LA-UR-12-21729
Approved for public release; distribution is unlimited.

Title: Verification of MCNP5-1.60 and MCNP6-Beta2 for Criticality Safety Applications

Author(s): Kiedrowski, Brian C.
Brown, Forrest B.
Bull, Jeffrey S.

Intended for: 2012 American Nuclear Society Winter Meeting, 2012-11-11/2012-11-15 (San Diego, California, United States)

Verification of MCNP5-1.60 and MCNP6-Beta2 for Criticality Safety Applications

Brian C. Kiedrowski, Forrest B. Brown, and Jeffrey S. Bull

Los Alamos National Laboratory, PO Box 1663, Los Alamos, NM, 87544, bckiedro@lanl.gov

INTRODUCTION

The MCNP Monte Carlo code has been used for high-fidelity analyses of criticality safety problems since the 1970s. Periodically, with support of the US Department of Energy Nuclear Criticality Safety Program (NCSP), the MCNP development team runs criticality calculations using the latest versions of MCNP with various compilers, parallel options, and on different platforms to ensure that the code is still capable of producing accurate keff results.

Over the last several years, Los Alamos National Laboratory (LANL) has devoted much effort to version 6 of MCNP, or MCNP6 [1]. Most of this effort has been devoted to merging the MCNP5-1.60 [2,3] and MCNPX code bases. Very few of the MCNPX features are relevant to criticality safety practitioners, except the depletion or burnup feature in MCNPX for isotopic inventories that may be useful for burnup credit. Also new to MCNP6 is the nuclear data sensitivity capability, which is not available in any previous version of MCNP5 or MCNPX.

Because of the large amount of structural change to the core transport routines between MCNP5-1.60 (the latest verified and validated version [4]) and MCNP6, users have a legitimate concern that criticality results may be negatively impacted. Results of LANL calculations show that the results of MCNP6-Beta2 (the most recent version of MCNP6 at the time of this writing) agree very well with MCNP5-1.60, often matching results exactly despite different platforms, parallel options, and compilers.

DESCRIPTION OF THE VERIFCATION TESTS

Verification was performed by comparing results of MCNP6-Beta2 with MCNP5-1.60 (a version that has been verified and validated previously). To do this, three Validation Suites that are packaged with MCNP were used:

VALIDATION_CRITICALITY - The "Criticality Validation Suite" [5] consisting of 31 problems from the *International Handbook of Evaluated Criticality Benchmark Experiments* [6], using the ENDF/B-VII.0 nuclear data libraries

VALIDATION_CRIT_EXPANDED - The "Expanded Criticality Validation Suite" [7] consisting of 119 problems from the *International Handbook of Evaluated Criticality Benchmark Experiments*, using the ENDF/B-VII.0 nuclear data libraries,

CRIT_LANL_SBCS – A suite of 194 ICSBEP criticality benchmark problems from the LANL SB-CS Group used to verify and validate MCNP for their work, using ENDF/B-VI nuclear data libraries.

Both versions of MCNP were compiled with versions 10, 11, and 12 of the Intel Fortran-90 Compiler. Compilers are very complex programs, and are not bug free. Additionally, different versions may perform floating-point arithmetic slightly differently, leading to different answers because of numerical round off – this can lead to a divergence in the random number sequence in a criticality calculation because the random walk taken during a cycle is dependent on the random walks in the previous cycles.

The MCNP development team tries to choose options for different Fortran-90 compilers and computer platforms that are as consistent as possible for building MCNP5. Nevertheless, computer round-off differences will occur with different compilers/hardware. Round-off differences are not considered errors. Careful examination of these differences is necessary in the verification process to ensure that these differences are solely a result of round off, and not errors in MCNP coding or compilers

For all compilations, the `-O1` optimization flag was used. Past testing typically showed only small gains in performance with higher optimization levels, at the expense of tremendous complications in verification because of small round-off differences.

The compilers used on the Mac OS X (Mac Pro, 2 quad-core Xeon, OS X 10.6.8) were Intel-10.1.008 (32-bit), Intel-11.1.088 (32-bit), and Intel-12.0.0 (64-bit). For Windows (Windows PC, 2 hex-core i7, Windows7 OS), the Intel-12.03.175 compiler with 32- and 64-bit addressing was used. Linux calculations were run on the Turning cluster at LANL (Quad-Core AMD Opteron model 8354 at 2.2 GHz or the model 8356 at 2.3 GHz) with the Intel-11.1.072 with 64-bit addressing.

All of the testing for Mac or Windows was done in parallel, using either OpenMP (OMP) threading with 8-12 CPU cores. On Linux, the Message Passing Interface (MPI) with 128 processes is used. This difference is to capture both flavors of parallelism.

Table I. Differences with Different Fortran Compilers, Mac OS

	MCNP5-1.60		MCNP6-Beta2					
	Intel 12		Intel 10		Intel 11		Intel 12	
	keff	std	deltak	std	deltak	std	deltak	std
TT2C11	1.0008	(7)	0.0010	(10)	0.0000	(9)	0.0000	(9)
ZEUS2	0.9972	(7)	0.0002	(9)	0.0000	(9)	0.0000	(9)
ZEBR8H	1.0196	(5)	-0.0001	(7)	0.0000	(7)	0.0000	(7)
HISHPG	1.0118	(5)	0.0004	(7)	0.0000	(8)	0.0000	(8)

Table II. Expanded Criticality Validation Suite Differences, Linux

	MCNP5-1.60		MCNP6-Beta2	
	keff	std	deltak	std
heu-met-fast-019-case-2	1.0073	(2)	0.0001	(4)
heu-met-fast-011	0.9989	(2)	-0.0005	(4)
ieu-comp-therm-002-case-3	1.0044	(2)	-0.0002	(4)

RESULTS OF THE VERIFICATION TESTS

The following comparisons were made using VALIDATION_CRITICALITY: different Fortran-90 compiler versions on MacOS for MCNP5-1.60 versus MCNP6-Beta2, and both code versions on each of the different platforms. Comparisons of MCNP5-1.60 and MCNP6-Beta2 for VALIDATION_CRIT_EXPANDED were done with Linux. Similar comparisons were done with CRIT_LANL_SBCS, but with Mac OS X and against a 2003 version, MCNP5-1.25, as well.

Results are shown here for only those that disagree. A larger report with all the results that was submitted to the US DOE NCSP is available online at the MCNP website [8].

VALIDATION_CRITICALITY: Different Fortran Compilers

MCNP6-Beta2 and MCNP5-1.60 have exact agreement for 27 out of 31 of the results, despite different compiler versions (all these calculations use ENDF/B-VII.0 nuclear data). Table I shows the results and differences for the four cases with disagreement. The differences are within 1-σ (one standard deviation of the Monte Carlo statistical uncertainty) and indicate computer round off (most likely from reordering of arithmetic because of compiler optimization), not errors in either MCNP or the Intel compilers. Note that: (1) Using the Intel-11 or Intel-12 compilers, results for MCNP5-1.60 and MCNP6-Beta-2 match exactly for all 31 cases. (2) MCNP6-Beta-2 compiled with the Intel-10 compiler shows round-off differences for the same 4 cases noted previously, and in fact matches MCNP5-1.60 compiled with Intel-10. (3) MCNP6-Beta-2 compiled with the Intel-12 compiler in 64-bit addressing mode is roughly 20% slower than MCNP5-1.60.

While not shown here (see full report in Ref. 8), the differences observed are fairly consistent with, although not exactly, the results when comparing different compilers between MCNP5-1.51 and MCNP5-1.60.

VALIDATION_CRITICALITY: Different Platforms

The VALIDATION-CRITICALITY suite was run on MacOS, Windows, and Linux with the same compiler version with ENDF/B-VII.0 nuclear data. Results for all 31 cases match exactly. This shows that MCNP5-1.60 and MCNP6-Beta2 is stable between platforms.

VALIDATION_CRIT_EXPANDED: Different Code Versions

Results of keff from MCNP5-1.60 and MCNP6-Beta2 were obtained for the 119 problems in the Expanded Criticality Validation Suite using ENDF/B-VII.0. Both sets of calculations were run on a Linux cluster using 128 MPI processes and the same Intel-11 compiler.

All but 3 of 119 the benchmarks agree exactly between the two versions. The results that disagreed are given in Table II. These are minor round-off differences between MCNP5 and MCNP6 results. Two show round-off differences less than 1-σ, and the other case shows round off of just over 1-σ. These differences are judged to be insignificant, and simply the normal-round off differences between the two codes that are expected when running very many calculations.

CRIT_LANL_SBCS: Different Code Versions

Results were obtained with MCNP5-1.25, MCNP5-1.60 and MCNP6-Beta2 for 194 ICSBEP problems in the LANL SBCS Group criticality validation suite, using

ENDF/B-VI nuclear data. MCNP5-1.25 is from 2003 and compiled with the Intel-9.

Table III shows the results of the 54 benchmarks that disagree from MCNP5-1.25. 43 of those cases agree within 1-σ and 11 agree with differences between 1-σ and 2-σ. The other 140 cases all matched within four decimal places of precision.

This agreement is excellent, considering that nearly a decade of MCNP5 development and bug fixes have occurred, and three generations of Fortran-90 compilers separate the old MCNP5 results from the new ones. All of the 194 cases show differences of less than 2-σ. (Note: 3 of the 11 problems showing differences between 1- and 2-σ from old results actually had problem input errors that were corrected for the MCNP5-1.60 and MCNP6-Beta-2 calculations.)

Comparing MCNP5-1.60 and MCNP6-Beta-2 results for the 194 cases, 188 cases match, three show differences less than 1-σ, and three more show differences between 1- and 2-σ. These differences are judged to be insignificant, and simply the normal round-off differences between the two codes that are expected when running very many calculations

Additional testing was performed using the Intel-10.1 Fortran-90 compiler for MCNP5-1.60 and MCNP6-Beta-2. The results are not shown, but they match exactly for 192 of the 194 cases, and differ within 1-σ for the other two.

SUMMARY AND RECOMMENDATIONS

The general conclusions from this testing are:

- Both MCNP5-1.60 and MCNP6-Beta-2 perform correctly for criticality safety applications.
- While small differences were noted for a few cases, these are strictly from computer round off and are not a concern for verification/validation.
- MCNP5-1.60 and MCNP6-Beta-2 yield the same results on different computer platforms – Mac OS X, Linux, and Windows – for criticality safety applications.
- MCNP5-1.60 and MCNP6-Beta-2 yield the same results using OMP threading and/or MPI message-passing parallelism.
- Using the Intel-12 compiler and 64-bit addressing produces roughly a 20% speedup in the MCNP executables compared to using older compilers.
- MCNP6-Beta-2 runs roughly 20% slower than MCNP5-1.60.

As a result of this testing, the MCNP development team recommends that all future development for MCNP be accomplished using the latest Fortran-90 compiler, Intel-12, rather than older versions of the compiler. Using the Intel-12 Fortran-90 compiler with 64-bit addressing permits the solution of very large problems that could not be run with older compilers and 32-bit addressing (where array sizes were limited to less than 2 GB), and also provides a speedup of roughly 20% in code execution.

Criticality safety analysts should consider testing MCNP6-Beta-2 on their particular problems and validation suites, to prepare for the eventual migration from MCNP5 to MCNP6. It is expected that this migration should be accomplished within the next 1-2 years. Currently, no further development of MCNP5 is planned, and all future MCNP improvements will be targeted to MCNP6.

ACKNOWLEDGMENTS

Funding for this work was provided by the US DOE Nuclear Criticality Safety Program.

REFERENCES

1. MCNP6 DEVELOPMENT TEAM, “Initial MCNP6 Release Overview”, *J. Nucl. Technol.*, Accepted for Publication, also LA-UR-11-07082, Los Alamos National Laboratory (2011).
2. X-5 MONTE CARLO TEAM, “MCNP – A General N-Particle Transport Code, Version 5 – Volume I: Overview and Theory”, LA-UR-03-1987, Los Alamos National Laboratory (April, 2003).
3. F.B. BROWN, B.C. KIEDROWSKI, J.S. BULL, "MCNP5-1.60 Release Notes", LA-UR-10-06235, Los Alamos National Laboratory (2010).
4. F.B. BROWN, B.C. KIEDROWSKI, J.S. BULL, M. GONZALES, N. GIBSON, "Verification of MCNP5-1.60", LA-UR-10-05611, Los Alamos National Laboratory (2010).
5. R.D. MOSTELLER, "Validation Suites for MCNP," *Proc. of the American Nuclear Society Radiation Protection and Shielding Division 12th Biennial Topical Meeting*, Santa Fe, New Mexico (April 2002).
6. J.B. BRIGGS (Ed.), International Handbook of Evaluated Criticality Safety Benchmark Experiments, NEA/NSC/DOC(95)03, OECD Nuclear Energy Agency (2008).
7. R.D. MOSTELLER, “An Expanded Criticality Validation Suite for MCNP”, LA-UR-10-06230 (2010).
8. F.B. BROWN, B.C. KIEDROWSKI, J.S. BULL, "Verification of MCNP5-1.60 and MCNP6-Beta-2 for Criticality Safety Applications", LA-UR-12-21041 (2012) (Online at: http://mcnp.lanl.gov).

Table III. LANL SBCS Validation Suite Differences, Mac OS

	MCNP5-1.25		MCNP5-1.60		MCNP6-Beta2	
	keff	std	deltak	std	deltak	std
heu-comp-therm-002-001	1.0082	(9)	0.0013	(13)	0.0006	(14)
heu-comp-therm-002-011	1.0123	(9)	0.0000	(13)	-0.0005	(13)
heu-comp-therm-002-015	1.0171	(7)	-0.0001	(11)	-0.0001	(11)
heu-comp-therm-002-023	1.0128	(8)	0.0005	(12)	0.0006	(12)
heu-comp-therm-002-024	1.0131	(8)	-0.0001	(11)	-0.0010	(11)
heu-comp-therm-002-025	1.0113	(8)	-0.0006	(12)	-0.0006	(12)
heu-met-fast-001-001	0.9952	(6)	0.0015	(9)	0.0015	(9)
heu-met-fast-002-001	1.0002	(7)	-0.0002	(9)	-0.0002	(9)
heu-met-fast-002-002	1.0014	(6)	-0.0005	(9)	-0.0005	(9)
heu-met-fast-002-003	0.9984	(6)	0.0012	(9)	0.0012	(9)
heu-met-fast-002-004	0.9980	(6)	0.0010	(8)	0.0010	(8)
heu-met-fast-002-005	1.0006	(7)	-0.0010	(9)	-0.0010	(9)
heu-met-fast-002-006	1.0010	(6)	-0.0001	(9)	-0.0001	(9)
heu-met-fast-003-001	0.9925	(6)	-0.0004	(8)	-0.0004	(8)
heu-met-fast-003-002	0.9895	(5)	0.0007	(7)	0.0007	(7)
heu-met-fast-003-003	0.9974	(6)	-0.0005	(8)	-0.0005	(8)
heu-met-fast-003-004	0.9952	(6)	0.0009	(9)	0.0009	(9)
heu-met-fast-003-005	0.9999	(6)	-0.0007	(8)	-0.0007	(8)
heu-met-fast-003-006	1.0002	(7)	-0.0002	(9)	-0.0002	(9)
heu-met-fast-003-007	1.0009	(6)	-0.0002	(9)	-0.0002	(9)
heu-met-fast-003-009	1.0044	(5)	0.0008	(9)	0.0008	(9)
heu-met-fast-003-010	1.0089	(6)	-0.0009	(8)	-0.0009	(8)
heu-met-fast-003-011	1.0135	(6)	-0.0011	(8)	-0.0011	(8)
heu-met-fast-003-012	1.0053	(5)	-0.0004	(8)	-0.0004	(8)
heu-met-fast-004-001	0.9886	(7)	0.0002	(10)	0.0002	(10)
heu-met-fast-007-035	0.9920	(8)	-0.0000	(12)	-0.0014	(11)
heu-met-fast-007-039	1.0038	(7)	-0.0000	(10)	-0.0013	(10)
heu-met-fast-028-001	1.0016	(6)	-0.0003	(9)	-0.0003	(9)
heu-met-fast-030-001	1.0046	(7)	-0.0001	(9)	-0.0001	(9)
heu-met-fast-067-002	1.0075	(6)	0.0003	(8)	0.0003	(8)
heu-sol-therm-001-001	0.9987	(10)	-0.0003	(15)	-0.0003	(15)
mix-met-fast-001-001	0.9966	(6)	-0.0005	(8)	-0.0005	(8)
pu-met-fast-005-001	1.0065	(7)	0.0008	(9)	0.0008	(9)
pu-met-fast-006-001	1.0019	(7)	0.0005	(9)	0.0005	(9)
pu-met-fast-008-001	1.0060	(6)	0.0005	(8)	0.0005	(8)
pu-met-fast-009-001	1.0027	(6)	-0.0010	(8)	-0.0010	(8)
pu-met-fast-010-001	0.9997	(6)	-0.0002	(8)	-0.0002	(8)
pu-met-fast-011-001	0.9958	(7)	-0.0002	(10)	-0.0002	(10)
pu-met-fast-012-001	1.0044	(7)	0.0006	(10)	0.0006	(10)
pu-met-fast-013-001	1.0064	(6)	-0.0005	(9)	-0.0005	(9)
pu-met-fast-014-001	1.0044	(6)	0.0003	(8)	0.0003	(8)
pu-met-fast-015-001	0.9990	(6)	0.0009	(9)	0.0009	(9)
pu-met-fast-018-001	0.9986	(13)	0.0017	(14)	0.0017	(14)
pu-met-fast-021-001	1.0030	(7)	0.0005	(9)	0.0005	(9)
pu-met-fast-021-002	0.9914	(6)	-0.0007	(8)	-0.0007	(8)
pu-met-fast-045-001	0.9983	(6)	-0.0004	(9)	-0.0004	(9)
pu-met-fast-045-002	1.0021	(6)	0.0009	(9)	0.0009	(9)
pu-met-fast-045-003	1.0017	(6)	0.0002	(8)	0.0002	(8)
pu-met-fast-045-004	1.0000	(6)	0.0007	(9)	0.0007	(9)
pu-met-fast-045-005	1.0065	(6)	-0.0007	(8)	-0.0007	(8)
pu-met-fast-045-006	1.0011	(7)	0.0009	(9)	0.0009	(9)
pu-met-fast-045-007	1.0023	(6)	-0.0004	(8)	-0.0004	(8)
pu-sol-therm-004-003	1.0009	(8)	0.0012	(11)	0.0012	(11)
pu-sol-therm-004-006	1.0009	(8)	0.0002	(11)	0.0002	(11)