MEMORANDUM

ERI/NRC 2020-09-01

To: D. Palmrose and J. Davis (NRC/NRR)

From: M. Zavisca, E. Ball and M. Khatib-Rahbar

Subject: Verification and Validation of RADTRAN 6.02.1 and NRC-RADTRAN Graphical User Interface Version 0.99

Date: September 1, 2020

1. INTRODUCTION

The RADTRAN computer code is a command-line program used to calculate the risk involved in transport of radioactive materials by highway, rail, or water, which dates back to 1977. A graphical user interface (GUI) for RADTRAN was created by Sandia National Laboratories (SNL) in the Java programming language and was named RadCat. Pursuant to the statement of work for contract NRC-HQ-25-14-E-0005, Task Order number NRO-19-0039-EWC-UN-46-17, Energy Research, Inc. (ERI) is required to:

1. Make the NRC-specified changes to the Fortran source code of RADTRAN version 6.02 and recompiling it into a new version, to be denoted RADTRAN 6.02.1. The changes were not expected to be numerically or functionally consequential, other than to remove date-based license restrictions and to permit the code to compile on modern platforms.
2. Modernize the GUI by rebuilding it in C# using Visual Studio with Windows Presentation Foundation (WPF), to make the interface more visually pleasing and intuitive to use, as well as to add various functionalities and bring it more into line with other current-day interfaces. The new interface is denoted NRC-RADTRAN.

This work was completed and the new version of RADTRAN 6.02.1 was built and delivered to the NRC, while the release candidate of the NRC-RADTRAN GUI (denoted Version 0.99) is the subject of the present verification and validation (V&V) effort. Note that, with this V&V having approved that release candidate version 0.99 functions properly, it was subsequently relabeled as version 1.0 for release, with no functional changes other than to labeling and to the supplied user documentation.

NRC has provided a baseline for this V&V by assembling a set of 15 RADTRAN input files and running the cases using a version of RADTRAN 6.02 available at the NRC. These input and output files were delivered to ERI for use in the V&V process.

The V&V process consisted of the following:

1. Run the input files provided by NRC from the command line using the ERI-recompiled version of the code, RADTRAN 6.02.1, and verify that the results match or are equivalent to those obtained by the NRC using RADTRAN 6.02.

2. Load the input files provided by NRC into the NRC-RADTRAN version 0.99, launch calculations from the GUI, and verify that the results also match the originals.

Section 2 of this memorandum describes the 15 cases provided by the NRC. Section 3 documents the results of the RADTRAN 6.02.1 command-line code verification, while Section 4 contains the results of the NRC-RADTRAN version 0.99 verification. Results and insights of the V&V process are summarized in Section 5.

2. DESCRIPTION OF CASES

ERI was provided by the NRC 15 RADTRAN scenario input files, along with the corresponding output files generated by the NRC using the version of the RADTRAN 6.02. These cases are listed in Table 1. It can be seen that there are actually only four completely independent scenarios, with the others being relatively minor or inconsequential alterations of the base ones:

1. The “Barnwell” scenarios (2, 3, and 4) differ only with respect to the comment fields and the formatting of the input (not the numerical values).
2. The second “Richland” scenario (6) differs from the first (5) only in what appears to be a correction to one erroneous input parameter value in the LINK input.
3. The various “Yucca Mountain” scenarios (7 through 15) differ only in that they use package inventories corresponding to different reactors or fuels, and in that the “R1” set correct what appears to be one erroneous input parameter value in the RELEASE input for Crud.

One fact that became immediately apparent about the 15 input files supplied by the NRC was that most of the files specify output verbosity level 4 (i.e., the third input parameter on the PARM input line). This verbosity level is no longer supported in the version of the RADTRAN 6.02 source code that was used by ERI to generate RADTRAN 6.02.1, both of which will automatically downgrade higher verbosity levels to a maximum value of 3. This implies that the build or branch of RADTRAN 6.02 used by the NRC to run these cases does not correspond to the source code used by ERI, and also that the output sections written by the code will not be entirely the same.

Another common feature of the supplied input files is that there is a non-zero number of LOS probabilities entered on the PARAM line, but there are no actual LOS probability data entered in subsequent input blocks. This is permitted by RADTRAN, even though it is functionally equivalent to there being no LOS probabilities.

3. VERIFICATION OF RADTRAN

All 15 input files listed in Table 1 were run from the command line using RADTRAN 6.02.1 Build AC, which was the modified and compiled version of the code created by ERI using a copy of the RADTRAN 6.02 source code supplied by the NRC. Other auxiliary input files (i.e., RT6_Defaults.INFILE, RT6_INGESTION.BIN, RT6_Isotope.INFILE, and RT6_Standard.INFILE) were drawn from the files that were supplied with the RADTRAN 6.02 source code. It is presumed that NRC used these same copies to generate their versions of the output files, since, from the file dates, it appears that these are relatively static (the last change to any of the above files was made in July of 2013). The output verbosity level (i.e., third input parameter on the line beginning with the PARM keyword) was changed from 4 to 3 where necessary, since RADTRAN 6.02.1 does not support levels higher than 3. Copies of all of the input and output files generated as part of this exercise are supplied in the archive accompanying this memorandum.

Table 1 Description of Verification and Validation Cases

Number	Name	Description
1	Generic_FF	Generic unirradiated fuel shipment
2	PSEG_7-4_Barnwell_RW	Highway transport of waste to Barnwell facility
3	PSEG_7-4_Barnwell_RW_copy	Highway transport of waste to Barnwell facility (comments added)
4	PSEG_7-4_Barnwell_RW_R1	Highway transport of waste to Barnwell facility (changed line order and formatting and numerical formatting)
5	PSEG_7-4_Richland_FF	Highway transport of fresh fuel from Richland, Washington to PSEG
6	PSEG_7-4_Richland_FF_R1	Highway transport of fresh fuel from Richland, Washington to PSEG (corrected one erroneous value in LINK input data)
7	PSEG_7-4_YM_SNF	Transport to Yucca Mountain of irradiated, unidentified fuel
8	PSEG_7-4_YM_SNF_R0_ABWR	Transport to Yucca Mountain of irradiated ABWR fuel
9	PSEG_7-4_YM_SNF_R0_AP1000	Transport to Yucca Mountain of irradiated AP1000 fuel
10	PSEG_7-4_YM_SNF_R0_USAPWR	Transport to Yucca Mountain of irradiated USAPWR fuel
11	PSEG_7-4_YM_SNF_R0_USEPR	Transport to Yucca Mountain of irradiated USEPR fuel
12	PSEG_7-4_YM_SNF_R1_ABWR	Transport to Yucca Mountain of irradiated ABWR fuel (corrected one erroneous value in Crud release input data)
13	PSEG_7-4_YM_SNF_R1_AP1000	Transport to Yucca Mountain of irradiated AP1000 fuel (corrected one erroneous value in Crud release input data)
14	PSEG_7-4_YM_SNF_R1_USAPWR	Transport to Yucca Mountain of irradiated USAPWR fuel (corrected one erroneous value in Crud release input data)
15	PSEG_7-4_YM_SNF_R1_USEPR	Transport to Yucca Mountain of irradiated USEPR fuel (corrected one erroneous value in Crud release input data)

In all cases, the output files differ in the following inconsequential ways, in addition to any consequential ones:

1. The listing of the file version (in the headers of the echo input and output sections) specifies 6.02.1 Build AC, rather than 6.02, and
2. Dates and times in the page headers correspond to the time at which ERI ran the cases instead of the dates and time NRC ran them.

The following other differences are noted for each major case that was examined:

1. Rarely, a few numerical values are off by 1 in the last significant digit in the output. These are presumed to be inconsequential discrepancies arising from floating-point precision combined with differences in Fortran compilers and compilation options. RADTRAN floating-point output values are generally represented with three significant digits. The IEEE 754 floating-point specification includes 23 bits for the mantissa, which corresponds to approximately 7 significant digits in decimal form. Therefore, randomly distributed numerical differences in the last significant digit of an internally represented floating-point number might be expected to change the third significant digit of an output value in about one out of 10,000 numbers (e.g., cases where 1.584999 gets flipped to 1.585000 due to imprecision, changing the rounded three-digit value from 1.58 to 1.59). Since an average RADTRAN output file contains several thousand numbers, it is expected that this would happen in approximately the frequency observed (i.e., one case in each of scenarios 6 and 13, and no cases in scenarios 1 or 4).
2. NRC's outputs include sections for "50-Year Population Dose in Person-Sv" and "50-Year Societal Ingestion Dose-Effective" which are absent from the ones generated in RADTRAN 6.02.1. Examination of the source code verifies that these two sections are only output at verbosity level 4, which is currently disabled in the code, forcing us to run using verbosity level 3.
3. ERI's outputs include sections for "Incident-Free Importance Analysis Summary", which are absent from the ones provided by NRC. Examination of the source code verifies that this section is only output at verbosity level 3 (not at level 4), and therefore it is simply an artifact of our being forced to use a lower verbosity level.
4. In rare cases, RADTRAN 6.02.1 outputs a different number of casks compared to RADTRAN 6.02. Note that the number of casks is calculated using the volume of a cask and the volume of resin, based on values in the ECONOMIC section of the input. The cases for which there is a discrepancy in cask count are ones for which no ECONOMIC input is present, and therefore dummy or default values defined in the Fortran code for RADTRAN are being used. In our view, for these cases the number of casks is therefore a meaningless result, and the discrepancy is unimportant. It is interesting, nonetheless, and it perhaps derives from the same difference noted in item #1, above; if the volume of resin is exactly divisible by the cask volume, then small numerical differences in the last significant digit could affect this result, since the number of casks is rounded up to an integer value (e.g., 1.00000 gives 1 cask, while 1.0000001 gives 2 casks).

After allowing for differences in code version, run time, and output verbosity level, the only differences is a rare discrepancy in the last significant digit of a floating-point output value, which appears likely to arise from unavoidable differences in the Fortran compiler and compiler optimizations used to create the executable; and a rare difference in the calculated number of casks in cases where no cask data (i.e., ECONOMIC input block is present, which perhaps derives from the same numerical difference. Both of these discrepancies are deemed ultimately unimportant.

4. VERIFICATION OF GRAPHICAL USER INTERFACE

The NRC-supplied files for the four main scenarios described in Table 1 were also loaded and run using the latest version of the NRC-RADTRAN graphical user interface as of the start of the final V&V effort (version 0.99). Note that version 0.99 is functionally identical to the version 1.0 delivered to NRC as the formal release, since the release candidate was approved and simply

relabeled at the conclusion of this V&V. Since NRC-RADTRAN includes the same build of RADTRAN 6.02.1 to actually perform the calculations, this exercise is performed mainly to verify that the GUI is exporting the input files properly. For each of the four cases examined, the output files created by NRC-RADTRAN following completion of the calculations are extracted from their automatically generated folders and compared with the output files that were written during the RADTRAN V&V (Section 3).

All 15 input files listed in Table 1 were also run using version 0.99 of the NRC-RADTRAN graphical user interface. Note that NRC-RADTRAN runs calculations using the same version of the RADTRAN executable examined in Section 3 of this memorandum. Therefore, any substantial differences arising in this portion of the V&V would be expected to derive purely as a result of automated input manipulation or reformatting by the GUI. For all cases, the input was successfully read in by the GUI, with only the warning message that files with a specified output verbosity of 4 were being automatically downgraded to verbosity level 3.

Differences between the output files generated by running input files from the GUI, as compared to running them with RADTRAN from the command line, are as follows:

1. The echo input sections of the files are significantly different, which is to be expected. The input format for RADTRAN is very free regarding the order of input lines and blocks, and regarding the number of fields per line (e.g., if ten floating-point values are expected, the program will continue reading more lines until it obtains that many numbers). NRC-RADTRAN writes input files in a specific order and format which generally will not match exactly that of the original input files. However, it was verified for all four main V&V cases that the input exported by NRC-RADTRAN is equivalent to the original file (i.e., all parameters are present with the same values).
2. As part of remark #1, it should be noted that the numerical values exported by NRC-RADTRAN are theoretically equivalent but often in a different format. For example, “1.0” may be written as “1”, or “5.6E01” may be written as “56”.
3. Rarely, one or a few output values differ by 1 in the last significant digit.
4. In the listing of parameter importances in the “Incident-Free Importance Analysis” section, occasionally the parameters are ranked in a slightly different order. These parameters are listed in descending order of the importance value, and differently ordered parameters all share in common that they have identical listed importance values (i.e., they are tied). It appears likely that this is a consequence of the phenomenon of remark #3; a tiny imprecision in the value of a parameter, even one small enough not to show up in the three-figure values that are written to the file, can cause it to move above or below others with theoretically equal importance.
5. In RADTRAN, it is permissible to enter a non-zero number of LOS probabilities but then fail to enter any LOS data for them. NRC-RADTRAN enforces consistency by requiring that the number of LOS probabilities on the PARAM line of the input file equal the dimensionality of the LOS data entered (or zero, if there are none). Most of the input files supplied by NRC for the V&V effort have a non-zero number of LOS probabilities on the PARAM line but no actual LOS data. This is reflected in the NRC-RADTRAN output in that (a) the echo input section sets the number of LOS probabilities to zero, and (b) the results section for the non-GUI output includes several blocks of LOS data that are either empty (i.e., column headers only with no rows of results) or consist entirely of zeroes. In other words, the original output may indicate 18 LOS probabilities then followed by blank tables or tables with 18 rows of zeroes; whereas NRC-RADTRAN’s output indicates zero LOS

probabilities and no LOS output tables whatsoever. This is intended behavior from the standpoint of NRC-RADTRAN and not a substantial difference in function.

While it is understandable that differences in compiler and compilation options could cause infinitesimal differences in arithmetic (such as were remarked upon in Section 3), it is less clear why the GUI could introduce this behavior. An examination of the RADTRAN input parsing routines sheds some light on the matter. Instead of reading floating-point values in free format, RADTRAN micro-manages the process with a complex routine that reads in the field character-by-character and uses arithmetic to successively build the floating-point value after each character read. A consequence of this is that the arithmetic differs depending on the input format. For example, an input field of "1" would be reconstructed as $(1)(1.0)(10.0)^{(0)}$, whereas a field of "1.0" would be reconstructed as $(1)(10.0)(1.0)(10.0)^{(-1)}$. Although both expressions are theoretically equal to 1, the different arithmetic introduces the possibility of imprecision in the last significant digit.

The hypothesis that input format is resulting in differences in arithmetic precision in RADTRAN was tested by taking the input file generated by NRC-RADTRAN for one of the main V&V scenarios (13, the Yucca Mountain AP1000 radwaste scenario) and manually adjusting all numerical fields to match the formats in the original file. For example, if a field was "0.0" in the original NRC file and written as "0" by NRC-RADTRAN, we reverted it back to "0.0" in the modified file. The modified input file was then run through RADTRAN 6.02.1 on the command line, and the resulting output file was compared to the one that was written as part of the RADTRAN V&V (Section 3). The result was that the files were exactly identical other than the page-header dates and times, and in the line ordering/formatting of the echo input section. This confirms that the exact formatting of numerical values in the input file, including that inadvertently performed by NRC-RADTRAN, can marginally affect the results in the last significant digit. The magnitude of this problem does not appear to justify improvements in either program.

6. SUMMARY AND RECOMMENDATIONS

Verification and validation of the RADTRAN 6.02.1 computer code (executed from the command line) and on the NRC-RADTRAN 0.99 GUI using the 15 input and output files supplied by the NRC. Because many of the supplied cases were minor variants of one another, only four cases were examined in detail, with the others subjected to more cursory study. The following observations are noteworthy:

1. The build or branch of RADTRAN 6.02 currently in use by the NRC does not correspond to the source code made available to ERI at the start of this contract, with the most visible example being the fact that it permits output verbosity level 4, which is unavailable in the available source code. Therefore, any changes in behavior could reflect not only those due to a new compiler or ERI modifications to the code, but unknown differences in the two initial code bases.
2. Despite remark #1, it appears that the outputs are functionally identical, with the numerical values being off by 1 in the last significant digit, in rare cases when they differ at all. This is most likely due to differences in implementation of arithmetic or optimizations in different Fortran compilers, although it is possible that the unknown differences in the two initial code bases also contribute.
3. Because the V&V cases were run with output verbosity level 4, and RADTRAN 6.02.1 is forced by the initial code base to run with a maximum verbosity level of 3, there are some differences in sections of results that are written to the output file. The differences are as expected, given the coding in the output routines of RADTRAN. However, NRC should be

aware that any users who rely on level 4 output will be deprived of it when using NRC-RADTRAN. Remedying this would require code changes to RADTRAN 6.02.1 in order to re-enable the option.

4. The NRC-RADTRAN graphical user interface faithfully represents the input data when saving files or launching calculations. There are expected changes in line order and formatting, due to the fact that NRC-RADTRAN writes the inputs in a deterministic order and format regardless of whatever free ordering and formatting was present in the original input file. However, due to the way in which parsing routines are coded in RADTRAN, the input parameter formatting (e.g., “5” versus “5.0” or “5.00E0”) can actually have a slight numerical impact on the results (i.e., difference of 1 in the last significant digit in some output parameters). ERI views this as a shortcoming in RADTRAN rather than the GUI coding, as it would be unhelpful to either make unasked-for changes to numerical formatting or to warn the user that the input formatting that is typed-in may result in unpredictable changes in RADTRAN predictions. And, in any case, the differences in output values caused by this problem are very small.
5. The calculated number of casks sometimes differs between RADTRAN versions 6.02 and 6.02.1 occasionally differs in cases where no cask input data (i.e., ECONOMIC input block) have been entered. It is believed this may also be a symptom of the numerical differences observed in point #4, above. In any case, since no meaningful ECONOMIC data have been entered for these calculations, the cask output is also meaningless, and the discrepancy is therefore unimportant.
6. Because NRC-RADTRAN enforces consistency between the number of LOS probabilities entered on the PARAM line and the dimensionality of actual LOS data entered, there are some unimportant differences in the LOS sections of the output results (e.g., a table of meaningless zeroes or empty rows is now omitted).