Improvements in RELAP5-3D Verification Capability

George Mesina

June, 2014

Two improvements have been developed in the highly-accurate, automated verification process.

Background

Automated verification is possible through the use of sequential verification. Where traditional computer program verification compares two things: coding against underlying algorithms and equations, and calculations against analytical and manufactured solutions on every version, sequential verification checks only the calculations of every version. Comparison of coding to algorithms and equations is performed for the initial version only. Thereafter, if the calculations must remain the same, comparison of coding and theory is deemed unnecessary. However, when any change in a calculation is found, it must be justified and fully explained. If not, it is treated as an error to be solved.

Sequential verification uses numerical comparison of pairwise consecutive versions to form a trace back to the original coding that was compared against algorithms and equations.

Comparison is made between verification files that store the L_1-norm of primary variables. The primary variables of the governing equations are summed because they are affected by almost every variable in the program and affect every variable in return. If the calculated value of any variable changes, it affects the formation of the governing equations and therefore the primary variables on the very next advancement. To detect changes, the L_1-norm is summed in quadruple precision (128-bit) which exceeds 32-digit accuracy.

Features of the Original Verification Method

Our process is documented in an ICONE-22 paper[1]. Important code features relevant to nuclear power plant modelling were identified and test cases for these collected. The verification test suite contains 43 input decks comprising over 160 separate cases (through the use of multi-case input files) that test over 200 features, many of which are tested by multiple cases. The verification suite is laid out visually as a matrix with the rows representing code features and the columns for input files.

Output to the verification files is user controlled. Users select when to stop and start dumping verification data, and the frequency through a single input card. Thus if a difference is detected, the exact time that a difference first occurs can be discovered by rerunning with dumps every step.

The verification file also allows verification of major processes within RELAP5-3D. One process is restart, which allows the code to continue a new calculation from an intermediate time or final time of a previous calculation. Restarting from an intermediate point does not always reproduce the exact same calculations.

RELAP5-3D performs backups, a repeat of an advancement with the same timestep and slightly modified equation. The conditions are: water-packing, velocity direction-flip in a junction, and appearance of a noncondensable in a control volume that previously had none. Via verification input, the user can force the code into behaving as if a backup occurred to test it. Standard testing forces a backup on every advancement so that backup process can be tested in a variety of flow conditions.

Verification can be used to track down the source of the error. Once a difference is recognized on a verification dump, verification can be activated for every advancement prior to a that time ti find the first advancement on which any difference occurred. It also helps locate the source of the change by isolating the primary variables involved.

Development of New Verification Capabilities

Both the availability of the verification process on different computer platforms and the major RELAP5-3D processes that can be verified have been expanded.

The first development has been the porting of the verification test suite to MS Windows platforms. Originally, the verification file was developed on SUSE Linux platform for use on native Linux platforms. A Makefile has been developed that runs in the native DOS form through the windows utility NMAKE. During the past three months, a Makefile has been developed for the Linux emulator, CYGWIN. Unfortunately, there are sufficient differences between the three make utilities that all three Makefiles are different.

The second development is the capability to test multi-case input. For some multi-case input files, a later case will fail, yet that same case will run as an individual input deck. A user questioned whether the same calculations were produced when individual cases were run as opposed to multi-cases. A C-shell script was written that uses Linux utilities to rip a multi-case deck apart and form a pair of input decks for each case. The first deck runs only the individual case. The second runs all the cases up to and including that case. Each member of the pair dumps verification output only for the case under examination.

This latter development has shown that in all but one of the 43 input decks from the verification test suite, all multi-case calculations are exactly the same as the separate case runs. The singular case has been submitted as a user problem.

Reference

1. G. L. Mesina, et al, “Automated, Highly-accurate Verification of RELAP5-3D, ” Proceedings of the 22nd International Conference on Nuclear Energy, Prague, Czech Republic, July 7-11, 2014.