RELAP5-3D Verification 2014

Dr. George Mesina

RELAP5 International Users Seminar

Idaho Falls, ID

September 11-12, 2014

Outline

- Verification status in 2013
- Progress on code issues revealed by verification
- New Verification testing
- New Verification capability

Verification Capability: Summary

- Automated verification capability introduced in 2013 to detect code errors (detection) for many code features (coverage)
- DETECTION: RELAP5-3D writes a verification file that records primary variable sums to 32+ decimal place
- COVERAGE: The verification test suite has
  - Tests 194 code features
  - 43 test problems with 125 input cases
- Comparing verification files for the same input reveals changes between code versions or application of code capability
- THEOREM: Verification Comparison NEVER finds differences falsely.
  - Provided it is programmed correctly.

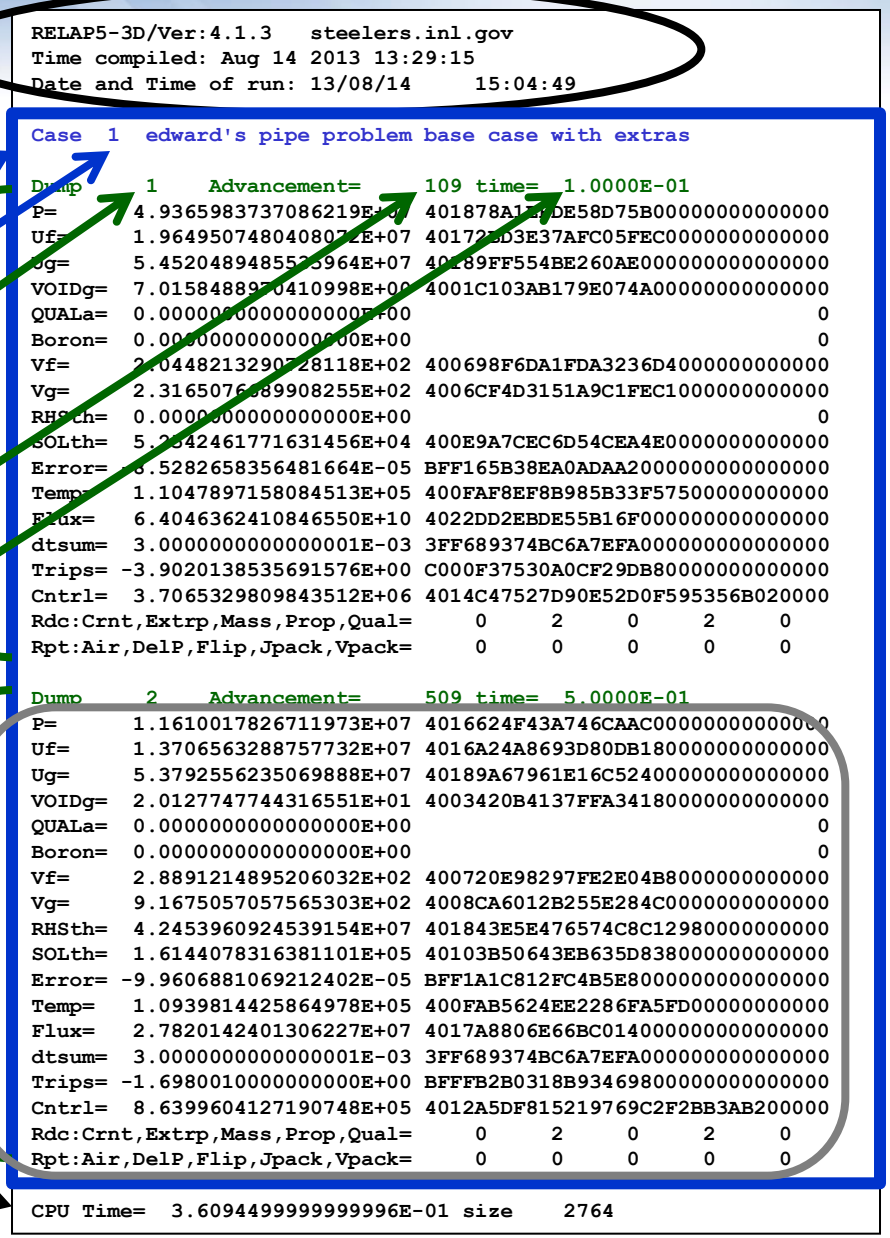
COVERAGE: Summary of Code Features Tested

Feature Category	Number of features
Hydro component	29
Volume flag - tlpvbfe	7
Wall friction options	6
Junction flag - jefvcahs	14
Junction form loss	4
Flow regimes	6
Heat structure type	3
Heat transfer modes	7
Heat structure BC types	8
Heat source options	5
Material Prop	3
Metal-Water	3
Subtotal	95

Feature Category	Number of features
Enclosure	2
Reactor kinetics	12
Decay Heat	11
Trips	2
Control variables	32
Tables	8
Flow regimes	6
Equation Solvers	5
Card 1 Options	9
Proprietary	5
Other Major Options	7
Subtotal	99
Total	194

DETECTION: Verification File

- **Header**
  - Code & Computer ID
  - Data/Time Compiled
  - Date/Time Run
- **Input Case**
  - Case # & Title
- **Dump**
  - Dump #
  - Advancement #
  - Cumulative Time
  - L_1 norms
- **Footer**
  - CPU Time
  - Bytes (Upper Limit)



Verification Capability: 2013 Summary

- Automated verification capability introduced in 2013.
  - Tests 194 code features via 43 test problems, with 125 input cases
  - Records primary variable sums to 32+ decimal places
- Findings based on comparing two verification files
  - Null test compares between two versions
  - Restart compares base run and run restarted from middle
  - Backup compares base run to one that repeats every timestep
- Failure means that at least on sum was not the same.

Version 4.1.3	Failures in 43 Test Problems	Failures in 125 cases
Null Testing	6/43	6/125
Restart Testing	25/43	52/125
Backup Testing	37/43	62/125

Verification Capability Revealed Issues

- Inexact Restart
  - Half caused by failure to write required data to restart file.
    - Some recent modules did not have restart read/write routines.
  - Other half caused by a difference in final bit of the cumulative time.
    - Resolved by updating certain “integer time” information at edit points rather than at timecard end times only
- Example
  - ANS79 has 9 cases. Four failed that represented more than one nuclide for decay heat.
  - The variable that recorded that was not written to the restart file.

Verification Capability Restart Issues

- Other half caused by a difference in final bit of the cumulative time.
- The issue was that the cumulative time did not match across a restart.
  - TIMEHY = “Start of timecard” + “integer time”*dt_{small}.
  - On a restart, there were two situations
    - From end of a timecard, integer time was zero
    - From middle of timecard it was non-zero
    - For large integers, the product could be off in the final bit due to the number of digits involved
- Solution was to “recalibrate at edit times” by changing to:
  - TIMEHY = “Prior edit” + (“integer time” – “integer edit time”)*dt_{small}
  - This solved the verification restart issue
- However, it caused three PVM installation problems to fail

Verification Capability Revealed Issues

- The PVM problems that failed were synchronously coupled.
- The solution was to introduce the same recalibration:
  - In the PVM DTSTEP subroutine
  - In the related section of RELAP DTSTEP
  - This solved the PVM installation problem issue while keeping the verification restart solution

Verification Capability Revealed Issues

- Inexact Backup
  - Most caused by failure to record an old-time value of some important quantity, such as QUALE.
    - Resolved by adding an old-time quantity to the relevant module data (usually a derived type).
  - Most changes occur in subroutine MOVER which restores data to previous timestep values on a backup.
    - Not all were in MOVER. Some were in TRAN, HYDRO, and DTSTEP.
  - Changes to VERFBACKUP were necessary to resolve an issue with the repeat-count sums.

Verification Capability Progress

- Progress on original 43 Test Problems

Test Case	4.1.3 Failures	4.2.1 Failures	4.3.1 Failures
Null Test	6	0	0
Restart Test	25	18	0
Backup Test	37	20	0

- Improvements for correcting the verification issues do NOT cause INL standard test cases to fail.

New Verification Capability and Modifications

- Added verification capability
  - New input decks added
  - Multi-case input decks
  - PC verification
  - Input modification

New PVM Verification Problems Supplied

- New non-PVM input decks added
  - Increase coverage
- New PVM Coupling Problems
  - Backup issues
  - Restart issues
- These required expansion of the Verification Directory and Makefiles
  - Each new problem type has its own subdirectory with Make.inp
  - The subdirectory Makefile now has PVM null, restart, and backup targets
  - A new Input variable for PVM tests added to Make.tests
  - The main Makefile can run all PVM null, restart, and backup tests

New PVM Verification Problems Supplied

- Backup issues had a variety of causes, including:
  - Backups cannot proceed from the first step after an explicit coupling exchange with a leader-follower.
    - Had to postpone backups till the second advancement
  - TestMatrixDT coding was triggered by setting the air appearance flag. This was changed to setting lsuces = -1.
    - This allows a section of coding in subroutine MOVER to be exercised
- Some restart problems fail because the time step does not match.
  - One fix is resetting integer time, ITIME, on restart
  - Another is converting integers to real in quadruple precision
- These problems are being worked.

Added Verification Capability

- Multi-case input deck testing
  - User found the code failed in a multi-case input, but when run as two separate input decks, both decks ran
  - Questioned whether the code produced the same calculations in multi-case form.
- Test with verification capability
  - Script written to break an input deck with N cases into N pairs of input decks
  - E.G. for deck.i with N input cases, these are named
    - Deck.cK.i **c**ollapses the first K cases into one by eliminating the “slash” input case separator
    - Deck.sK.i runs the first K cases **s**eparately by keeping slashes
  - The runs are compared

Added Verification Capability

- Only two of the 43 verification test suit input problems show differences
- PC verification Makefiles
  - The Makefiles that run the verification test suite for Linux have been rewritten to run on the Windows 7 PC.
  - There are some differences in the way the DOS nmake utility operates compared with Linux make.
- Running showed that RELAP5-3D/version 4.1.3 had the same performance on Windows 7 as Linux

Change to 199 Card Verification Control Card

- The 199 activates verification
  - 199 Word(1) Word(2) Word(3) Word(4)
  - Word(1) can be “verify” or “noverify”
  - Word(3) = start, *integer* advancement *or real* time
  - Word(4) = shut off-advancement control (integer)

These three stay the same.

- Word(2) values <u>will change</u>. In 4.2.1 these are the values:
  - dump – write verify dumps on specified steps
  - backair – backup for air (non-condensable) appearance
  - backpck – backup for water packing
  - backvel – backup for velocity flip-flop
  - backall – backup every timestep, 2 forward / 1 back
- **<u>Change</u>**: Starting in 4.3.1, eliminate backpck, backvel, and backair

SUMMARY

- The verification capability is being used to locate code problems with:
  - Unexpected calculation changes going from version to version
  - Restart issues
  - Backup issues
  - Multi-case issues
  - PVM coupling issues
- All issues uncovered with the original Verification test suit problems have been solved (as of version 4.3.1).
- New issues have been reported with multi-case and PVM coupling. These are being worked
- New verification capability has been developed.