Title: LANL Summer 2016 Report

Author(s): Mendoza, Paul Michael

Intended for: Internal report

Issued: 2016-08-29

Disclaimer:
Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

LANL Summer 2016 Report

Paul Mendoza

August 26, 2016

Abstract

The Monte Carlo N-Particle (MCNP) transport code developed at Los Alamos National Laboratory (LANL) utilizes nuclear cross-section data in a compact ENDF (ACE) format. The accuracy of MCNP calculations depend on the accuracy of nuclear ACE data tables, which depend on the accuracy of the original ENDF files.

There are some noticeable differences in ENDF files from one generation to the next [2], even among the more common fissile materials. As the next generation of ENDF files are being prepared, several software tools were developed to simulate a large number of benchmarks in MCNP (over 1000), collect data from these simulations, and visually represent the results.

1 Introduction

Each generation of ENDF files released by the US Cross Section Evaluation Working Group (CSEWG) combines experimental cross-section data with nuclear models to generate a more consistent and accurate dataset than the previous [3]. Although the first generation of ENDF files came out with particular focus on fissile materials in 1968, there are some noticeable differences in ENDF files from one generation to the next [2]. Whether due to the shear volume of data required for a “complete” nuclear cross section dataset, which would require information for numerous reactions for over 3000 isotopes, or due to the complexity of experiments required to generate cross section data for the ENDF files, there remains needed refinement in the area of nuclear data.

Monte Carlo N-Particle (MCNP) application data, which is data that is usable by the MCNP program, is produced when a version of the ENDF/B files are processed through a program called NJOY. NJOY recreates ENDF files by reconstructing resonances, doppler-broadening the cross sections, and coupling the cross section data with other desired data across a common energy grid [4].

The MCNP application data is used with MCNP to simulate systems based on geometry, materials, and physics specifications for phenomena to follow. A large number (over 4000) of critical experimental systems have been systematically catalogued by the International Criticality Safety Benchmark Evaluation Project (ICSBEP) for cross section library and code validation [1]. Input decks, which define systems for MCNP to simulate, have been prepared for 1152 of the ICSBEP experiments and are used to validate the MCNP application data through comparison with the real-world experimental results.

The present work focuses on automating the MCNP simulation of the ICSBEP MCNP input decks on Los Alamos National Laboratory’s (LANL’s) supercomputers for quicker data testing and analysis turnaround. Further, work included preparing a dataset with static information for these benchmarks, developing algorithms for consolidating the results from the simulations and the static information dataset, and finally, visually representing the data. Various applications were developed to accomplish the above four tasks, and are described in the next section. Cumulative results are presented in the following section, with closing remarks thereafter.

2 Applications Developed

2.1 benchmark_runner

benchmark_runner is an application developed that sets up files designed to pass jobs to a LANL supercomputer. The jobs are assumed to be ICSBEP benchmark simulations with MCNP. The algorithm for *benchmark_runner* first processes input information, then prepares to create moab scripts, and then creates scripts for benchmark execution.

There are 19 different inputs that *benchmark_runner* will process. These are for specifying which MCNP input decks should be executed and the conditions for their execution. There are also options for specifying whether or not the particular execution of *benchmark_runner* is a 'rerun', expected speed of MCNP calculations, and frequency for receiving updates about the calculation.

The inputs are then used to count the number of jobs that will be submitted to the supercomputer, determine the number of particle histories that MCNP will simulated, and count how many processors are on the current supercomputer. This information is used to calculate approximately how long each simulation will take, and use that information for requesting resources on the system. In this portion of the code, directories are set up so that output files are either organized in the same directory as the input files, or all grouped in another directory entirely.

This code will then create a moab script for each MCNP input file. A moab script is a bash script that will first, request resources from a LANL supercomputer, and second, once those resources are granted, execute a series of linux commands using those resources for the calculations. benchmark_runner will also create two additional programs to submit the moab scripts and to limit the number of job requests on the system.

2.1.1 check_benchmarks

Several secondary programs were developed to determine the success of *benchmark_runner*: *check_benchmarks*, *check_fatal*, *check_time*, and *check_basic*. These programs are all variations of check_benchmarks, which will analyze the moab and MCNP outputs (slurm and .o .r .s .m files) and summarizes the status of the simulations providing information about the number of successful jobs, and jobs with fatal errors or ran out of time of the moab client. This tool also prints out all the unique fatal errors encountered and the last completed cycle for benchmarks that ran out of time.

benchmark_runner will allocate all processors for each node resource granted to a single job. Figure 1 shows the execution time on the LANL supercomptuer 'mustang', which has 24 processors per node, for each of the MCNP input decks. Each symbol represents a different benchmark sweep, where the benchmarks are numbered on the x-axis, and the execution time is on the y-axis. Figure 1 shows that the different types of benchmarks will take different amounts of time on the same system. This is due to the differences in the geometric complexity for each system.

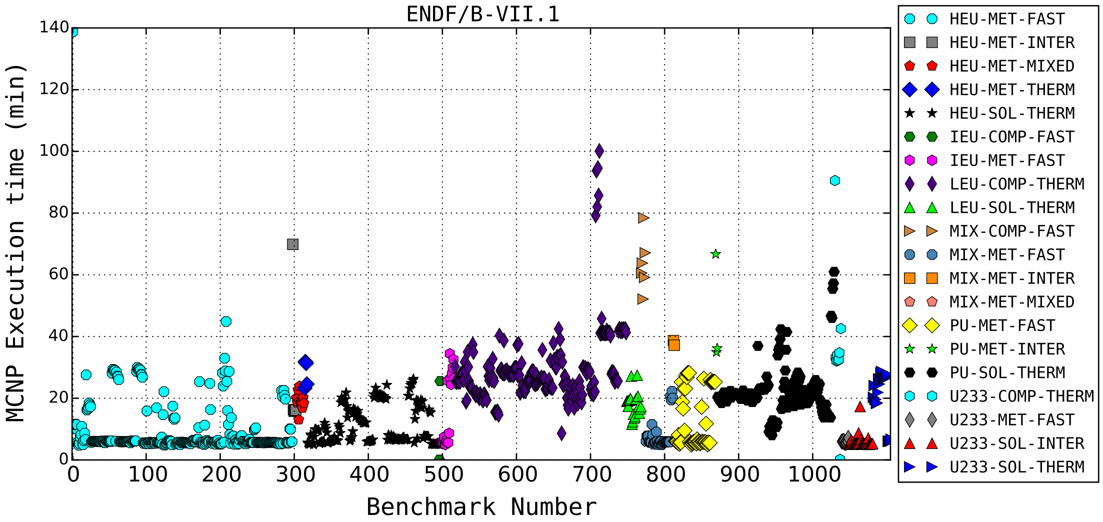

Figure 1: MCNP wall time for mustang.

Figure 2 is a histogram of the previous data, showing that the majority of the input decks required less than 20 minutes to execute. This is useful to know because the moab client will allocate resources based on quantity and the length of the time request. The larger the resource request, the longer it takes to receive the resources on the supercomputing system. Therefore, minimizing the request times for the majority of the MCNP calculations is optimal.

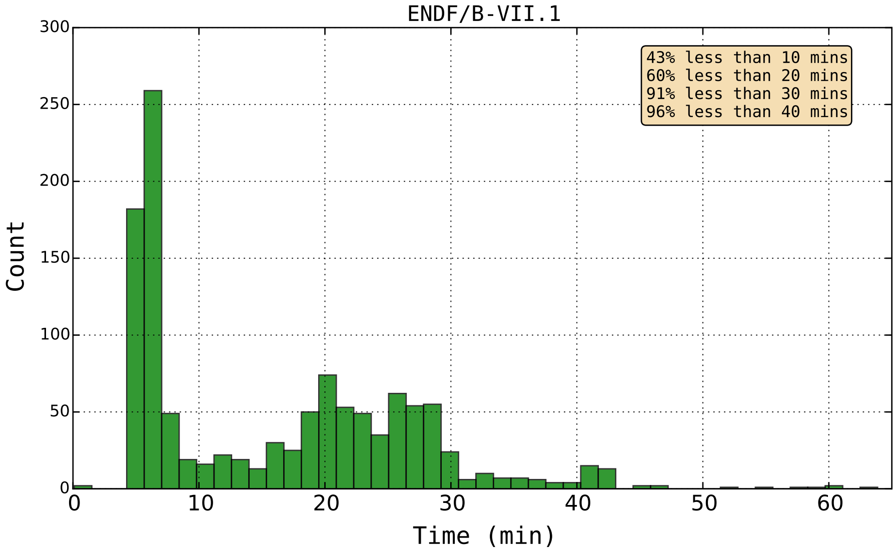

Figure 2: Histogram for time of execution on mustang.

2.2 format_static_bench_info

As mentioned above, ICSBEP has compiled criticality benchmark experiment data into a standardised format for over 4000 configurations [1]. This project utilized a subset of 1152 of these benchmarks, where MCNP input decks were created for each. The goal of this project is to be able to simulate these input decks with different cross section datasets. Certain data, such as the experimental k-eigenvalue with its associated error, geometry, number densities, and volumes, will remain constant from one cross section dataset simulation to the next. It was deemed advantageous to store this information in a single dataset.

A program called *format_static_bench_info* was used to accomplish this task. Information about the k-eigenvalues with error and geometry were provided in an excel spreadsheet. *format_static_bench_info* started with the spreadsheet, and added additional information about number densities and volumes. The most difficult portion of this task was ensuring that the naming convention for the benchmarks remained consistent so that the data could be properly linked together, both at this stage of the data manipulation and the next.

Another difficulty encountered during this portion of the project was organizing the number density and volume information. Across the 1152 input decks, there were a variety of isotopes that were used. Creating a dataset with a column for every isotope is possible, but was not attempted. Rather, important isotopes were selected in the categories of: fissile, fertile, moderator, and poison. The sum total of isotopes gathered was 37 isotopes as shown in Table 1. This list is not exhaustive, but *format_static_bench_info* was written in a way that this list could be very easily added to.

Table 1: List of isotopes collected in static benchmark dataset

Fissile	Fertile	Moderator	Poison
^{233}U	^{232}Th	^{1}H	^{10}B
^{235}U	^{232}U	^{2}H	^{11}B
^{239}Pu	^{234}U	^{3}H	^{3}He
^{241}Pu	^{236}U	^{nat}C	^{111}Cd
-	^{238}U	^{12}C	^{113}Cd
-	^{238}Pu	^{13}C	^{154}Gd
-	^{240}Pu	^{9}Be	^{155}Gd
-	^{241}Am	^{54}Fe	^{156}Gd
-	-	^{56}Fe	^{157}Gd
-	-	^{57}Fe	^{158}Gd
-	-	^{58}Fe	^{9}B
-	-	^{nat}Fe	^{6}Li
-	-	-	^{7}Li

Due to the fact that these isotopes may exist in any number of MCNP cells, an average volume weighted atom density was determined with Eq. 1, where ρ is the atom density, A

is the atom fraction for the particular isotope, and V is the volume. This calculation was possible if MCNP was able to calculate the volume for each of the cells. The term $V_{isotope}$ was determined with Eq. 2, where M is the mass fraction for the particular isotope.

$$\rho_{isotope} = \sum_{cell=1}^{N} A_{cell}^{isotope} \rho_{cell} \frac{V_{cell}}{V_{isotope}} \tag{1}$$

$$V_{isotope} = \sum_{cell=1}^{N} M^{isotope} V_{cell} \tag{2}$$

2.3 compile_bench_data

compile_bench_data was developed to combine the previous dataset with the simulation results from MCNP. The algorithm stores the static dataset in a matrix, gathers information while looping through the output MCNP files, and links the two datasets together in a csv file. Output parameters collected from MCNP were k-eigenvalues, percent of fissions caused by thermal, intermediate, and fast neutrons, and average energy/lethargy of neutrons causing fission.

2.4 plot_benchmarks

A program was developed to visually represent the above data. Given that plots are generated for each set of simulations, with potential slight modifications, a plotting program was developed that works by producing figures based on a provided input deck and data. The input deck specifies the type of plot and data, and the plotting program produces the plot. This approach was chosen so that the syntax of plotting is reduced to setting a given library of well defined variables.

3 Results

The above programs were used to execute the MCNP simulations for the benchmarks for ENDF/B-VII.1 and ENDF/B-VIII.0 β 1. The following results present a comparison between the two different cross section sets. Most of the following data utilized a ratio between the MCNP calculated k-eigenvalue (MCNP k_{eff}) and the experimentally determined k-eigenvalue (Experiment k_{eff}). If MCNP perfectly modeled the physics with exactly correct cross section data, and if the experimentally determined k-eigenvalues perfectly represented reality, then this ratio would be precisely one (given that MCNP simulated an infinite number of particles).

Figure 3 shows a histogram plot for both ENDF/B-VII.1 and ENDF/B-VIII.0 β 1 k-eigenvalue ratios. A Gaussian curve is shown both of the plots because MCNP utilizes random numbers for its simulations. If the above three asumptions were correct, then a Gaussian curve would be expected.

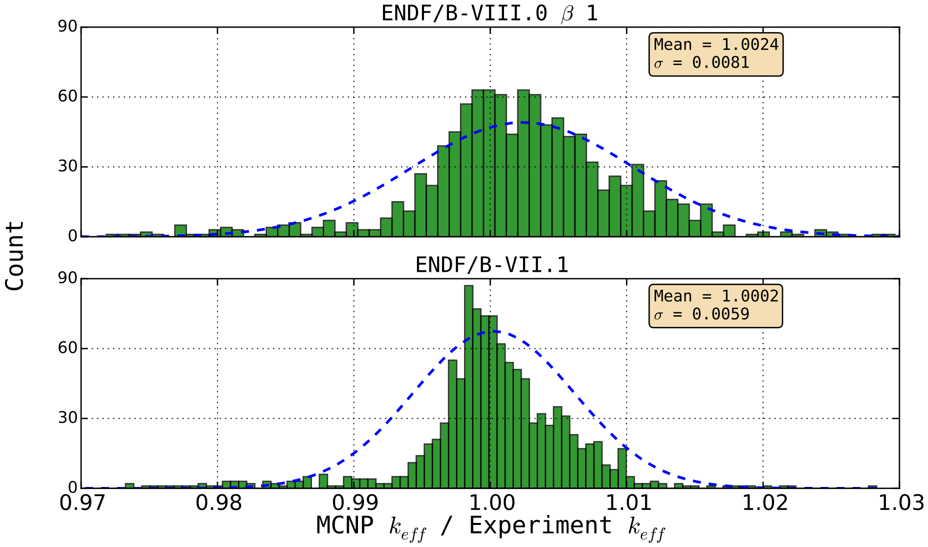

Figure 3: Histogram Comparison of Cross Section Libraries

The ENDF/B-VII.1 Gaussian fit is wider than the peak area because the peak is off center, and the distance to the edge data for the dataset is fairly large. Overall though, the narrower fit and mean value for the ENDF/B-VII.1 data suggests that ENDF/B-VIII.0 β 1 cross section data does not reproduce experimental k-eigenvalues as well.

In order to narrow in on one of the sources of these differences, the ratio between the two different MCNP calculations were plotted and shown in Figure 4 with the legend provided below in Figure 5 and the errors highlighted as background color. Figure 5 provides legend information for Figures 4, and 6.

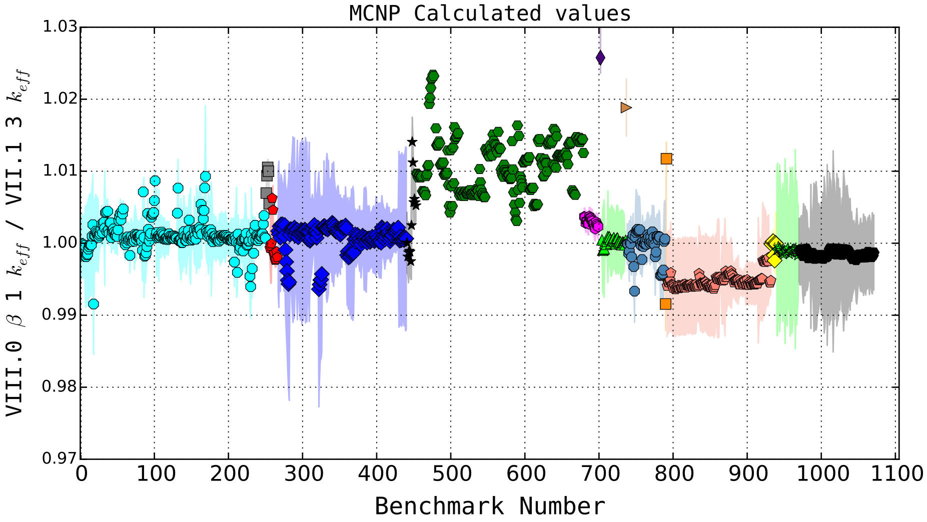

Figure 4: K-eigenvalue (MCNP) Ratio vs. Benchmark Number (legend shown in Figure 5)

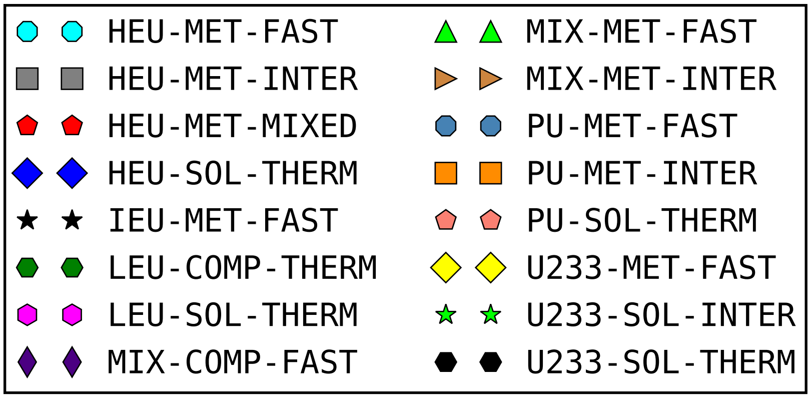

Figure 5: Legend for Figures 4,6

Two cases stand out in Figure 4 as being clearly different between the two different cross section sets, "LEU-COMP-THERM" and "PU-SOL-THERM" which stand for low enriched uranium-composite-thermal, and plutonium-solution-thermal, respectively. Other benchmark groups have large differences as well, but these two groups have a large number of cases with sizable differences from unity.

An examination of the benchmark groups shows that the "LEU-COMP-THERM" group is one of the only two groups with LEU, the other of which also has higher estimates for the k-eigenvalue. A possible reason for this is that the ^{238}U cross section changed between the two datasets. If this were the case, then larger quantities of ^{238}U in a system would lead to larger values of the MCNP/Experiment ratio because the k-eigenvalues for ENDF/B-VIII.0 β 1 increased from ENDF/B-VII.1.

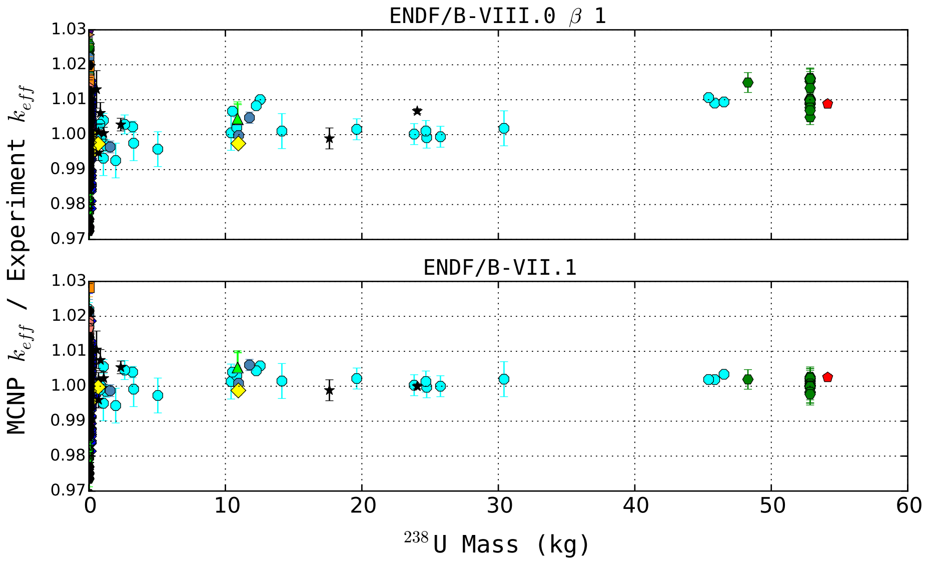

Figure 6: K-eigenvalue Ratio vs. ^{238}U Mass (legend shown in Figure 5)

Figure 6 is a plot of the MCNP/Experiment ratio plotted against the mass of ^{238}U for ENDF/B-VII.1 and ENDF/B-VIII.0 β 1. This figure shows that as the mass of ^{238}U in the system increases, MCNP/Experiment values for ENDF/B-VII.1 are relatively flat. ENDF/B-VIII.0 β 1 does seem to have an increase in the MCNP/Experiment ratio, but a closer examination shows that a few select cases of some groups increased, and that the "LEU-COMP-THERM" group increased, but that the majority of the points on the plot stayed relatively constant. There is probably more that is contributing to the increase in k-eigenvalue estimates.

3.1 Conclusions

Several software tools were developed to simulate a large number of benchmarks in MCNP (over 1000), collect data from these simulations, and visually represent the results. A simple analysis was also discussed to show some of the differences between two cross section datasets concluding that ENDF/B-VIII.0 β 1 has a better Gaussian shape, but that ENDF/B-VII.1 is closer to experimental k-eigenvalues. Also, that ^{238}U is not the main contribution for the differences in the "LEU-COMP-THERM" results between the two cross section datasets.

These programs reduce the simulation and analysis time from on the order to weeks to on the order of days. The *benchmark_runner* program has the capability to parse out benchmark jobs so that if a node were avaiable for each job, the simulation time would be reduced to around 4 hours, and data collection with corresponding visualization would take less than an hour.

Additional work for this project includes gathering more information from the MCNP output files. The information listed above that was collected and organized is a small fraction of the total data produced by these simulations. With further focus on collecting data, more avenues for understanding the nuclear cross section data become available. Other areas for future work would be in the area of fitting mathematical models to sets of the data and spending more time analyzing trends.

References

[1] J Blair Briggs, Virginia F Dean, and Lori Scott. The international criticality safety benchmark evaluation project (icsbep). Experimental Needs In Criticality Safety, page 109, 2003.

[2] M.B. Chadwick, P. Obložinskỳ, et al. Next generation evaluated nuclear data library for nuclear science and technology. Nuclear data sheets, 2006.

[3] RB Kidman. Endf/bv, lib-v, and the csewg benchmarks. Technical report, LA-8950-MS, Los Alamos National Laboratory, Los Alamos, NM, 1981.

[4] RE MacFarlane, RJ Barrett, DW Muir, and RM Boicourt. Njoy nuclear data processing system: user's manual. Technical report, Los Alamos Scientific Lab., 1978.