

An Efficient Approach for Modeling Hexagonal Lattice in McCARD

Min Jae Lee and Jae-Yong Lim

Fast Reactor Development Division, Korea Atomic Energy Research Institute
111 Daedeok-daero 989 beon-gil, Yuseong-gu, Daejeon 34057, KOREA

*Corresponding author: lmj@kaeri.re.kr

1. Introduction

McCARD [1] is a Monte Carlo simulation code developed by SNU and it has been widely used for neutron and gamma transport calculations. It equips a built-in PYTHON script interpreter for input processing, and this allows users to prepare a well-structured input file based on the simple script functions.

Recently, KAERI uses McCARD code for fast reactor design and analysis, and it is found that modeling fast reactors in McCARD can be quite complicated and messy compared to other MC codes such as MCNP [2] and Serpent [3] since other MC codes support simple core loading input processing through lattice input cards while McCARD does not. The lack of lattice option, users have to specify every single surfaces and cells in a hexagonal lattice and this leads to a large probability of human errors. To overcome this, a simple but efficient approach for developing hexagonal lattice in McCARD suggested in this paper, hoping that the suggested approach can reduce human resources that are wasted for struggling complicated input processing in McCARD.

2. McCARD Input Processing for Hexagonal Lattice

2.1 Defining Surfaces of Hexagons

In order to define the surface on a hexagonal lattice, three axis are defined first as plotted in Fig 1.

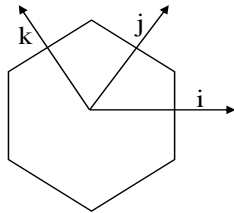


Fig. 1. Definition of axis in a hexagonal lattice

The surfaces orthogonal to axes i , j , k are called surface A , B and C as plotted in Fig. 2. The positive direction and negative directions of each surface are also appeared in Fig 2. Note that the index 0 stands for the origin.

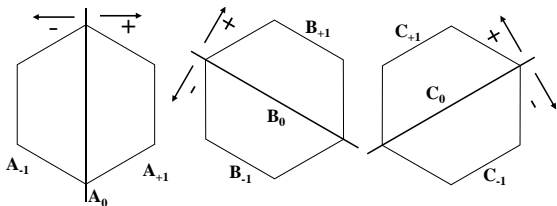


Fig. 2. Surfaces of a hexagonal lattice

Assuming that the pitch of hexagon is P , the equation that presents each surface is give as follows:

$$A_i : x = \frac{P}{2} \cdot i \quad (1)$$

$$B_i : x + \sqrt{3}y = P \cdot i \quad (2)$$

$$C_i : -x + \sqrt{3}y = P \cdot i \quad (3)$$

In this manner, the hexagon is defined in McCARD as follows:

$$\{ +A_{-1} \text{ AND } -A_{+1} \text{ AND } +B_{-1} \text{ AND } -B_{+1} \\ \text{ AND } +C_{-1} \text{ AND } -C_{+1} \} \quad (4)$$

2.2. Surface Index in Hexagonal Lattice

The ordering of hexagons in a hexagonal lattice can be simply presented as Fig. 3. The index of capital I , J , K , L , M and N mean rotational indexes of a hexagonal lattice, which represents 1/6 of a core. The index (j, i) of each hexagon represents j -th row and i -th column in each rotational index.

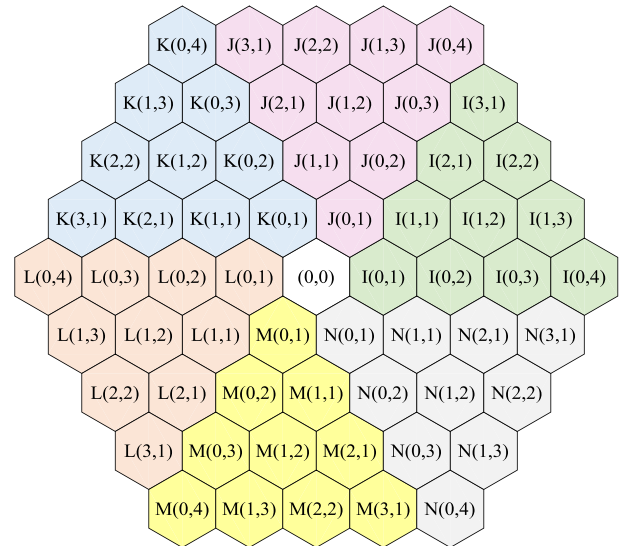


Fig. 3. Indices of hexagons

In this system, the hexagon at $I(0,1)$ can be defined with as follows:

$$\{ +A_{+1} \text{ AND } -A_{+3} \text{ AND } +B_0 \text{ AND } -B_2 \\ \text{ AND } +C_{-2} \text{ AND } -C_0 \} \quad (5)$$

Comparing Eq. (5) to Eq. (4), the indices are increased 2 and 1 for Surface A and B respectively, while it is decreased by 1 for Surface C. In this manner one can find index increment of each surface for the next hexagon. The index change for the next column is summarized in Table I.

Table I: Index change for the movement of next column

Rotational Index	Surface A	Surface B	Surface C
I	2	1	-1
J	1	2	1
K	-1	1	2
L	-2	-1	1
M	-1	-2	-1
N	1	-1	-2

Note that the index change for the next row is exactly the same as the movement for the next column of the next rotational index. For example, the movement for next row in rotational index I is exactly the same as the movement for next column in rotational index J. Therefore, the index increment for the next row can be summarized as Table II.

Table II: Index change for the movement of next row

Rotational Index	Surface A	Surface B	Surface C
I	1	2	1
J	-1	1	2
K	-2	-1	1
L	-1	-2	-1
M	1	-1	-2
N	2	1	-1

With table I and II, the surrounding surface at arbitrary position can be easily found. For example, the index increments at $I(3,1)$ can be defined as follows:

$$\begin{aligned} \text{Surface A} &: 3x_1 + 1x_2 = 5 \\ \text{Surface B} &: 3x_2 + 1x_1 = 7 \\ \text{Surface C} &: 3x_1 + 1x_1 = 2 \end{aligned}$$

Therefore, the hexagon at $I(3,1)$ can be defined in McCARD as follows:

$$\{ +A_{+4} \text{ AND } -A_{+6} \text{ AND } +B_{+6} \text{ AND } -B_{+8} \text{ AND } +C_{+1} \text{ AND } -C_{+3} \} \quad (6)$$

The translation vector is also required in McCARD input processing, and the translation vector in x-y plane can be also simply obtained the same way as finding surface index. Table III shows the x and y position change for x-direction move.

Table III: Position change for the movement of next column

Rotational Index	x	y
I	P	0
J	P/2	$\sqrt{3}/2 \cdot P$
K	-P/2	$\sqrt{3}/2 \cdot P$
L	-P	0
M	-P/2	$-\sqrt{3}/2 \cdot P$
N	P/2	$-\sqrt{3}/2 \cdot P$

2.3 Algorithm for Building a Hexagonal Lattice

With tables I and II, one can easily find 6 surfaces surrounding a hexagon at an arbitrary position. So a hexagonal lattice can be simply constructed by following algorithm at Fig. 4

```
// Cells
Define a hexagon at origin
for irot in (I, J, K, L, M and N):
  for j in ( 1 .. NRing )
    for i in ( 1 .. NRing - j )
      Define a hexagon at irot (j, i)
    end
  end
end

// Surfaces
for i in (surface indexes)
  Define surface Ai (SUR Ai PX  $\frac{P}{2} \cdot i$ )
  Define surface Bi (SUR Bi PG  $1 \sqrt{3} 0 P \cdot i$ )
  Define surface Ci (SUR Ci PG  $-1 \sqrt{3} 0 P \cdot i$ )
end
```

Fig. 4. Algorithm for building a hexagonal lattice

Note that the boundary of hexagonal lattice is the out-most boundary of the problem, user have to specify void hexagon cells so that McCARD Monte Carlo simulation can stop tracking the neutron beyond these cells.

2.4 Treatment for Duct

Concave cells are not allowed in McCARD, duct or duct-like structures should be split into 6 cells. Defining each cell in a duct can be sometimes complicated especially for dual-duct structures such as control rod assemblies. By assigning proper surface indexes, duct cells can be defined with a simple pattern. Figs. 5 and 6 show an example defining a duct in McCARD. Note that the function for surface A, B and C are the same as Eqs (1) through (3). Duct inner and outer pitches are used instead of $P \cdot i$.

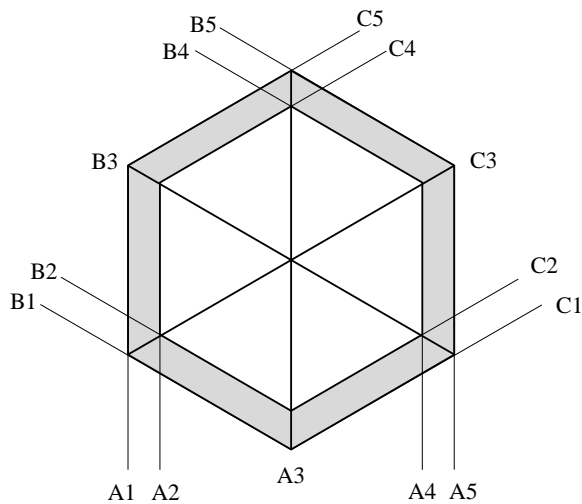


Fig. 5. Typical duct and surfaces

[3] Leppanen, J., et al., The Serpent Monte Carlo code: Status, development and applications in 2013. Ann. Nucl. Energy, Vol.82, P. 142, 2013.

DUCT1 : { +A4 AND -A5 AND +B3 AND -C3 }
DUCT2 : { +B4 AND -B5 AND +C3 AND +A3 }
DUCT3 : { +C4 AND -C5 AND +B3 AND -A3 }
DUCT4 : { +A1 AND -A2 AND +C3 AND -B3 }
DUCT5 : { +B1 AND -B2 AND -C3 AND -A3 }
DUCT6 : { +C1 AND -C2 AND +A3 AND -B3 }

Fig. 6. Definition of duct cells in McCARD

Multiple-duct structure also can be defined similarly using the simple pattern found in Fig. 6.

3. Conclusions

Although McCARD does not provide lattice input cards, the built-in PYTHON script interpreter allows users to model a complex nuclear reactor with reduced efforts. However, modeling a hexagonal lattice problem is still complicated because of its complex geometry.

In this work, an effective approach to model hexagonal lattice problem is suggested. The index of surface and hexagonal cells are properly defined, and the scheme for finding surfaces for each cell is provided explicitly. With this approach, even new McCARD users, who are not familiar with hexagonal lattice system, can model hexagonal reactor problems efficiently.

ACKNOWLEDGEMENTS

This work was supported by a National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIP) (No. NRF-2012M2A8A2025622).

REFERENCES

- [1] H. J. Shim, B. S. Han, J. S. Jung, H. J. Park and C. H. Kim, McCARD: Monte Carlo code for advanced reactor design and analysis. Nuclear Engineering and Technology. Vol.44. 2012.
- [2] D. Pelowitz, MCNP6 User's Manual Version 1.0, LA-CP-13-00634, May 2013.