Calculation of Effective Multiplication Factor by OpenMC and MCNP

Dong Hun Shin[a], Kyoon Ho CHA[a*]
[a]Dept. of Quantum and Nuclear Engineering, Sejong University, 209 Neungdong-ro, Gwangjin-gu, Seoul, Republic of Korea (05006)
[*]*Corresponding author: Khcha@sejong.ac.kr*

1. Introduction

Recently, as interest in SMR [1] has increased, the development of new nuclear fuel has become an important issue. For this, it is essential to have the ability to perform accurate depletion calculation on newly developed fuels. MCNP [2], which is originally recognized as a code for licensing calculation, is mainly used to evaluate the performance of the assembly. However, due to the development of computer technology, many high-performance codes such as OpenMC [3] have been developed and the number of evaluation tools that could be selected during the calculation process has increased.

OpenMC is an open-source simulation code in Monte Carlo. It supports XML [4] input and python API-based input. Therefore, scalability is easy, and it is easy to implement a structure with many repetitive tasks such as reactor design process. In addition, since the ACE file-based library is used, calculation values can be directly compared with MCNP codes widely used in criticality analysis and shielding calculation in the nuclear power field. In this paper, an assembly geometry is created and deplete it using OpenMC and MCNP code. Then, the effective multiplication factors calculated by the two codes are compared.

2. Methods and Results

In this section, we describe the process of implementing an assembly and deplete it with OpenMC and MCNP code to compare the effective multiplication coefficients. The assembly consist of nuclear fuel, coolant and guide tube.

2.1 Implementation of Assembly

OpenMC basically uses an xml format. It can also be implemented as a python API. OpenMC has three essential xml files that must be created. The First is a material.xml file that defines density, mass and cross section of material used under a given problem. In this study, python API is used to generate material.xml. The elements and density of each material is defined by the material method. Then, each defined material is grouped into a material method. This series of processes makes it easier to generate xml file using Python API.

The method of implementing geometry is quite similar to the structure of the MCNP input. Like MCNP, the surface is first defined, and then the relevant material and surface are specified in the cell card. Table 1 is summarizing the fuel rods used in this study.

Table 1 Fuel rod information

Material	Fuel rod 1	Fuel rod 2	Fuel rod 3
U-235 (a/o)	1.03659e+21	9.21665e+20	1.47561e+20
U-238 (a/o)	2.16703e+22	2.17838e+22	2.03462e+22
O-16 (a/o)	4.53922e+22	4.53922e+22	4.39109e+22
Gd-nat (a/o)	-	-	4.18113e+19

The density of fuel 1 and 2 is 10.176 g/cc and the density of fuel 3 is 9.992 g/cc. fuel outer radius is 0.413 cm, and cladding inner and outer radius are 0.4215 cm and 0.485 cm. The outside of the fuel is filled with a coolant, which is composed of H_1 and O_{16} in an atom percent ratio of 1:2, and the density is 0.699 g/cc. A space between the inner and outer radius of the cladding is filled with helium gas, and the density is 0.0014 g/cc. Table 2 provides material information and density for fuel cladding and guide tube respectively.

Table 2 Fuel cladding and guide tube

	Fuel cladding	Guide tube
Percent_type	Atom percent	Atom percent
Zirconium	3.638e+22	4.148e+22
Stannum	5.556e+20	6.335e+20
Ferrum	5.450E+20	6.216e+20
Chrom	3.704e+19	4.223e+19
Nicola	1.852e+19	2.111e+19
Density (g/cc)	5.8075	6.528

Like MCNP, OpenMC can implement universe. This is important when implementing assembly, so we create a universe for each fuel. The width and height of the universe are set to 1.258, which is the pitch of nuclear fuel. Figure 1 is visualization of universe of fuel rod 1 to 3 implemented with OpenMC.

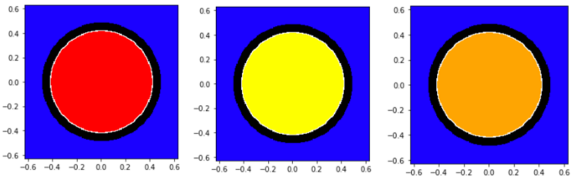

Fig. 1. Visualization of fuel rods.

In the case of the guide tube, it is twice as long in width and length as 1.258 cm, which is the nuclear fuel pitch. Therefore, the guide tube should be divided into four parts, top, bottom, left, right and implemented as each universe. Figure 2 is guide tube divided into four parts.

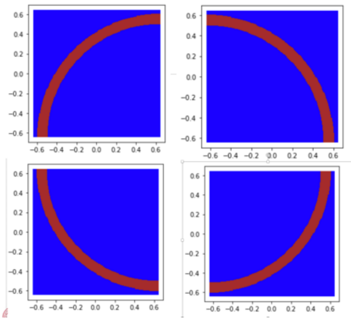

Fig. 2. Visualization of guide tube.

An assembly is implemented by combining the universe of fuel rod and guide tube. Python methods already exist in OpenMC to make assembly easy. To use this method, the location of the assembly and the number of pitches must be set. The position of the assembly is set as a reference point, and the total number of pitches is 16 by 16. Figure 3 shows the assembly that is finally implemented.

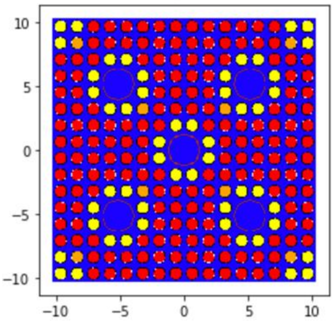

Fig. 3. Visualization of assembly.

2.2 Calculation of Effective Multiplication Factor

Through the above process, material.xml and geometry.xml are created. To deplete in OpenMC, setting.xml is additionally required. Setting.xml consists of number of batches, the number of inactive bathes, and the number of particles, and etc. The same number is applied to MCNP and OpenMC and set the boundary condition to reflective for full-core simulation. A Cross section is used ENDF/B-VII.$_1$. [5] Table 3 describe parameter values that used in the simulation and calculation result.

Table 3 Parameters and result

	MCNP	OpenMC
Batch	600	
Inactive batch	300	
Particles	1000000	
Time step (s)	2592000	
Result	1.38076 +/– 0.0005	1.38191 +/– 0.0004

3. Conclusions

In this paper, assembly geometry is implemented through OpenMC. To perform this task, the type and density of the material to be used first is determined and defined as one object. Then, the geometry is defined, and each fuel rod and guide tube are designated as a universe, and the assembly is constructed with their combination.

The effective multiplication factor for assembly is calculated by MCNP and OpenMC code, and result is compared. The values calculated by OpenMC show similar values to MCNP, indicating that there are, various tools that can be used in designing new nuclear fuels in the future.

REFERENCES

[1] IAEA, Advances in small modular reactor technology developments. A supplement to: IAEA advanced reactor information system (ARIS), International Atomic Energy Agency, 2014.

[2] J.F. Briesmeister (Ed.), mcnp-A General Monte Carlo N-Particle Transport Code, Version 4C, Los Alamos National Laboratory, Report LA-13709-M (April 2000).

[3] P.K. Romano, A.R. Siegel, B. Forget, K. Smith
Data decomposition of Monte Carlo particle transport simulations via tally servers

[4] Bray,T., Paoli,J. and Sperberg-McQueen,C.M. (1998) Extensible markup language (XML) 1.0, Available via the World Wide Web at http://www.w3.org/TR/1998/REC-xml-19980210.

[5] M. B. Chadwicket al., "ENDF/B-VII.1: Nuclear Data for Science and Technology: Cross Sections, Covariances, Fission Product Yields and Decay Data," Nucl. Data Sheets, 112, 2887 (2011).