Selected User Problem Resolutions

George Mesina

April, 2017

The resolutions for several recent user problems are described and explained. Three are error fixes and two are code upgrades.

Introduction

Various recent user problems were resolved in the last quarter. Some of the highlighted fixes include user problems 17004, 17006, 17008, 17009, and 17012. Each problem is identified by User Problem (UP) number and submission date, the category of user problem, its title. A detailed description, comments about the work (if more involved than normal), and status of the user problem are reported. All problems reported here were resolved by Dr. George Mesina (GLM). The problems reported are:

1. Compressor rcpr.i fails with debug Windows version of RELAP5-3D/4.4.0
2. Cannot read or write CSV plot files
3. Collect Common Modules
4. Multi-deck kinetics adaptive timestep failure

The Selected User Problems

1. UP 17004 from 1/2017

- TYPE: Code execution failure – Compressor error
- TITLE: Compressor rcpr.i fails with debug Windows version of RELAP5-3D/4.4.0
- DESCRIPTION: With Version 4.4.0, the compressor rcpr.i fails with debug Windows version of RELAP5-3D/4.4.0 in cprssr.F. Found that the failure is caused by accessing pcmp(mn)%p4.
- EXPLANATION: The compressor fails with version 4.4.0 at line 317 of cprssr.F called from vexplt.F line 399. The statement is "s = pcmp(mn)%p4" and is bypassed to statement number 294 by the previous block of code if pcmp(mn)%p4 is not associated. With debug on, this pointer has a target that is a bad value. This coding is very similar to pump.F coding. Removing cards 3500500 through 3500506 and making an appropriate change to card 3500301 allows the code to run rcpr.i to completion. By comparison to version 2.4.2, variable "s" is pmpold, the pump speed in RPM.
- STATUS: RESOLVED (GLM): With Version 4.4.0, it was found that the compressor rcpr.i fails with debug Windows version of RELAP5-3D/4.4.0 in cprssr.F if pcmp(mn)%p4 is accessed. The test that accessed it checked if it had been assigned or not, which would work if it have been either assigned

or nullified. Located the section of coding where nullification or setting occurs is in ICOMPN. Fortran 2003 does not have an intrinsic function to check if a pointer has been nullified, therefore when the user inputs a pump speed table, the pointer pcmp(mn)%p4 must be nullified first and then set from the input if appropriate.

2. UP 17006, 2/2017

- TYPE: Code capability improvement – Plotting upgrade
- TITLE: Cannot read or write CSV plot files
- DESCRIPTION: An INL analyst noticed that the code can write a CSV strip file and recognizes the CSV keyword for reading a plot file, but does not implement the reading of CSV plot files. Currently, CSV input (103 card) defaults to ASCII format input for strip problems. In fact for new problems, the CSV format specifier on the 104 card produces an error message.
- STATUS: RESOLVED (GLM): Module plotmod and several routines are upgraded to handle specification of CSV input for both new and strip options and allow specification of csv on the 104 card for new problems. Specifically, module plotmod and subroutines PLTWRT, RRESTF, RRSTD, STRIPPLOT, and WRPLID were upgraded to correctly process specification of CSV input for both new and strip options and to allow specification of CSV on the 104 card for new problems. Coding was added to PLOTMOD internal subroutines to write plot and strip files in CSV format. Further, STRIPPLOT was reprogrammed to read CSV input from a plot file for stripping. A new inputfile, plotcsv2strip.i, tests it.

3. UP 17008, 2/2017

- TYPE: Generic/Installation – Architecture Upgrade
- TITLE: Collect Common Modules
- DESCRIPTION: With many Fortran modules becoming shared between two or more directories, this task seeks to eliminate duplicates both for maintenance, efficiency, and reduction of complexity. Accessing only one copy of each module reduces storage, but more importantly, eliminates the chance that future updates will leave one or more copies unmodified, as has happened. Common modules have been collected into the modules directory from the following directories:
  - envrl – environmental
  - jacdir – Jacobian evaluator
  - polate – Fluid property interpolation programs
  - relap – source code for the RELAP5-3D program
  - r5exec – source code for the RELAP5 Executive program

  Many of these require lower level modules to build, so all of those must be included in directory modules.
- NOTE 1: These modules are removed from all other directories.
- NOTE 2: modules in the fluids directories that have the same names, but are different, are not included in the modules directory because they are intended to be distinct.

- STATUS: RESOLVED (GLM): All Fortran modules common to two or more directories are stored in the modules/ and accessed via soft-links. The "softlink" target is available in the make-files of all affected directories, namely, envrl/, jacdir/, polate/, r5exec/, and/ relap. The common modules are built in the directories themselves. In relap/Makefile, new target "depend" makes the dependencies on both *.H files and modules and places them at the bottom of relap/Makefile.

4. UP 17009, 3/2017

- TYPE: Code execution failure – Time-stepping failure
- TITLE: Mutli-deck kinetics adaptive timestep failure
- DESCRIPTION: For a large ENEA input model with NESTLE kinetics, the RELAP5-3D timestep cards (201-299) and the adaptive timestep cards (2201 - 2299) had identical edit times. The dt for the kinetics, dtkin, became zero at the final time, according to the dtk-file, and the code went into an infinite sub-cycling loop in the kinetics section. A solution was submitted by the user for testing.
- COMPLICATION: When the solution was implemented, the ENEA problem ran, however testing on other kinetics time-step input models revealed that they did not all work with the adaptive time-step. This was traced to the comparison of dtk, the upcoming kinetics time-step, being compared to zero for a particular stopping condition. Because dtk resulted from a floating point subtraction of the kinetics and thermal-hydraulic cumulative times, it non-zero value related to unit round-off.
- STATUS: RESOLVED (GLM): This new issue was submitted as UP17012. The code changes were submitted for inclusion in version 4.4.1. The original issue, UP17009, is resolved.

5. UP 17012, 4/2017

- TYPE: Code execution failure – Time-stepping failure
- TITLE: Mutli-deck kinetics adaptive timestep failure
- DESCRIPTION: In the KinDt subset of release test cases, problem hex2dk-dt1 fails when the kinetics time-step, dtkn, becomes zero at the end of the transient. The hex2dk-dt1.dtk file shows that the kinetics coding goes into sub-cycling with a zero time-step and can never advance to the end time, even though the kinetics cumulative time equals the TH time to the number of decimal places printed. The problem is a restart of hex2dk-dt0, but is being listed as a code execution failure rather than a restart failure. NOTE: This does not occur for the developers of the kinetics adaptive time step coding at ISL (Innovative Systems Laboratory) with their operating system and compiler.
- SOLUTION: One problem was traced to use of floating point subtraction that produces unit roundoff related values (UROUND times magnitude of numbers subtracted) which was compared to zero in some tests. In debugging on the phone with Doug Barber of ISL, we found and corrected another problem calculating the logical variable LLOCK. Converting the floating point subtraction that was involved into an integer time subtraction with multiplication by dtsmallest solved the problem. Afterwards, I created a data dictionary for dtkin.F to help understand the logic, converted the

floating point subtractions to integer subtractions multiplied by dtsmallest and converted tests to use integer time.

- COMPLICATION: All the KinDt test cases except typpwrr2-krlv-nem-dtExtrapLong.i now run. The code changes were submitted for inclusion in version 4.4.1.
- STATUS: RESOLVED (GLM): This new issue was submitted as UP17013. The original issue, UP17012, is resolved.