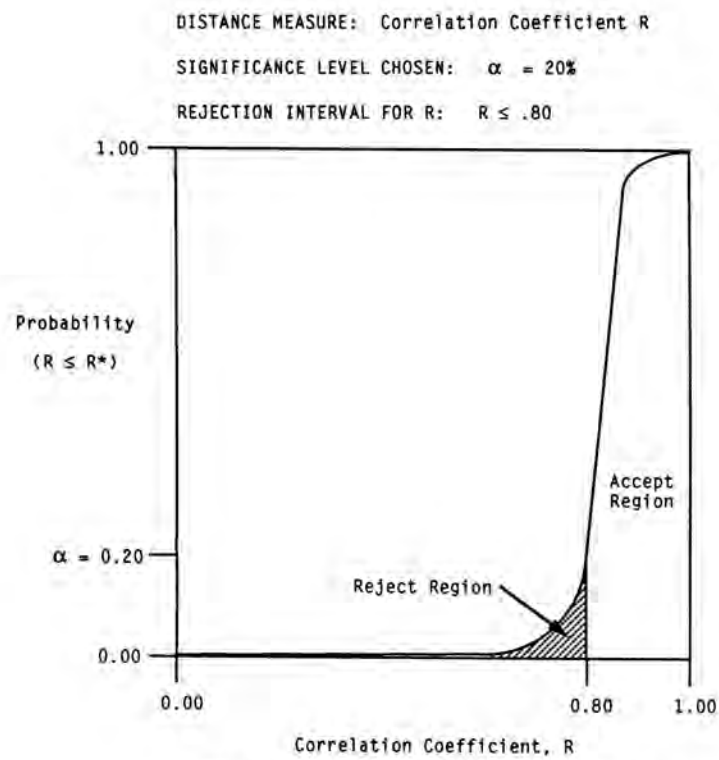

Fig. 2. Example Empirical Distribution.

predetermined significance level (Fig. 2). Several example measures from the field of mass transport are possible:

- The differences in arrival time of the median value of a tracer (effective porosity);
- The differences in the spread of the breakthrough curve (dispersion);
- The sum of the squares of the residuals between two breakthrough curves (functional form).

Each of these distance measures has a usefulness in testing the software. None of them can, a priori, be claimed to fit any of the standard distributions used in statistical testing.

Once the empirical distribution of the distance measure is developed, the results of the experiment can be evaluated. The experimental data is compared to the reference analysis and the same distance measure is formed. The value of the distance measure is then compared against the empirical distribution to determine the associated significance level of the measure. An accept/reject decision can now be made by testing the significance level against that determined by expert judgement to be acceptable.

SIGNIFICANCE LEVELS

The significance level of a statistical test is the primary measure of accuracy for that test. The determination of appropriate significance levels is determined by expert judgement in the validation methodology. The significance level, alpha, identifies the probability that a type I error will occur. A type I error occurs when the hypothesis (the software is correct) will be rejected when it is in fact true. The larger the chosen alpha the more stringent the comparison.

The chosen alpha also affects the number of Monte Carlo trials needed to develop the distributions of the test statistics. For example, 100 to 500 trials are commonly considered necessary to adequately characterize an unknown distribution. The expected value of a test statistic used in this process will be near the median (50 percentile) value of the discrete distribution. The choice of an alpha close to the expected value of the test statistic implies a higher desired precision in establishing the test statistics value at the alpha point in the curve. This would lead to a larger number of trials than a lower alpha where less precision can be accepted.

PROGRAMMATIC FACTORS

The methodology described here has some definite constraints that must be considered in planning a validation effort.

- A major commitment to quality assurance and quality control is required early in the program. This allows the QA and QC aspects to be incorporated into the program planning.
- Senior personnel who are well versed in statistical process control and statistical analysis should be included in the staff
- Software not originally designed for Monte Carlo application will require pre- and post-processors to be developed. This is not only to allow them to be operated as Monte Carlo codes, but to perform the comparisons between the output data sets. This represents a major software coding effort that should not be overlooked.
- Monte Carlo simulations are typically computer resource intensive. The availability of substantial computing power (i.e. class VI mainframe) will be required.

Experiment design should avoid single valued results in favour of multiple data points from a single experiment. This increases the reliability of the data from the standpoint of internal consistency and statistical soundness.

It is emphasized that validation of software is an ongoing process. An early commitment to the validation of the software is required.

SUMMARY

The primary value of software validation lies first in the determination of the accuracy of the software, and second in providing the data necessary for the improvement of the accuracy of the software. The determination of software accuracy requires comparison against field or experimental data plus a credible estimate of the errors associated with the comparison. Traditional methods of error analysis (e.g. calculus and replication) are not productive for complex numerically based geotechnical software. The physical experiments that provide data for comparison are often unique or limited in their ability to be replicated, leaving physical data with a poor statistical basis for error estimates. Software that relies on numerical techniques such as finite element methods cannot be analyzed via calculus based methods to determine the affect of error propagation on the output data.

The described methodology explicitly incorporates the necessary error propagation data through Monte Carlo execution of the software. The sources for the variability in the input data is determined through statistical process quality control and measurement calibration procedures. The resulting output of the Monte Carlo analysis is used to develop a discrete distribution of distance measures that can be tested and assigned significance levels.

In this methodology, software can be initially tested with literature data. As the test program continues, the progression to lab and field data provide for increased levels of confidence in the program. Each step in the progression represents an increase in assurance that the software is a correct representation of the system for which it is intended.