LA-UR-00-2532

*Approved for public release;
distribution is unlimited.*

Title: MCNP SOFTWARE QUALITY: THEN AND NOW

Author(s): Gregg C. Giesler, CIC-12

Submitted to: Proceedings of 10th International Conference on
Software Quality
16-18 October 2000
New Orleans, Louisiana

MCNP SOFTWARE QUALITY: THEN AND NOW

Gregg C. Giesler
Los Alamos National Laboratory
Los Alamos, NM USA

ABSTRACT

MCNP is the Monte Carlo N-Particle radiation transport code whose history dates back more than half a century to the early days of computing. From a simple beginning, its uses have grown to include fields such as criticality safety, radiation shielding, oil well logging, and medical imaging and diagnostics and an international user community of over 3000 users. This large user community could only happen by the maintainance of sofware quality throughout its history. This paper will describe how the quality was maintained in the past, how the process is being improved today, and directions for future efforts.

Introduction

Think back to the early days of electronic digital computing, back to the days before personal computers, back before whirling disk drives, back before spinning tape reels, back before transistors, back to the days when computers were made of vacuum tubes and relays and filled huge rooms, back to when they were attended by armies of workers, back to the days when code was entered on plug boards. Back in the distant past of computing, over half a century ago, the first runs of what is now MCNP were made.

The Los Alamos National Laboratory (LANL) Monte Carlo N-Particle radiation transport code, MCNP, has a history that dates back more than fifty years. Since its origin in the late 1940s at LANL, its use has grown so that it is now a highly respected code that is used throughout the world. Not only is it used for radiation protection and shielding applications, it also has many uses in medical and other industries. In order to support this growing user community and variety of applications, new features and enhancements are continually being added.

How does a code receive this broad acceptance? Only by maintaining a high standard of quality. In this paper, I will first describe the origins and history of MCNP and then include a brief description of the Monte Carlo technique. This will be followed by a broad description of uses of MCNP. Then, I will described the recent past software quality procedures and the procedures and practices currently being implemented. Finally, in the discussion, I will describe some of the future directions planned for this code.

MCNP History

Perhaps the earliest documented use of random sampling to solve a mathematical problem was that of Compte de Buffon in 1772. In the following next two centuries, this technique had a number of other uses. In the 1930s, Enrico Fermi used it to solve problems in neutron physics, although he never published his results. In Los Alamos during World War II, Fermi along with Stan Ulam, John von Neumann, Nicholas Metropolis, and others discussed the application of this statistical sampling technique to the problems they were working on. Ulam pointed out the use of electromechanical computers to overcome the long and tedious nature of the calculations, and Metropolis named this previously unnamed technique “Monte Carlo” after Ulam’s uncle who borrowed money from relatives because he "just had to go to Monte Carlo” (the gambling casino).

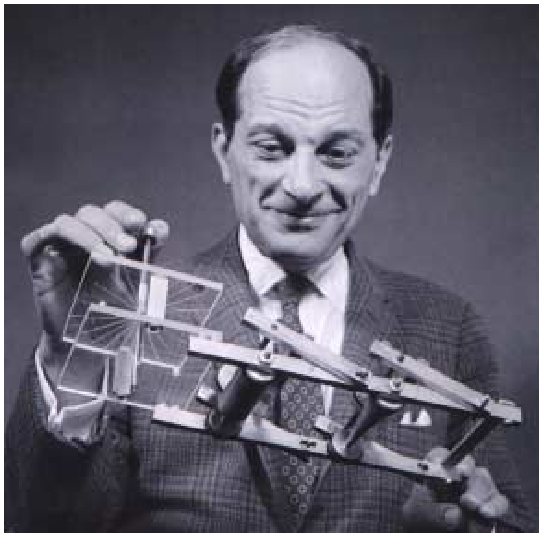


Figure 1. Stan Ulam Holding FERMIAC

On March 11, 1947, John von Neumann sent a letter (Richtmyer, 1947) to the Theoretical Division leader proposing the use of this technique on ENIAC to solve neutron diffusion and multiplication problems. This was the first proposal to use the Monte Carlo technique on an electronic digital computer. Also in 1947, Enrico Fermi had FERMIAC (Figure 1), a mechanical analog computer, programmed to run Monte Carlo problems. In 1948, the first runs on a digital computer took place on ENIAC (Figure 2). In the late 1940s and early 1950s, many papers were written describing the Monte Carlo method and its use in solving problems in radiation and particle transport and other areas. The first open Monte Carlo conference was held at UCLA in the summer of 1949. Many of those methods are still in use today including the random number generation method used in MCNP.

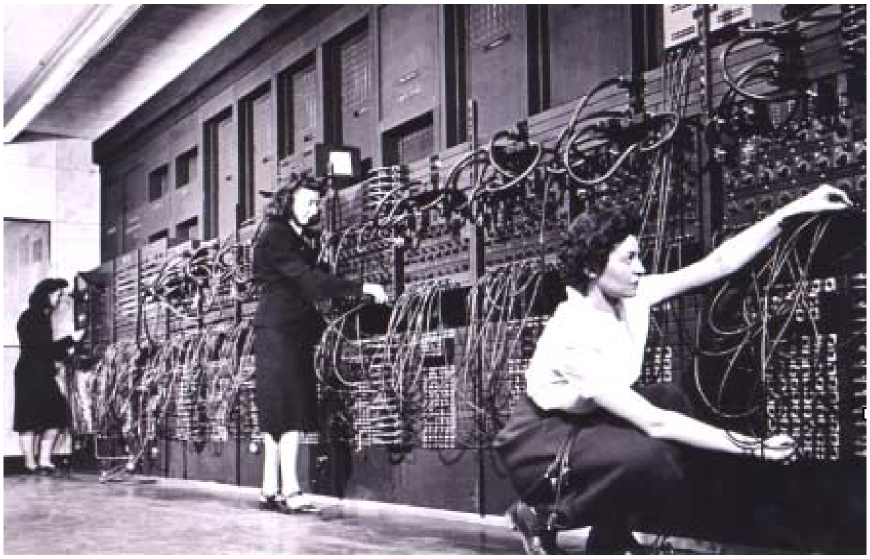
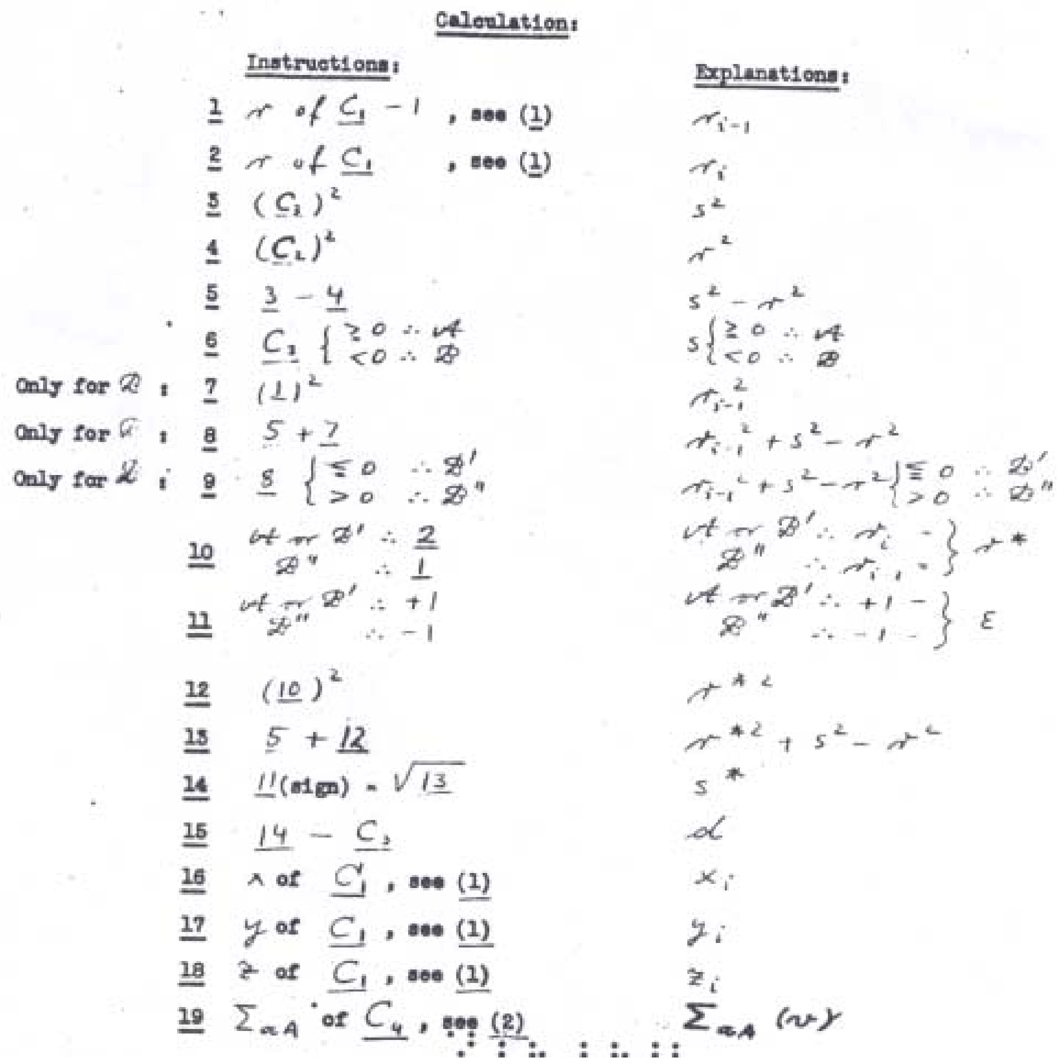


Figure 2. ENIAC



Calculation:

		Instructions:	Explanations:
	$\underline{1}$	r of $\underline{C_1} - 1$, see ($\underline{1}$)	r_{i-1}
	$\underline{2}$	r of $\underline{C_1}$, see ($\underline{1}$)	r_i
	$\underline{3}$	$(\underline{C_3})^2$	s^2
	$\underline{4}$	$(\underline{C_2})^2$	r^2
	$\underline{5}$	$\underline{3} - \underline{4}$	$s^2 - r^2$
	$\underline{6}$	$\underline{C_3}\begin{cases} \geq 0 \therefore \mathcal{A} \\ < 0 \therefore \mathcal{B} \end{cases}$	$s\begin{cases} \geq 0 \therefore \mathcal{A} \\ < 0 \therefore \mathcal{B} \end{cases}$
Only for $\mathcal{B}$:	$\underline{7}$	$(\underline{1})^2$	r_{i-1}^2
Only for $\mathcal{B}$:	$\underline{8}$	$\underline{5} + \underline{7}$	$r_{i-1}^2 + s^2 - r^2$
Only for $\mathcal{B}$:	$\underline{9}$	$\underline{8}\begin{cases} \leq 0 \therefore \mathcal{B}' \\ > 0 \therefore \mathcal{B}'' \end{cases}$	$r_{i-1}^2 + s^2 - r^2\begin{cases} \leq 0 \therefore \mathcal{B}' \\ > 0 \therefore \mathcal{B}'' \end{cases}$
	$\underline{10}$	$\mathcal{A}$ or $\mathcal{B}' \therefore \underline{2}$ $\mathcal{B}'' \therefore \underline{1}$	$\left.\begin{matrix} \mathcal{A} \text{ or } \mathcal{B}' \therefore r_i - \\ \mathcal{B}'' \therefore r_{i-1} - \end{matrix}\right\} r^*$
	$\underline{11}$	$\mathcal{A}$ or $\mathcal{B}' \therefore +1$ $\mathcal{B}'' \therefore -1$	$\left.\begin{matrix} \mathcal{A} \text{ or } \mathcal{B}' \therefore +1 - \\ \mathcal{B}'' \therefore -1 - \end{matrix}\right\} \varepsilon$
	$\underline{12}$	$(\underline{10})^2$	r^{*2}
	$\underline{13}$	$\underline{5} + \underline{12}$	$r^{*2} + s^2 - r^2$
	$\underline{14}$	$\underline{11}$(sign) $\cdot \sqrt{\underline{13}}$	s^*
	$\underline{15}$	$\underline{14} - \underline{C_3}$	d
	$\underline{16}$	x of $\underline{C_1}$, see $\underline{(1)}$	x_i
	$\underline{17}$	y of $\underline{C_1}$, see $\underline{(1)}$	y_i
	$\underline{18}$	z of $\underline{C_1}$, see $\underline{(1)}$	z_i
	$\underline{19}$	Σ_{aA} of $\underline{C_4}$, see $\underline{(2)}$	$\Sigma_{aA}(v)$

Figure 3. First Monte Carlo Code

At Los Alamos, Monte Carlo computer codes developed along with computers. The first Monte Carlo code was a simple 81-step program written on computing sheets attached to von Neumann's letter. Figure 3 shows a copy of the first sheet of the code. As computers evolved, the follow-on codes were written in machine language, and each code solved a specific problem. In the early 1960s, the standardization of programming languages such as FORTRAN allowed the development of more generalized codes.

The first Los Alamos general-purpose particle transport Monte Carlo code was MCS (Johnston, 1963). Actually, MCS was a set of codes. They included MCA (or MCB) to process the input and set up the tapes for MCS, MCS to do the calculations, and MCR to multiply the MCS results by the collision probabilities to get the final results. MCH with MCI to process the input was a simplified version of MCS and MCR. Scientists could now solve modest problems without having to program or do the mathematical analysis themselves.

In 1965, MCS was followed by MCN (Cashwell, 1972) which could solve problems of neutrons interacting with matter in a three-dimensional geometry and used physics data stored in separate libraries. In 1973, MCN was merged with MCG (Cashwell, 1973), a Monte Carlo Gamma code that treated higher energy photons, to form MCNG. MCNP, Monte Carlo Neutron Photon, was formed in 1977 by merging MCNG with MCP (Cashwell, 1973), a Monte Carlo Photon code with detailed photon physics down to 1 keV. Since then, MCNP has been changed to stand for Monte Carlo N-Particle.

The major release history of MCNP is shown in Figure 4. It includes major features added to MCNP over the years. It is a major tribute to the quality of MCNP that versions could stand for three years without requiring bug correction releases. A major release represents major changes in the code. MCNP3 was the first version written entirely in ANSI standard FORTRAN. It was also the first version released internationally through what is now the Radiation Safety Information Computational Center (RSICC) at Oak Ridge, TN. MCNP4 included the first UNIX version of the code.

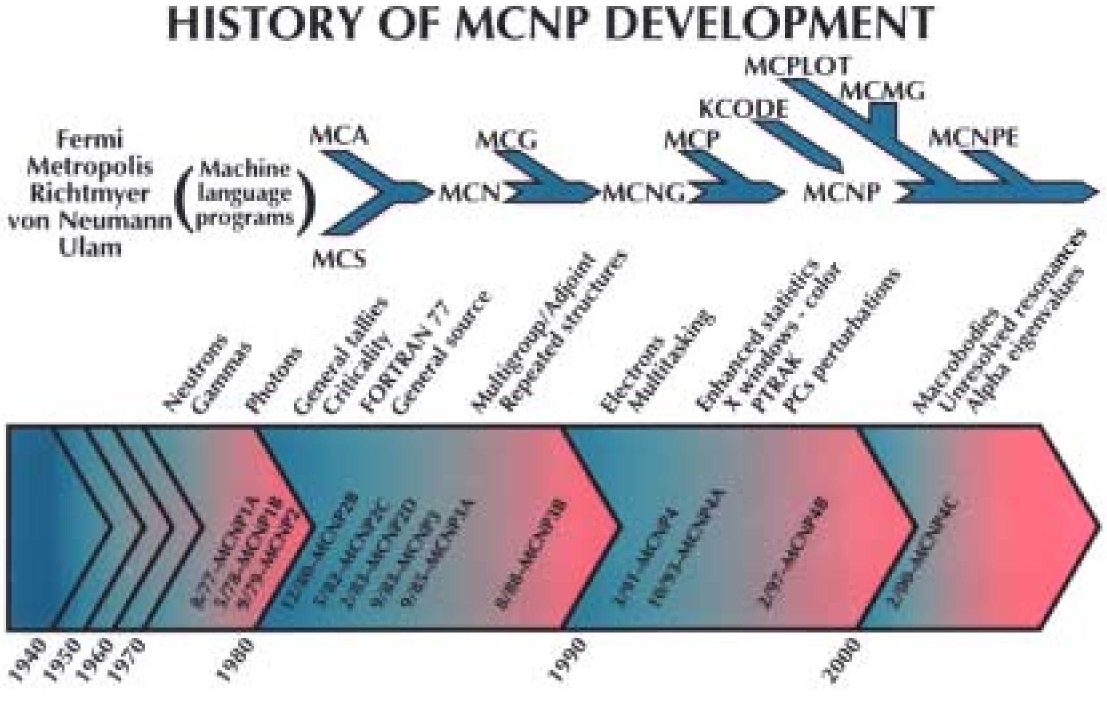


Figure 4. MCNP Development History

For more than half a century, almost the entire history of digital computing, MCNP and its predecessors have been supported on a variety of assemblers, compilers, and hardware platforms. A timeline showing the history of MCNP and the “super” computers installed at LANL are shown in Figure 5. With the international use of MCNP, this support covers a large range of processors and capabilities. Figure 6 lists the environments supported by the current version, MCNP4C. Within these, the same problem can be run, in a massively parallel configuration on all 6144 processors of Blue Mountain or on your desktop PC. However, the speed at which the problem runs will vary greatly.

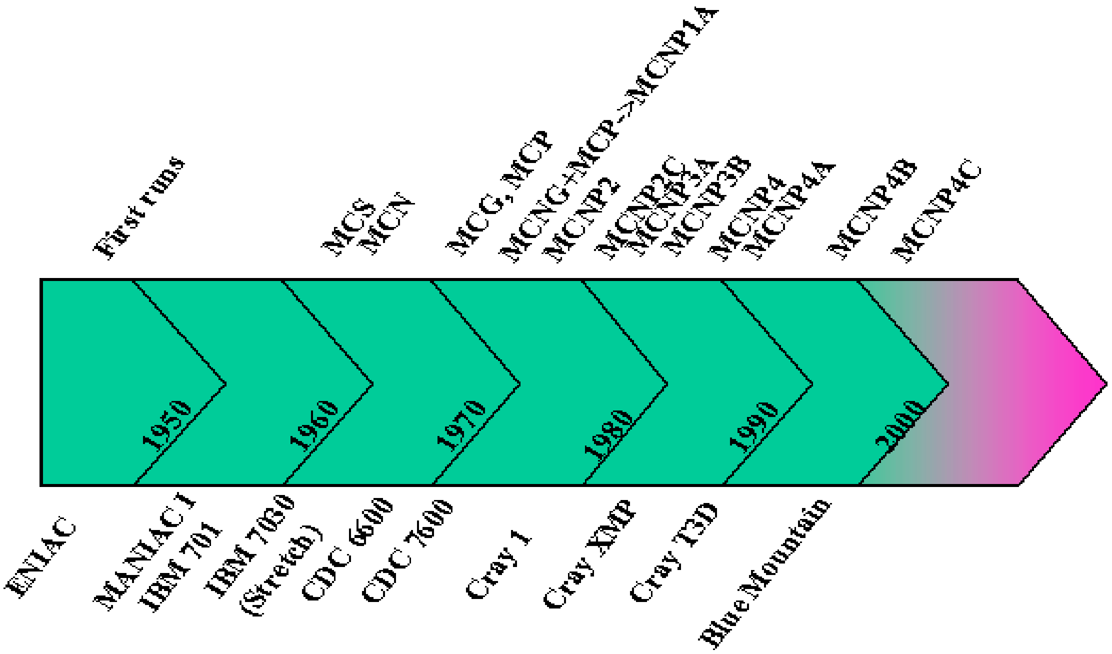
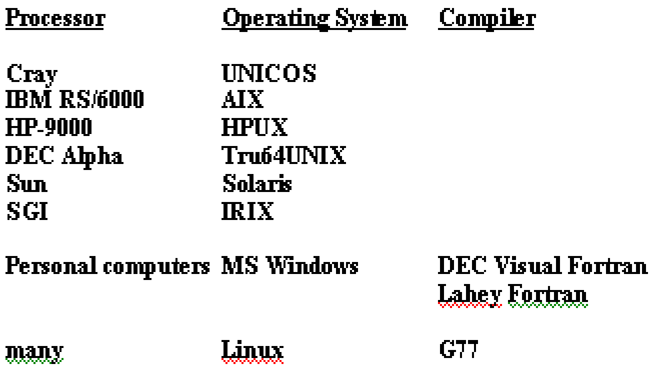


Figure 5. MCNP and LANL Super Computer Development History

Monte Carlo Technique

You are all familiar with dice. The sum of two dice can range from 2 to 12, with 7 being the most probable. While there is only one way to produce a 2 or a 12, there are six ways to produce a 7. For one throw, only one value will be obtained, although it may be the low probability result of 2 or 12 or the highest probability result of a 7. However, if the dice are thrown many times, say 1000, the distribution of values would approach the normal distribution.

The outcome of an interaction of a particle with matter also follows a statistical probability distribution. The outcome depends on the incident particle and its energy and the characteristics of the matter it is interacting with. Many different interactions can occur, each with a different probability or cross section, and additional particles can be produced.



Processor	Operating System	Compiler
Cray	UNICOS	
IBM RS/6000	AIX	
HP-9000	HPUX	
DEC Alpha	Tru64UNIX	
Sun	Solaris	
SGI	IRIX	
Personal computers	MS Windows	DEC Visual Fortran Lahey Fortran
many	Linux	G77

Figure 6. MCNP4C Supported Platforms

The probabilities for these various interactions are stored in large data tables that can be loaded into MCNP. MCNP starts by emitting a specified particle from the described source. A new random number is used to determine if and where a collision occurs. When the particle interacts with matter, MCNP generates a random number or numbers to select the interaction and its outcome. If another particle is produced, that particle's characteristics are stored for later use. The initial particle is tracked through its numerous interactions until it disappears. In turn, each of the generated and stored particles is tracked until it disappears. When all particles have been tracked, a new source particle is started. This sequence of events is repeated as many times as necessary to obtain the desired statistical accuracy of the results. It can take a thousand, a million, even a billion or more source particles to get the desired precision.

MCNP Uses

MCNP is used for many different types of applications. It currently has over 3000 users in more than 260 institutions worldwide. Initial uses included criticality safety and radiation shielding problems. These uses expanded to complex geometries and related uses such as material safeguards and nondestructive assay. The size of the source can vary from a small simple radioactive source, to a Co-60 or x-ray irradiation device and its facility to a large particle accelerator.

An example of this is the Particle Bed Reactor Critical Experiments (CX) conducted at Sandia National Laboratories around 1990 as part of the Space Nuclear Thermal Propulsion Program. A picture of the CX reactor (Parma, 1993) is shown in Figure 7. A comparison (Selcow, 1993) of experimental results with MCNP3B benchmark calculations is shown in Figure 8. The numbers in parentheses are 1 sigma standard deviations. The MCNP results are considered to be in close agreement with the experimental results, and in significantly better agreement than those from other computational methods.

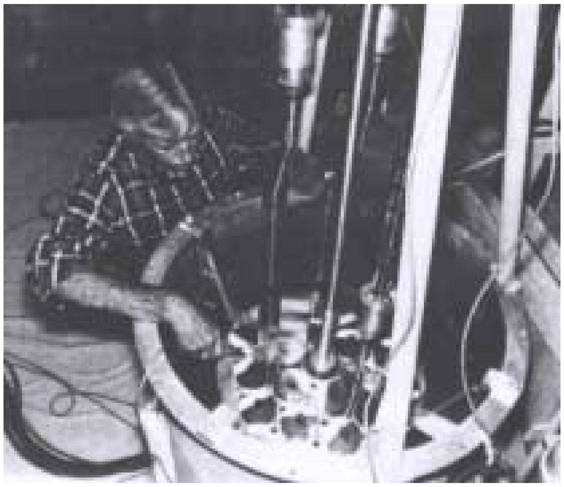


Figure 7. CX Reactor

Another related area is radiography. MCNP is used to model images to be produced from a radiograph of an object and to analyze the images produced from radiographs. It is also used for analyzing signals produced from logging oil wells with radioactive sources.

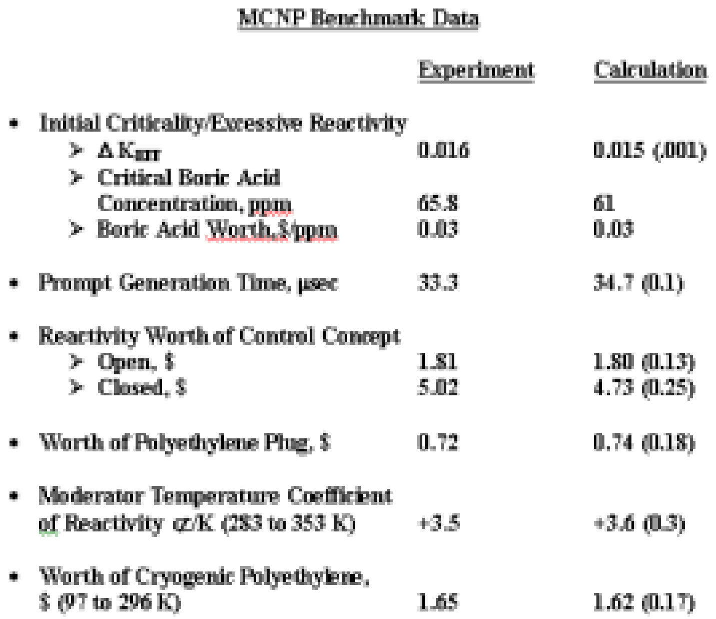
The medical sciences are also big users of MCNP. Not only is MCNP used to calculated dose to whole organs, it is used for calculating doses to tissues down to the cellular level such as blood vessel walls. Figure 9 shows the human chest as described for MCNP calculations used to plan nuclear medicine imaging, neutron capture therapy, and other medical imaging and therapy procedures.

Although the uses described here are on the meter scale, the physical size of the applications MCNP is used to analyze varies from hundreds of meters for accelerators to less than 10^{-6} m for cells.

Quality Control: Then

In order for a code to continue to be used worldwide by many different users for many different applications including ones of health and safety, it has to be a high quality code. MCNP is a code that is characterized by a strong emphasis on quality control, documentation, and research. Large codes such as MCNP have become a repository for physics knowledge. For example, MCNP represents more than 500 person-years of sustained effort. It contains about 50,000 lines of executable code in about 300 routines. The worldwide user community has strongly encouraged the quality efforts by the MCNP team.

In the early years, software quality control was done by publishing the physics equations to be solved and the coding used to solve them. Von Neumann's letter not only included the proposal and the coding sheets, but also the detailed equations that were being solved. This practice continued as the code evolved. In 1957, Cashwell and Everett (Cashwell, 1957) wrote a report detailing the physics and variation reduction methods used for the Monte Carlo codes used at LANL. Much of the physics described in that document is still in MCNP. They also included descriptions of twenty test problems that were run on Maniac I (Figure 10).



MCNP Benchmark Data

	Experiment	Calculation
• Initial Criticality/Exessive Reactivity		
➢ ΔK_{eff}	0.016	0.015 (.001)
➢ Critical Boric Acid Concentration, ppm	65.8	61
➢ Boric Acid Worth,$/ppm	0.03	0.03
• Prompt Generation Time, µsec	33.3	34.7 (0.1)
• Reactivity Worth of Control Concept		
➢ Open, $	1.81	1.80 (0.13)
➢ Closed, $	5.02	4.73 (0.25)
• Worth of Polyethylene Plug, $	0.72	0.74 (0.18)
• Moderator Temperature Coefficient of Reactivity ¢/K (283 to 353 K)	+3.5	+3.6 (0.3)
• Worth of Cryogenic Polyethylene, $ (97 to 296 K)	1.65	1.62 (0.17)

Figure 8. CX - MCNP Comparisons

The reports describing MCS (Johnston, 1963) and MCN (Cashwell, 1972) also included descriptions of the physics being solved. In addition to the physics, flow charts and code listings were included as well as a users manual for running the code suite. The report on MCG and MCP (Cashwell, 1973) included the physics, a users manual, sample problems with results, and code listings. With the merger of the various MC codes to form MCNP, the documentation was updated again. This manual (LANL TD-6, 1978) referred the reader to the publications by Cashwell (1957, 1972, 1973), for descriptions of the physics used. It then provided a users manual describing the input cards, their uses, and how to run the code on the then current supercomputer, the CDC 7600. Descriptions of the data structures, tally and plotting packages, and cross section libraries were also included. The current users manual (Briesmeister, 1997) continues in this direction, but also includes a summary discussion of the physics currently used in MCNP.

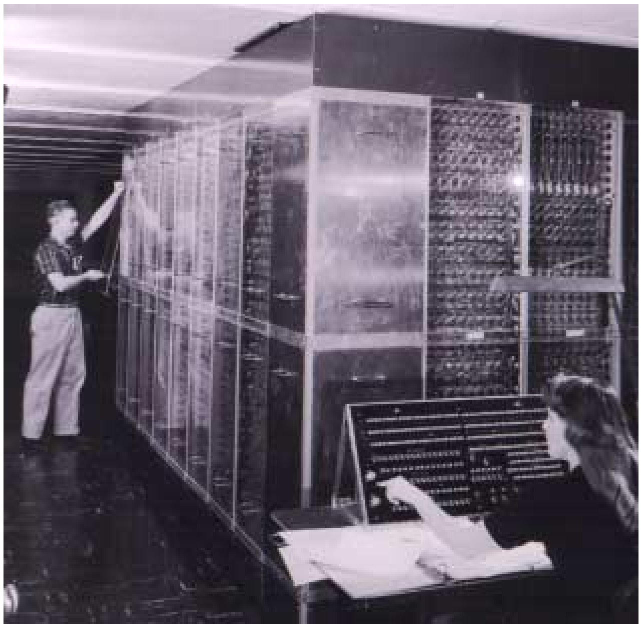


Figure 9. MANIAC I

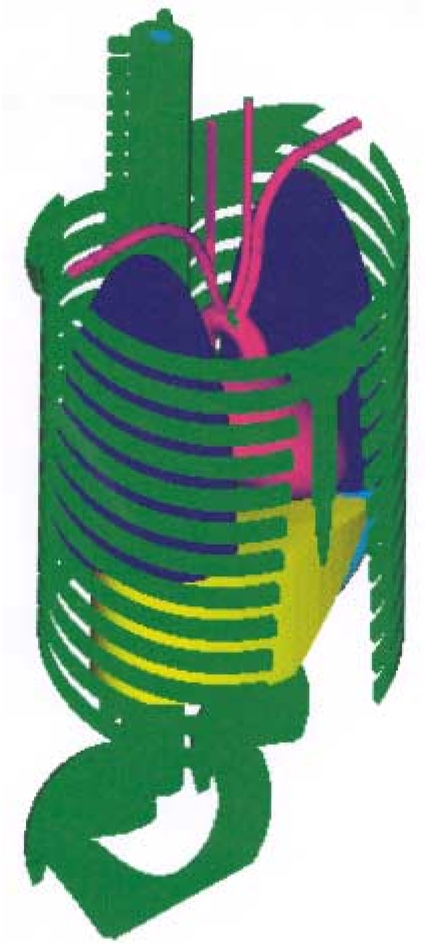
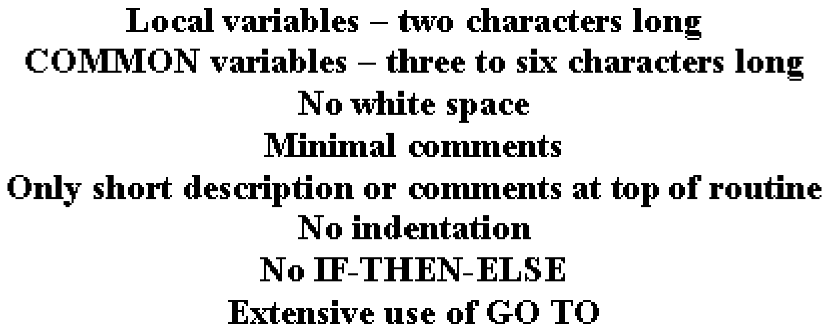


Figure 10. MCNP Input Human Chest

An increased emphasis on software quality occurred under the direction of Dr. Thomas Godfrey, the principal MCNP programmer from 1975 – 1989. He was given the job of creating a quality code out of the spaghetti from the merged versions. He also became the chief integrator of changes to the code, thereby controlling and reviewing everything that went into the code. To accomplish these tasks, he developed and enforced a coding style whose principle characteristics are terseness, concise code comments, and strict compliance to the ANSI FORTRAN 77 standard. Everything is accomplished in as few lines as possible with the goal of having routines and comments that are short enough to fit on a single page. It was felt that if a routine is short, another programmer could figure it out more easily. As a result, MCNP does more than some codes ten times its size. Some elements of this style are shown in Figure 11. One problem with this style is that is has not been published. Although team members know the style, they don’t always use it. As a result, one of the time-consuming tasks that the integrator has to perform is to convert submitted code into this style. Although this style is counter to modern programming philosophies, it has served MCNP well and preserved stylistic consistency throughout (Briesmeister, 1997).

When MCNP was created by combining codes in the 1970s, the principal format for source code was punched cards. Because the size of MCNP represented many drawers of punched cards, it was maintained as a disk file using Update and later Historian. Changes to the source code were made by adding modification decks that inserted, replaced, and deleted cards in the base released source version. By the time a major release was made, the source code contained many modification decks, and some cards were replaced many times as the modification decks were applied. When a major release was done, all the changes were merged into the base source deck producing a new single source deck. Also, the comment field on the cards was used to identify the programmer and date of that line of code. As a result, the source of every line of code in the program can be identified.



Local variables – two characters long
COMMON variables – three to six characters long
No white space
Minimal comments
Only short description or comments at top of routine
No indentation
No IF-THEN-ELSE
Extensive use of GO TO

Figure 11. Elements of Godfrey Formatting Style

In 1996, as a result of request from the user community, the MCNP Software Quality Assurance (SQA) Plan[10] was written and published. In it, an overview of the process used to maintain, review, and release the code was given. The function of a Board of Directors (BoD) composed of MCNP team members and members of the user community was described. They review and prioritize all requests for new features and enhancements, and then review the results of the development and testing effort on that feature, especially any changes which affect the user interface. Within the feature development and release process, there are also several reviews of the effort by part or all of the team. Bug fixes are reviewed only by the team. For intermediate releases for local testing as new features are added, the process is controlled by the chief integrator, and the reviews are done by team members. Major release, the ones that are released to the international user community, are reviewed and approved by both the team and the BoD.

However, that SQA plan had some deficiencies. One deficiency was that many of the activities were not adequately documented. Therefore, they could not be audited. Also, responsibility for some activities was not assigned. As a result, the execution of some activities could not be monitored. Other problems with the plan include the processes were not described in a consistent or complete manner and reviews may not as complete as they should be.

Quality Control: Now

As a result of a CMM-like survey performed in 1999, it was decided to use a commercial tool, Razor, to aid in formalizing and tracking the processes outlined in the SQA plan. This tool has three parts. The first is used for problem tracking. It can be used to track issues such as proposed new features and enhancements and bug fixes as they move through the development process. It can also be used to track intermediate and major releases through their process. Meeting minutes and review findings can be recorded in it. Controls can be set as to who can advance an issue through a process and under what conditions.

The second part of Razor is used for file control. It controls who can access a file and under what conditions, such as relating an action to an open issue. It can also be used to control who and under what conditions a file may be checked back in as a new version.

Release management is the third part of Razor. It is used to connect a set of files into a package. They can be source files used to create a new executable, or set of documentation files, or some combination of other files. A set of source files can be a new release or test set for testing a new modification to the program.

The first step in implementing the processes in Razor was preparing a detailed description of the processes. This included describing the inputs, actor, actions, and outputs from each step of each process. Also, all actions for each actor and descriptions of each document were listed. With the descriptions of each state in each process, who is allowed to change the states, and the permissible new states, the problem tracking part of Razor was set up, and the controlling scripts were written. Scripts were written to require relating issues to file checkout and checkin and who can access files were written. Further details can be found in Giesler (2000).

Once the Razor implementation was ready for testing, the Monte Carlo team was trained in using this implementation. This training included both familiarization with the Razor tools and a training exercise in which team members performed their roles in moving real issues through the process to release of modified code. This training exercise is being turned into a training program for future new team members.

The current new release has been installed in the file control and release management parts of Razor. Based on experience gained during the training exercise, the custom scripts in the problem tracking tool were modified for easier use. Also, the current list of proposed features and enhancements and reported defects has been included, and the Razor implementation is now being tested as the development environment for this team. Further modifications of this implementation will be developed as the team gains experience using it.

A very important part of MCNP is the data libraries. High quality data libraries are absolutely necessary to get accurate results from MCNP. Without high quality data libraries, the use of MCNP could be just another case of garbage-in garbage-out. Therefore, the quality of the code and the data libraries is very interrelated. A separate team, related to the Monte Carlo team, works full time evaluating the latest nuclear data sets available in order to provide the best data libraries for use with MCNP. Like MCNP, the data libraries represent hundreds of person-years of effort.

The data in the libraries comes from several sources. Evaluation centers in the U. S., Europe, Russia, Japan, and China evaluate published scientific results for inclusion in their compilations which they, in turn, provide to the world wide scientific community. These compilations are then evaluated using a series of computer codes to check the consistency of various values and characteristics of the data set before it can be incorporated in an MCNP data library. Checks include making sure cumulative sums total to 1.0, that energy scales are monotonic, and that there are no gaps in the data, especially near reaction minimum energies. An MCNP library with an updated compilation is then tested by running benchmarks with MCNP to insure that the libraries are read properly and to compare the calculated results with the measurements taken of the benchmarked configuration. After all these tests are passed, the updated data libraries are included in the next MCNP distribution to all users. Although other libraries can be used with MCNP, that is rarely done because of the quality of the included data libraries.

As for verification and validation, an extensive test suite has been developed. It not only covers most of the code and it tests most of the functions of the code. It also provides tracking of calculated results from version to version and environment to environment. As stated earlier, this practice began in the 1950s. As modifications were made to the code, the test package was also updated and continues to be updated to provide as complete code coverage as possible. The test suite for MCNP4A is described in Brockhoff (1994). One of the uses of the test suite is to verify that the installed code is running correctly.

Numerous benchmarks are used to check the accuracy of the calculated results from MCNP and its associated data libraries. Problems are run to compare the MCNP results with those of published solutions to analytic problems. Comparisons are made to running the same problems with other computer codes. After testing has verified that the algorithms have been correctly incorporated into MCNP, further testing is done to compare MCNP calculated results with real world measurements to insure the accuracy of the code.

Additionally, many users have their own test suites that they use to validate each new release of the code comparing the results of their test suites with the results they obtained for those tests from previous versions of MCNP. As a result, each new release is actually benchmarked by many hundreds of test problems.

As an indication of the confidence of the MCNP team in the quality of the product they produce, a cash reward is offered to any outside user who finds a previously unreported bug in the code. This can be a problem as small as one that occurs in only one operating environment. Since this reward was first offered in 1991 with the release of MCNP4, it has been paid only 102 times and only five times since MCNP4B was released three years ago.

The Future

The improvements in the software quality efforts for MCNP are a continuing activity. With the use of Razor, the development of modifications to the code can be tracked and audited. Also, all parts of package will be stored in one place with stricter controls on access and changes.

Using Razor to track the progress of an issue through the process, the execution of the process will become more formal. Also, the reviews will become more formal and complete. One result of this increased formality will be the collection of metrics evaluating the performance of the process. The collection of these metrics should lead to more improvements in the process.

As stated earlier, the current source code is in ANSI standard FORTRAN-77. Efforts have started for the conversion to ANSI standard FORTRAN-90/95. Because quality is the most important characteristic of this code, this conversion is being done with great concern so as not to introduce new defects.

One of the requests from the international user community has been ISO 9001 registration for this code process. Because MCNP is used in many applications related to health and safety, this registration would be very important to users. With the process improvements described above, this may be possible in the near future.

In this paper, I have discussed MCNP and its history and uses. I have also discussed the past and present states of its software quality control and its near future directions. MCNP has received praise from the international user community for its quality and its versatility. The intent for the future is to build on that reputation to produce an even better and more versatile product.

I would like to thank the LANL Monte Carlo team for its support and encouragement in preparing the presentation. Without the hard work of them and their predecessors over the last half century, this presentation would not be possible.

References

Abhold, H. M. & Hendricks, J. S. (1996). MCNP Software Quality Assurance Plan (Los Alamos National Laboratory Report LS-13138) Los Alamos, NM.

Briesmeister, J. F., ed. (1997). MCNP-A General Monte Carlo N-Particle Transport Code, Version 4B (Los Alamos National Laboratory Report LA-12625-M) Los Alamos, NM.

Brockhoff, R. C. & Hendricks, J. S. (1994). A New MCNP™ Test Suite (Los Alamos National Laboratory Report LA-12839) Los Alamos, NM.

Cashwell, E. D. and Everett, C. J. (1957). A Practical Manual on the Monte Carlo Method for Random Walk Problems (Los Alamos National Laboratory Report LA-2120) Los Alamos. published by Pergamon Press, Inc., New York (1959); updated in Carter, L. L. and Cashwell E. D. (1975) Particle-Transport Simulation with the Monte Carlo Method (ERDA Critical Review Series TID-26607) Springfield, VA: National Technical Information Service.

Cashwell, E. D., et.al. (1972). MCN: A Neutron Monte Carlo Code (Los Alamos National Laboratory Report LA-4751). Los Alamos, NM.

Cashwell, E. D., et.al.(1973). Monte Carlo Photon Codes: MCG and MCP (Los Alamos National Laboratory Report LA-5157-MS). Los Alamos, NM.

Giesler, G. C. (2000). Eolus Software Development Process and Its Implementation in Razor ™ (Los Alamos National Laboratory document LA-UR-00-1437) Los Alamos, NM.

Johnston, R. R. (1963). A General Monte Carlo Neutronics Code (Los Alamos National Laboratory Report LAMS-2856). Los Alamos, NM.

LANL Group TD-6 (1978). MCNP – A General Monte Carlo Code for Neutron and Photon Transport (Los Alamos National Laboratory Report LA-7396-M) Los Alamos, NM.

Parma, E. J., et.al (1993). A Critical Assembly Designed to Measure Neutronic Benchmarks in Support of the Space Nuclear Thermal Propulsion Program. Proceedings of the Tenth Symposium on Space Nuclear Power and Propulsion, AIP Conference Proceedings 271, 785-793.

Richtmyer, J. D. & von Neumann, J. (1947). Statistical Methods in Neutron Diffusion (Los Alamos National Laboratory Report LAMS-551). Los Alamos, NM.

Selcow,E. C., et.al (1993). MCNP Benchmark Analyses of Critical Experiments for the Space Nuclear Thermal Propulsion Program. Proceedings of the Tenth Symposium on Space Nuclear Power and Propulsion, AIP Conference Proceedings 271, 805-810.

MCNP is a trademark of the Regents of the University of California, Los Alamos National Laboratory.
Razor is a trademark of Visible Systems Corporation, Oneida, NY.

Neither the Regents of the University of California, the United States Government, nor any agency thereof, nor any of their employees, makes any warranty, expresses or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process, disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by The Regents of the University of California, the United States Government, or any agency thereof.