LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-TR-207693

Solution Verification Linked to Model Validation, Reliability, and Confidence

Roger W. Logan, Cynthia K. Nitta

November 4, 2004

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

6/15/2004 1:01
NECDC 4-8 October 2004 at LLNL

Solution Verification Linked to Model Validation, Reliability, and Confidence (U)

Roger W. Logan
Cynthia K. Nitta
University of California
Lawrence Livermore National Laboratory

Oral Presentation Preference
Applications / V&V Track Preference

email: RWLOGAN@LLNL.GOV, NITTA1@LLNL.GOV

Title and Executive Summary UNCLASSIFIED per R.W. Logan, ADC, 15 Jun 04

Full paper and presentation may not be unclassified but can be made transmissible

---

Executive Summary:

The concepts of Verification and Validation (V&V) can be oversimplified in a succinct manner by saying that "verification is doing things right" and "validation is doing the right thing". In the world of the Finite Element Method (FEM) and computational analysis, it is sometimes said that "verification means solving the equations right" and "validation means solving the right equations". In other words, if one intends to give an answer to the equation "2+2=", then one must run the resulting code to assure that the answer "4" results. However, if the nature of the physics or engineering problem being addressed with this code is multiplicative rather than additive, then even though *Verification* may succeed (2+2=4 etc), *Validation* may fail because the equations coded are not those needed to address the real world (multiplicative) problem. We have previously provided a 4-step "ABCD" quantitative implementation for a quantitative V&V process:

A. Plan the analyses and validation testing that may be needed along the way. Assure that the code[s] chosen have sufficient documentation of software quality and Code Verification (i.e., does 2+2=4?). Perform some calibration analyses and calibration based sensitivity studies (these are not validated sensitivities but are useful for planning purposes). Outline the data and validation analyses that will be needed to turn the calibrated model (and calibrated sensitivities) into validated quantities.
B. Solution Verification: For the system or component being modeled, quantify the uncertainty and error estimates due to spatial, temporal, and iterative discretization during solution.
C. Validation over the data domain: Perform a quantitative validation to provide confidence-bounded uncertainties on the quantity of interest over the domain of available data.
D. Predictive Adequacy: Extend the model validation process of "C" out to the application domain of interest, which may be outside the domain of available data in one or more planes of multi-dimensional space. Part "D" should provide the numerical information about the model and its predictive capability such that given a requirement, an adequacy assessment can be made to determine of more validation analyses or data are needed.

Step “B” in the 4-step “ABCD” validation process is that of Solution Verification. Solution Verification is the process of assuring that a model approximating a physical reality with a discretized continuum (e.g. finite element) code converges in each discretized domain to a converged answer on the quantity of subsequent validation interest. This is accomplished in the spatial domain by subdividing the elements or cells on the entire grid or portions of the grid, as shown in Figure 1.

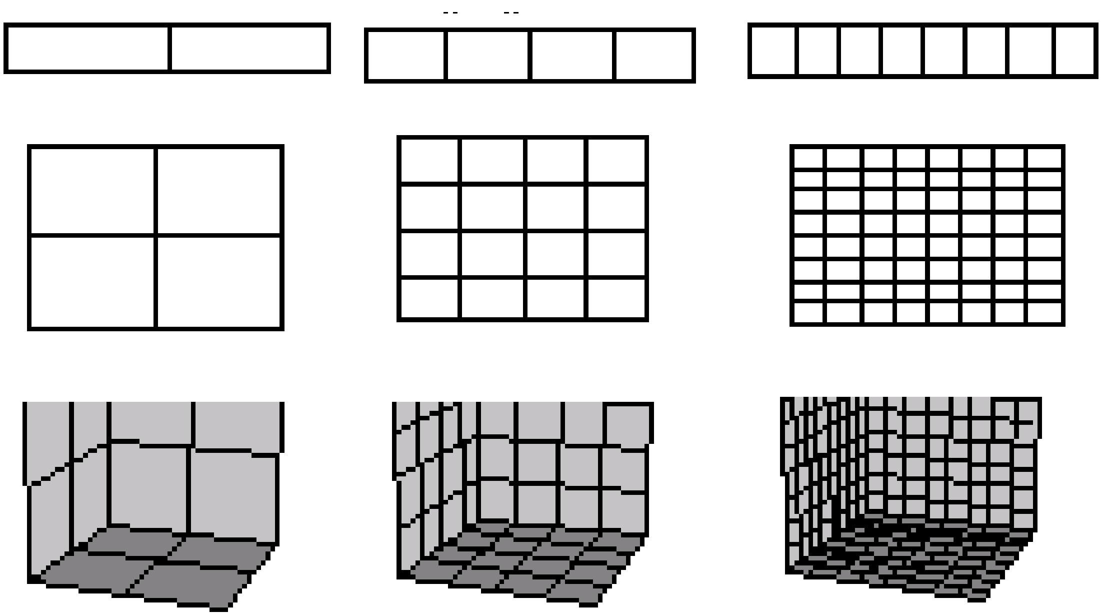

Figure 1. “B” of “ABCD”: Solution Verification quantifies the proximity to convergence, and the model uncertainties we may be forced to accept.

Stated to the extreme, we do not care, in solution verification, if the converged answer is right or wrong; the issue of obtaining the correct (i.e. 2+2=4) answer is the realm of code verification. If we modeled finer and finer meshes and converged to 2+2=5, with an order of convergence consistent with our numerical technique, we should be happy. The fact that we obtained 2+2=5 would be dealt with as a code verification error, not a failure of solution verification. So the process for solution verification would be to run our discretized continuum model at finer and finer meshes, and obtain both a converged solution and a smooth order of convergence. Such a “smooth” convergence plot is shown in Figure 2.

When the mesh convergence results such as the combustion example shown in Figure 2, we can proceed to obtain estimates of uncertainty due to Solution Verification as detailed in the full paper and in many prior references. The result is typically expressed as error estimate on a log-log plot, with a straight line showing the order (power) of convergence as in Figure 3:

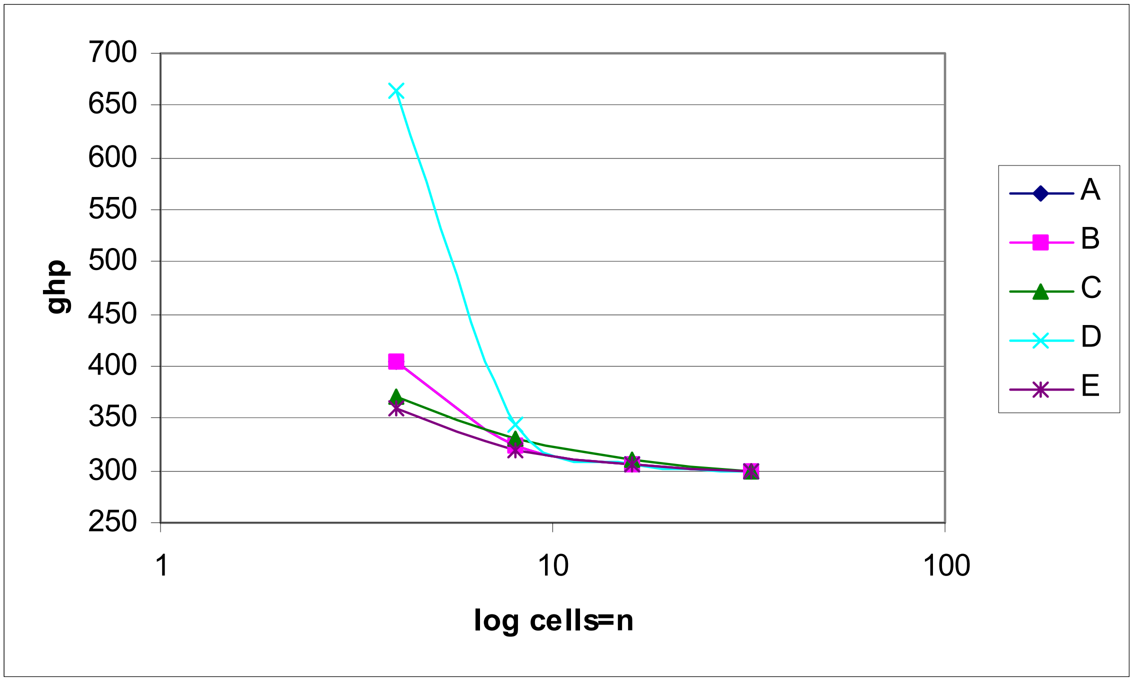

Figure 2. Mesh convergence study for solution verification: Combustion example. Smooth, monotonic results shown for this case.

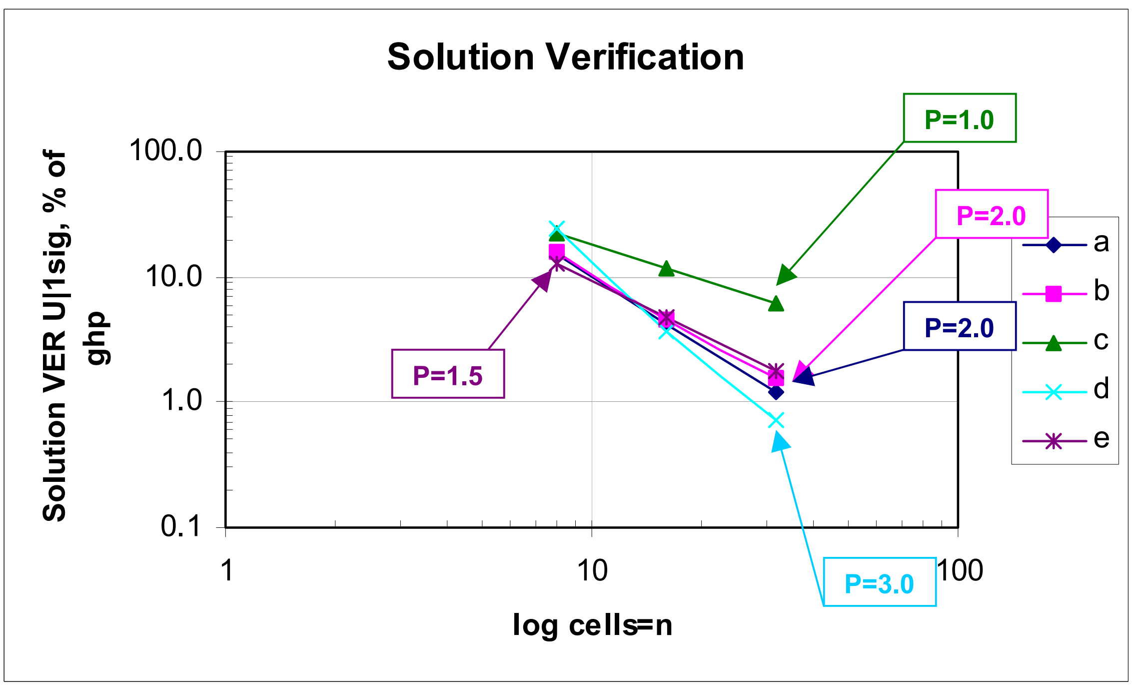

Figure 3. Order of convergence for the smooth example of Figure 2. Straight log-log lines and clear order of convergence for this monotonic example.

The modeling reality is that often we are modeling a problem with a discretized code because it is neither smooth nor continuous spatially (e.g. contact and impact) or in relevant physics (e.g. shocks, melting, etc). The typical result is a non-monotonic convergence plot that can lead to spurious conclusions about the order of convergence, and a lack of means to estimate residual error or uncertainty. We offer one emerging technique that enables a quantification of solution verification uncertainty at confidence and order of convergence for monotonic and non-monotonic mesh convergence studies. The method offers insight into both code development (convergence order versus that expected), and supplies the quantitative terms needed for inclusion into subsequent model validation, confidence, and reliability analyses. We show that this method can preclude the calculation of spurious high values of convergence order, and give reasonable uncertainty estimates for inclusion into the quantitative validation process, parts “C” and “D” of “ABCD” validation. We demonstrate this on a real system example.

(U)