MCNP-BRL: A Linkage between MCNP and CAD Geometry

Kursat B. Bekar and Thomas M. Evans

Oak Ridge National Laboratory, P.O. Box 2008, MS 6172, Oak Ridge, TN 37831, USA,*
bekarkb@ornl.gov, evanstm@ornl.gov

INTRODUCTION

MCNP-BRL, an extended computer-aided design (CAD) geometry-driven version of MCNP5, was developed by linking MCNP5 [1] to the BRL-CAD [2] via a new geometry interface called the Geometry Interface for Transport (GIFT) library Application Programming Interface (API).

In this summary BRL-CAD implementation of the GIFT library in MCNP5 and the main features of MCNP-BRL are introduced. In addition, performance of the new code and its limitations are discussed.

OVERVIEW OF MCNP-BRL

MCNP5 (hereafter referred to simply as MCNP) is a widely used Monte Carlo transport code that was developed at Los Alamos National Laboratory. The physical problem geometry in MCNP is defined by the user through specification of surfaces and volumetric cells, which are formed from the intersections and unions of the surfaces.

The representation of geometric cells using this type of approach can be very tedious and is not possible for extremely complex geometries. Moreover, there is no direct linkage between common CAD-based engineering formats and MCNP-style geometric descriptions. Thus, no integrated tool exists for users wishing to run MCNP on existing engineering blueprints or designs. Instead, complex CAD geometries are converted into MCNP geometrical constructs at great cost and subject to a high degree of error.

The objective of the work discussed in this paper was to rectify this deficiency by integrating MCNP with a more powerful geometrical processor. By defining the GIFT programming interface, the geometry descriptions in MCNP were replaced with an independent package that performs the setup and ray tracing for Monte Carlo transport. After testing the feasibility of the GIFT library with a simple geometry package, it was connected to the BRL-CAD code package. The details of the linkage and the structure of the code package are described in Ref. 3.

The BRL-CAD package was chosen as a geometry processor for this application because of its many attractive features: (1) it provides efficient ray-tracing machinery that can be used directly by MCNP; (2) it can read a wide variety of industry-standard CAD formats; (3) it is actively developed and supported; (4) it provides its own powerful geometry editor, MGED, which enables users to develop geometries without recourse to an expensive CAD application; and (5) it is open source, so it is available freely to all potential MCNP-BRL users.

A sample BRL-CAD geometry is depicted in Fig. 1. The BRL-CAD model of this armored vehicle consists of many complicated regions, and this model can be used by MCNP-BRL directly (i.e., without modifications to the BRL-CAD model).

Fig. 1. A sample complicated geometric model prepared using BRL-CAD; M60A1 tank model with its crew.

In MCNP-BRL execution the first two sections of the MCNP input (cell and surface declarations) are skipped, and a BRL-CAD geometry file is provided in their place. The problem is initialized on the BRL-CAD geometry model, and the ray tracing required by Monte Carlo transport is accomplished through the BRL-CAD API. In other words the user provides the problem geometry in a BRL-CAD geometry database file and the MCNP problem parameters (material declarations, source and tally definitions, variance reduction parameters, etc.) in an MCNP-BRL input file. An additional input file is also required to provide the database filename, material densities, and region/cell volumes to the code package.

Supported MCNP Features

In the current version of MCNP-BRL, all of the MCNP features are operational except the following items: (1) mode: e electron transport, (2) surface sources (including read/write), (3) surface tallies, (4) lattices/repeated structures, and (5) transformations. These features have been deactivated because they are not needed within current projects. However, when/if needed, the MCNP equivalents of these features could be designed and integrated.

*Oak Ridge National Laboratory is managed and operated by UT-Battelle, LLC, for the U.S. Department of Energy under contract no. DE-AC05-00OR22725.

TESTING MCNP-BRL

Three different sets of test problems were designed to comprehensively test MCNP-BRL:

1. Set-I: problems to test the functionality of MCNP-BRL compared to the standard MCNP,
2. Set-II: problems to test and improve the GIFT library API, and
3. Set-III: problems to test and further improve the code performance.

For each test problem, special attention was given to maintain geometric consistency, to the extent possible/feasible, between MCNP-BRL and MCNP models to enable meaningful systematic comparisons.

Testing Supported MCNP Features

Problems were constructed to directly test MCNP features, such as source declarations, tally declarations, variance reduction mechanisms, etc., individually or collectively. These numerical experiments were conducted to ensure both codes produce identical results or agree with each other within the statistical uncertainties.

The results of these studies verified that MCNP-BRL supports the entire standard MCNP features except those listed in "Supported MCNP Features" above.

Testing and Improving GIFT API

The second set of test problems was used to test and improve the GIFT library API. In this set the numerical experiments were conducted for erroneously defined input cards or incorrectly defined geometries. (The problem geometries and errors in the geometry declarations are identical for both codes.) In this way detailed information was obtained to integrate the error-handling mechanisms, especially for geometry errors and MCNP-like error reporting mechanisms to the GIFT library API.

Testing Code Performance

In MCNP-BRL particle tracking and boundary crossing mechanisms are performed using BRL-CAD's ray-tracing algorithm. By limiting the number of hits (boundary crossing for a ray) to three during ray tracing, MCNP-BRL determines the next collision boundary and cell/region at the same time [3]. In contrast, MCNP follows the particle up to the cell boundary, and then it finds the next cell/region for the particle track. Thus, the number of cells and also the complication of the cell declaration in the problem geometry affect the code performance.

To evaluate the performance of the code in terms of computational time, further test cases with complicated geometries were constructed. These experiments included the following:

- many regions/cells, most of them having simple declarations (a region is bound by 1–10 surfaces) and
- many regions/cells, most of them having complex declarations (a region is bound by 10–200 surfaces).

Using this set the computational performances of MCNP-BRL and MCNP were compared.

Effect of the Number of Cells on Code Performance

A PWR core/vessel model [4] was selected to evaluate the code performance while the number of cells increased in the problem geometry. One-fourth of the core/vessel geometry was modeled in detail for both codes. Then, by homogenizing several assemblies in the core, the geometry was simplified, thereby decreasing the number of cells in the problem geometry.

The results from both codes for each problem were confirmed to be identical. Figure 2 presents the total computation time (initialization and particle tracking) for both codes. The results show that MCNP-BRL is 4–12 times slower compared to MCNP for the geometries that include many cells with simple cell declarations.

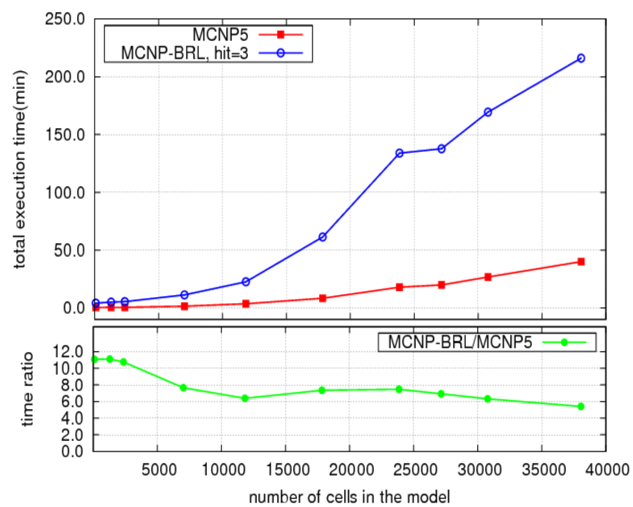

Fig. 2. Comparison of computational times.

Effect of Complex Cell Declarations on Code Performance

Because MCNP-BRL's particle-tracking and boundary-crossing features are not the same as those in MCNP, the complexity in cell declarations should also affect code performance. To investigate this a full-scale pressurized water reactor (PWR) facility [5], depicted in Fig. 3, was modeled for both codes. The geometry of this facility was divided into sections, and each section was also used for this analysis to demonstrate the different problems.

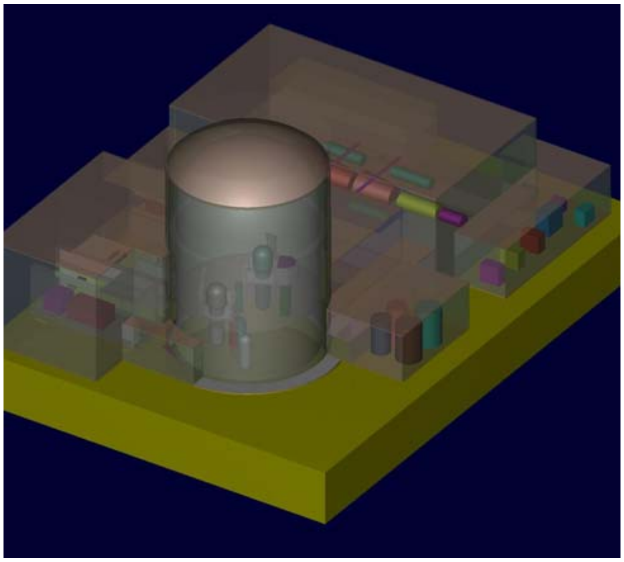

Fig. 3. BRL-CAD model for the PWR facility problem.

For this numerical exercise, some of the cells/regions were combined into one cell/region to obtain cells with more complicated declarations. In addition, MCNP was modified so that the “mlgc” parameter (which limits the number of words to 1,000 for a cell declaration) was increased from 1,000 to 10,000 such that MCNP could handle the cells with very complex declarations.

All these numerical experiments were simulated with both codes. The computational times of each numerical experiment for two codes, number of cells in the model geometry, and number of surfaces that bound the most complicated cell are presented in Table I. The results indicate that MCNP-BRL is 2–9 times slower than MCNP for this kind of application when the most complicated cell in each model problem is bounded by only 20–43 surfaces. However, for the geometries with the most complex cell declarations, the MCNP-BRL execution time becomes comparable to, and sometimes less than, the MCNP execution time. This is due to the efficient ray-tracing capabilities of BRL-CAD for complex cell descriptions.

All these results show that MCNP-BRL is operational while maintaining most of the MCNP features.

TABLE I. Effects of complex cell definitions on code performance.

Sections	Number of cells	Number of surfaces for most complex cell	CPU time (minutes)		Ratio
			MCNP	MCNP-BRL	
Turbine bldg.	126	20	11.03	23.96	2.17
	5	82	44.99	28.84	0.64
Auxiliary bldg.	195	27	12.38	36.61	2.96
	4	177	152.51	54.58	0.36
Containment	61	43	14.79	60.54	4.09
	6	86	37.17	69.33	1.87
Full model	393	43	15.36	135.28	8.81

CONCLUSION

MCNP-BRL, a CAD geometry-driven version of MCNP, was developed for nuclear applications in which complex geometry features and geometric input preparation become issues for standard MCNP. With this code package, the MCNP geometry package was replaced with a highly-capable independent code package, BRL-CAD, to perform the setup and ray tracing for Monte Carlo transport.

The code package was comprehensively verified by evaluating an extensive set of test problems. The results demonstrate that MCNP-BRL is operational with most MCNP features.

Although MCNP-BRL was found to generally be less efficient than MCNP in terms of execution time, MCNP-BRL enables simulations on very complex geometries that are otherwise not possible with standard MCNP. Furthermore, the direct coupling between CAD package and transport code has tremendous benefits in terms of eliminating/reducing the time required for transport-code geometric input generation, reducing analysis risks associated with human errors in input generation and increasing confidence and efficiency in performing analysis.

ACKNOWLEDGMENTS

The authors would like to thank Oak Ridge National Laboratory’s Radiation Transport Group members, J. O. Johnson, D. E. Peplow, E. D. Blakeman and J. C. Wagner for providing sample problems/models for us during this study. This work was sponsored by the Defense Threat Reduction Agency (DTRA). Oak Ridge National Laboratory is managed and operated by

UT-Battelle, LLC, for the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

REFERENCES

1. MCNP5 Manual, X-5 Monte Carlo Team, "MCNP-A General Monte Carlo N-Particle Transport Code, Version 5," Los Alamos National Laboratory (2005).
2. L. BUTLER and E. EDWARDS, *BRL-CAD Tutorial Series: Volume I—Overview and Installation,* Tech. Rep. ARL-SR-113, Arm Research Laboratory (2002).
3. K. B. BEKAR and T. M. EVANS, "MCNP-BRL: An External Geometry-Driven Version of MCNP," in preparation for submission to *Nucl. Sci. Eng*.
4. H. P. SMITH and J. C. WAGNER, "A Case Study in Manual and Automated Monte Carlo Variance Reduction with a Deep Penetration Reactor Shielding Problem," *Nucl. Sci. Eng.,* **149**, 23–37 (2005).
5. E. D. BLAKEMAN et al., "PWR Facility Dose Modeling Using MCNP5 and the CADIS/ADVANTG Variance-Reduction Methodology," ORNL/TM-2007/133, Oak Ridge National Laboratory (2007).