Order of Evaluation Issues, Part 2

George Mesina and Hope Forsmann

December, 2013

Left-to-right evaluation of Fortran if-clauses does not occur on all platforms. This issue causes errors for RELAP5-3D and other legacy programs. The method used to solve this is explained.

Background

ANSI specifications for the Fortran programming language[1, 2] do not require mathematical and logical expressions to be evaluated in left-to-right order. Many legacy Fortran language programs, such as RELAP5-3D, were developed on single processor platforms where this was not an issue. However, on platforms with multiple processing units and multiple cores, core dumps and incorrect results can result if the coding was developed so that left-to-right evaluation is required for correct processing.

In the 2013 First Quarter RELAP5-3D Newsletter[3], an article identified and explained the issue of left-to-right evaluation order for if-statements with AND-operands. The article developed means to solve the problem and reported on the work done to correct the issue. It also stated that the same issue could occur with OR-operands, but that no work would be undertaken until such an issue was received as a user problem report. This occurred in October 2013.

The or-operand issue is addressed in this article. In addition, ELSEIF-statements may have and-operators and or-operators connecting clauses in logical expressions and must be checked. Besides the RELAP directory, both IF and ELSEIF statements occur in subprograms contained in the environmental, fluids, and PVM-Executive directories.

This is important for many legacy computer programs besides RELAP5-3D.

Examples

For a logical statement comprised of many logical expressions connected only with OR-conjunctions, the entire statement is true if any of the expressions is true. To save runtime, many older Fortran compilers would cease evaluation of the if-statement as soon as one of the expressions evaluated as true. When the assumption of left-to-right evaluation was made, an expression to the left of another may have protected it. For example,

```
IF (j.le.0 .OR. a(j).eq.1.0) GO TO 100

IF (number.lt.0.0 .OR. sqrt(number).gt.1.0) GO TO 100
```

In the first example, left-to-right evaluation would prevent a(j) from being accessed when j is -1. However, an error occurs on multi-functional-unit platforms that evaluate the entire statement. In the second, the square root function is protected from acting on a negative value only for left-to-right evaluations. Many other potential errors can occur. These were reported in the first quarter article[1].

To correct the issue, the solution is to break such a statement into two if statements:

```
IF (j.le.0) GO TO 100

        IF (a(j).eq.1.0) GO TO 100
```

Analysis

In each of the four major source code directories of RELAP5-3D, the if-tests were examined. Table 1 shows the number of if-statements in version 4.1.3. As the first column indicates, there are over 43000 if-statements to consider. It would be inefficient to examine all of them, so simple requirements were identified to reduce the number needing visual examination. To qualify, an IF or ELSEIF statement must contain an AND/OR conjunction and two left parentheses, the second of which may signal an array. The simple requirements reduce the number that must be examined by more than an order of magnitude, but there still remain nearly 3000 to visually examine.

TABLE 1. IF/ELSEIF Counts by directory and kind

Directory	Active IF/ELSEIF Statements	Qualifying statements with AND-operator	Qualifying statements with OR-operator
Fluids	8275	100	164
Environmental	554	39	12
PVM Executive	961	38	17
RELAP	33426	1171	1312
TOTAL	43216	1348	1505

A second consideration is that many IF/ELSEIF-statements in RELAP5-3D span numerous lines of code. It is not sufficient to use a utility, such as grep, to pull all qualifying statements that contain an IF or ELSEIF into a file.

To aid manual examination of these, a program was written that combines IF- and ELSEIF-statements into single lines of potentially 2000-character length. With a Fortran 95 program, additional restrictions are simple to program. The code checks that:

- There is an AND- or OR-operand.
- There is an index-variable which is a variable name near a left-parenthesis that is preceded by a legitimate name for an array or function subprogram
- There are multiple occurrences of the index-variable

The list of index-variables is printed above each qualifying IF/ELSEIF -statement. Typically, an evaluation order issue occurs when an index-variable occurs outside an array reference, as in the first example. These “isolated” variables are marked with a special message above the qualifying statement.

Result

Beside the statements reported in Part 1 of this article, only five more were found and corrected by this effort. All involved the OR-operand and occurred in the relap directory. No issues were detected in the environmental, fluid properties, or PVM Executive directories among all the IF/ELSEIF-statements with either AND- or OR-operands.

These IF/ELSEIF-statements were corrected and will appear in the next code release in 2014.

References

1. J. C. Adams, et al, FORTRAN 90 Handbook, Complete ANSI/ISO Reference, McGraw-Hill, Inc. 1221 Avenue of the Americas, New York, Y, 10020, ISBN 0-07-000406-4, 1992.

2. J. C. Adams, et al, FORTRAN 90 Handbook, The Complete Syntax, Reatures and Procedures, Springer Science+Business Media, springer.com, ISBN 978-1-84628-378-9, 1992.

3. G. L. Mesina, “Order of Evaluation Issues,” RELAP5-3D Quarterly Newsletter, http://www.inl.gov/relap5/news/2013/q1/2013-1q-r5-3d-evaluation-issues.pdf, April, 2013.