8th. World Congress on Computational Mechanics (WCCM8)
5th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2008)
June 30 –July 5, 2008
Venice, Italy

Impact of Coding Mistakes on Numerical Error and Uncertainty in Solutions to PDEs

Patrick M. Knupp[1],* Curtis C. Ober[2] and Ryan B. Bond[3]

[1] Sandia National Laboratories
P.O. Box 5800, MS 1318
Albuquerque, NM 87185
pknupp@sandia.gov

[2] Sandia National Laboratories
P.O. Box 5800, MS 0316
Albuquerque, NM 87185
ccober@sandia.gov

[3] Sandia National Laboratories
P.O. Box 5800, MS 0825
Albuquerque, NM 87185
rbbond@sandia.gov

Key Words: *Verification, Numerical Error, Uncertainty Quantification.*

ABSTRACT

Within the Advanced Simulation and Computing (ASC) Verification and Validation (V&V) program, considerable attention is paid to Uncertainty Quantification (UQ), where one seeks to quantify various sources of uncertainty within a computational simulation. Uncertainties that are often considered are those associated with (1) specific parameter values, (2) "modeling error," (3) experimental error, and (4) discretization error due to the discrete approximation of the governing partial differential equations.

There are other sources of uncertainty, having to do with the calculation of the numerical solution that are not usually accounted for in estimating the overall uncertainty. One source of uncertainty investigated here is called the *numerical error (NE)*. Numerical error is essentially the difference between the numerical solution and the exact solution to the PDE, and arises from four sources of error within a numerical calculation: (1) discretization error (DE), (2) roundoff error (RE), (3) incomplete iterative convergence error (IICE), and (4) implementation correctness error (ICE). ICE arises from the presence of coding mistakes (bugs) that prevent the correct numerical solution from being computed.

The main purpose of this study [1] is to obtain some insight into the magnitude and effects of coding mistakes (ICE) on the numerical error and uncertainty in the solutions to PDEs. Although this study could have been performed using a real ASC production code, the numerous complications of using large complex physics code would greatly increase the effort and possibly obscure any general and useful trends (not to mention the possibility of uncontrolled ICE contaminating these results). Therefore a computer code to solve the 1D Poisson's equation was used to investigate the impact of coding mistakes on numerical error and uncertainty. This model PDE has an exact solution, and allows the evaluation of the numerical error (not ICE separately) and its effect on uncertainty.

To obtain a preliminary indication as to the behavior and importance of ICE, a computer code that solves the model 1D partial differential equation is used to explore the sensitivity of numerical PDE solutions to coding mistakes. The approach is to begin with a simple bug-free code and introduce 26 realistic bugs, so that the error due to the bugs (ICE) can be computed by comparing to a known exact solution. Measurement of ICE requires that one control the other components of NE. Having controlled DE, RE, and IICE, we calculate ICE in terms of global error and five "system response quantities", such as the

derivative at a boundary, the maximum solution value, the location of the maximum solution value, the integral of the solution over the domain, and the solution value at a given point.

Because a model problem was used to study ICE, one is faced with the issue of extrapolating any conclusions reached in this study to what might happen with a real ASC production code. Here are a couple of things that cannot be determined from this study: 1. the degree to which these observations carry over to ASC production codes, and 2. how the magnitude of ICE compares to modeling and experimental errors in realistic problems. *Assuming* that the results obtained on the model problem extrapolate reasonably well to complicated production codes, a number of conclusions can be drawn that could be applied to ASC codes.

Analysis of the numerical error for this problem demonstrated several interesting phenomena related to verification and ICE.

- The "winnowing of bugs" (the modest testing for the most egregious bugs) is beneficial in reducing the statistical values of the error metrics. Nevertheless, it appears that the more subtle bugs, which would not be found through the "winnowing" process, do not necessarily have insignificant errors.
- The use of mesh refinement without a known exact solution to find bugs can be useful, but may not catch bugs that allow the solution converge to an incorrect solution. This occurred with several of the introduced bugs.
- The presence of ICE can drastically affect error estimates. The essence of the discrepancy lies in the assumptions of the error estimators, which explicitly assume that the numerical solution converges (with order p) to the exact solution as the mesh is refined (via a ratio r).
- Parameter sensitivity studies can be compromised if a code contains one or more coding mistakes. Results show that the solution can be insensitive or inversely sensitive to a given parameters.

Additionally, the uncertainty in the system response quantities given the possibility of coding mistakes is assessed to obtain an insight into the possible variation in the solution. The problem can be stated as follows: given the 27 possible versions of the code that might have been instantiated, what is the uncertainty in each of the five SRQs calculated by the code? The uncertainty produced by these bugs will be presented and discussed.

In the final portion of this study, reduction of ICE is discussed as necessary for good V&V, UQ, and QMU practice. Assuming this is true, how does one go about curtailing ICE from a code or calculation? How would one go about substantiating a claim that ICE has been neutralized so that uncertainty due to ICE is not a major concern? We attempt to indicate which of the broad set of activities considered to be code verification are the most effective approaches to reducing or eliminating ICE. There are two aspects to this goal that are relevant. First, given due diligence, can an activity catch all the ICE-producing bugs in the critical parts of the code? Second, how easy is the activity to set up and perform, and to evaluate its results? Ideally, a code verification activity should be such that it can produce results within the time frame of the code development cycle.

REFERENCES

[1] P. M. Knupp, C. C. Ober and R. B. Bond. "Impact of Coding Mistakes on Numerical Error and Uncertainty in Solutions to PDE's". *Sandia National Laboratories Technical Report*, SAND2007-5341, 2007.

[2] P. J. Roache. *Verification and Validation in Computational Science and Engineering*. Hermosa Publishers, 1998.