Development of CAD-Based Geometry Processing Module for a Monte Carlo Particle Transport Analysis Code

Sung Hoon Choi [a*], Min Su Kwark [a], Hyung Jin Shim [a]
[a] *Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul, Korea, 151-742*
[*] *Corresponding author:hooni86@snu.ac.kr*

1. Introduction

As The Monte Carlo (MC) particle transport analysis for a complex system such as research reactor, accelerator, and fusion facility may require accurate modeling of the complicated geometry. Its manual modeling by using the text interface of a MC code to define the geometrical objects is tedious, lengthy and error-prone. This problem can be overcome by taking advantage of modeling capability of the computer aided design (CAD) system.

There have been two kinds of approaches to develop MC code systems utilizing the CAD data: the external format conversion and the CAD kernel imbedded MC simulation. The first approach includes several interfacing programs such as McCAD [1], MCAM [2], GEOMIT [3] etc. which were developed to automatically convert the CAD data into the MCNP [4] geometry input data. This approach makes the most of the existing MC codes without any modifications, but implies latent data inconsistency due to the difference of the geometry modeling system. In the second approach, a MC code utilizes the CAD data for the direct particle tracking or the conversion to an internal data structure of the constructive solid geometry (CSG) and/or boundary representation (B-rep) modeling with help of a CAD kernel. MCNP-BRL [5] and OiNC [6] have demonstrated their capabilities of the CAD-based MC simulations.

Recently we have developed a CAD-based geometry processing module for the MC particle simulation by using the OpenCASCADE (OCC) library [7]. In the developed module, CAD data can be used for the particle tracking through primitive CAD surfaces (hereafter the CAD-based tracking) or the internal conversion to the CSG data structure. In this paper, the performances of the text-based model, the CAD-based tracking, and the internal CSG conversion are compared by using an in-house MC code, McSIM, equipped with the developed CAD-based geometry processing module.

2. Methods

2.1 CAD-Based Tracking Algorithm

In the MC particle tracking, the geometrical data are generally needed for two purposes: to calculate the distance-to-surface (DTS) of a solid in which a particle is located for given location and direction and to find out the next solid to which a particle migrates. To do these, firstly the entire solid and face data are stored by using the OCC library functions to process CAD files. To calculate the DTS, crossing points of a line created from the particle location in its direction with bounding faces of the solid are searched. The next solid determination algorithm can be implemented by an in/out check routine for a solid which OCC offers.

This algorithm is easy to implement by using the CAD kernel library functions but requires intensive optimization to enhance the calculation performance. Note that the developed CAD tracking routine was implemented by using the supplied OCC library as it is.

2.2 Internal CSG Conversion Algorithm

he To speed up the CAD-based tracking algorithm, a solid of which boundaries are defined by the primitive surface such as plane, cylindrical, spherical surfaces can be converted into the existing CSG data structure in the MC code. In this case, the geometrical information extracted from the CAD data is as follows:

(a) Equations of the solid boundary surfaces,
(b) Direction (+, -) of the surfaces surrounding the solid,
(c) Surface connection data (AND, OR).

The (a) and (b) data can be easily obtained by using the OCC library functions while the (c) data can be obtained by the following sequence.

(1) Find all vertices that make up a solid.
(2) If the vertices are located only in one side of a plane which bounds the solid, then the plane is assigned to *convex-type*. If not, the plain becomes *concave-type*.
(3) If a plane is *concave-type*, divide the solid by the plane and connect the divided solids by the union (OR) operator.
(4) Repeat the process step 1~3 until all the divided solids become convex.
(5) Define the divided convex solid by the intersections (ANDs) of its boundary surfaces and combine all the divided convex solids with the OR operators

These sequences are described in Fig. 1.

3. Numerical Results

The performance of the developed CAD-based geometry processing module is tested by using

McSIM/CAD for the Godiva criticality [8] and the shielding benchmark problems [9].

The McSIM/CAD eigenvalue calculations are performed with 20 inactive and 100 active cycles on 100,000 histories per cycle varying the geometry processing algorithm: the text-based model, the CAD-based tracking, and the internal CSG conversion. Table I shows comparisons of k_{eff} and the elapsed times of three McSIM calculations. From the same results of k_{eff}, the geometry processing modules are validated. From the table, we can observe that the CAD-based tracking routine is 473 times slower than the text-based model while the internal CSG conversion method accomplishes almost the same calculation speed.

Table I: Performance of the CAD-based geometry processing module for Godiva

	McSIM (Text-based)	McSIM/CAD	
		CAD tracking	CSG Conv.
k_{eff}	0.99666 (23)[a]	0.99666 (23)	0.99666 (23)
Time	18.6 sec	8792.1 sec	18.7 sec

[a] standard deviations are described in brackets ( ).

In order to validate McSIM/CAD, its results are compared with other MC codes' for the shielding benchmark problem [9]. The shielding benchmark problem calculations are performed with 100,000,000 histories. The systems are one-dimensional spherical model. The Cf-252 source which is surrounded by the copper shell is assumed to be a point at the sphere's center. Outer region consist of an iron shell. Table II shows comparisons of the elapsed times of McSIM/CAD, McCARD calculations.

Table II: Performance of the CAD-based geometry processing module for the shielding benchmark problem

	McSIM/CAD	McCARD
Time	2001 sec	2120 sec

Figure 2 shows comparisons of the spectra of neutrons emitted from the surface of the iron sphere with diameter of 40.0 cm at whose center the Cf-252 radionuclide is placed for McSIM/CAD, MCNP, McCARD and measurements. In Fig. 2, the results of the three MC code results overlap each other with the average standard deviation of 0.49%.

From these results, we can draw the conclusion that CAD-based geometry processing module operate properly.

Acknowledgement

This work is supported by Agency for Defense Development (Contract No. UC080023GD).

REFERENCES

[1] A. Serikov et al., "Use of McCad for the Generation of MCNP Models in Fusion Neutronics," *Proc. Of International Conference on Mathematics, Computational Mothods & Reactor Physics* (M&C 2009), Saratoga Springs, NY, May 3-7, 2009, 2009.

[2] Y. Wu, FDS Team, "CAD-Based Interface Programs for Fusion Neutron Transport Simulation," *J. Fusion Eng. Des.*, **84**, 1987, 2009.

[3] S. Sato, et al., "Development of CAD-to-MCNP Model Conversion System and Its Application to ITER," *Nucl. Technol.*, **168**, 843, 2009.

[4] J. F. Briesmeister, "MCNP-A General Monte Carlo N-Particle Transport Code", Version 4C, LA-13709, *Los Alamos National Laboratory*, 2000.

[5] K. B. Bekar, T. M. Evans, "MCNP-BRL: a Linkage between MCNP and CAD geometry," Trans. Am. Nucl. Soc., 101, 623, 2009.

[6] F. Fleurot, K. Searson, "OiNC, Criticality Benchmark Models in CAD," ICNC2011-*International Conference on Nuclear Criticality*, Edinburgh, Scotland, Sep. 19-23, 2011.

[7] OpenCASCADE^{TM}, http://www.opencascade.org

[8] "International Handbook of Evaluated Criticality Safety Benchmark Experiments," NEA/NSC/DOC(95)03, OECD Nuclear Energy Agency, 2006.

[9] Gennady M., Yevgeniy R. and Lev T., "Neutron and photon leakage spectra from Cf-252 source at centers of six iron spheres of different diameters", NEA/NSC/DOC/(05)03, Vol. 8, 2008.

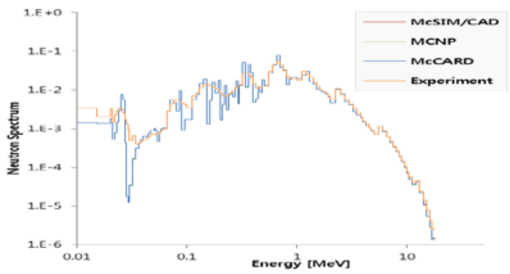

Fig. 2. Comparison of neutron spectra for the shielding benchmark problem

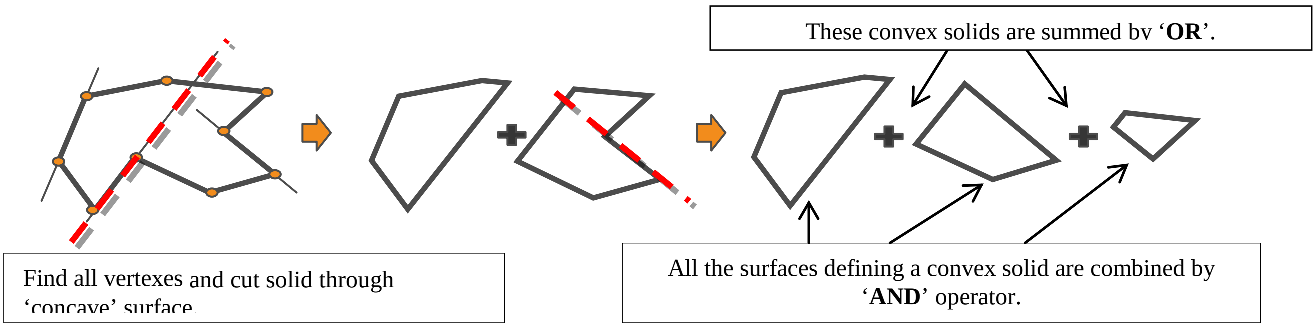

Fig. 1. Algorithm for making convex solids from a concave on