Development of Multi-group Cross Section Processing Program for MUST Unstructured Discrete Ordinate Transport Code

Myeong Hyeon Woo, Ta Duy Long, Ser Gi Hong [*]
Department of Nuclear Engineering, Hanyang University, 222, Wangsimni-ro, Seongdong-gu. Seoul, Korea
[*]*Corresponding author: hongsergi@hanyang.ac.kr*

1. Introduction

The MUST (Multi-group Unstructured-geometry S_N Transport) is a deterministic code developed to solve the transport equation in complex geometry based on tetrahedral mesh [1]. In this code, the transport equation is formalized using the Discontinuous Galerkin (DFEM) method and the self-developed LDEM-SCB method [2]. The MUST code has a built-in parser that reads and processes the nuclear cross section of the ISOTXS format, and uses this parser to change the ISOXS file into a transport table that can be directly applied to the transport calculation. However, The ISOTXS format is still widely used in today, but there are some limitations. ISOTXS format syntax is not extendable because it uses grammatically fixed flags and cannot add reactions or particles. The MATXS format is designed to classify all information using the Hollerith identifier to order to be extendable. The Hollerith identifier simply means a char-type variable, but for historical reasons this name is still used to describe the MATXS format, so it has been used here as well. In order to use the nuclear data libraries distributed in MATXS format, an additional procedure to change to ISOTXS was required. Therefore, in this study, in order to simplify the cross section processing procedure, we have developed a program that can directly process MATXS files and an automation program that builds MATXS library using NJOY code [3,4], thereby enhancing user convenience and code independence.

2. Cross Section Processing System

The developed system consists of two parts. One is a program that creates a MATXS file from ENDF/B library using NJOY cross section processing code, and the other is a program that creates a transport table for MUST code by reading and processing the MATXS file directly. Each program is an independent program, and the names of each program are as follows.

- NIBGenMATXS: Automatic NJOY Input and Batch file Generation for MATXS
- MATXST: MATXS-based XS processor for Sn Transport

The XS processing procedure of the MUST code before and after the development of the system is shown in **Fig** 1. The previous procedure should create ISOTXS files using TRANSX code [5]. other hand, by developing the MATXST, the transport table used for the MUST code was directly generated from the MATXS file.

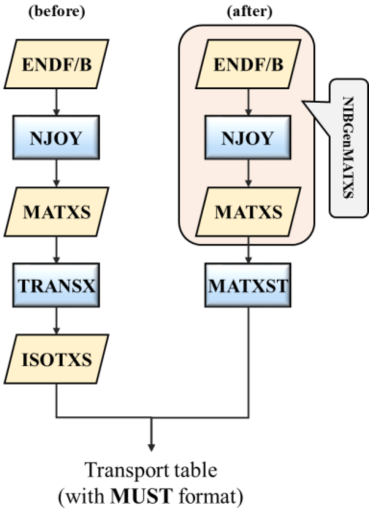


Fig. 1. Procedure of Processing XSection for MUST Code.

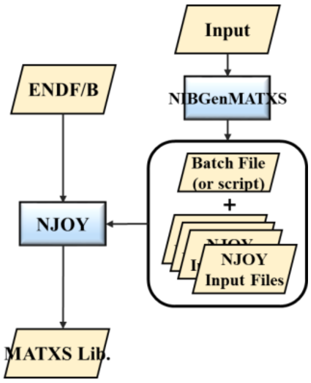


Fig. 2. Flow Chart of NIBGenMATXS Program.

2.1 Automatic MATXS Library Generation

NIBGenMATXS generates NJOY inputs and a batch file for generating MATXS files for a selected set of nuclides. When NIBGenMATXS is executed, it first reads ENDF/B tape and extracts the reaction list required for NJOY GROUPR module. The user does not have to consider the reactions to be written to the GROUPR, and just define temperature, background cross section, group structure and weight function options in the input. **Fig**. 2 shows the procedure of the NIBGenMATXS. The syntax of the NIBGenMATXS input was developed based on the ANJOY program by KAERI [6,7] and NIBGenMATXS was written in C++.

Using the NIBGenMATXS, the MATXS library for MUST code was generated. The present nuclear data processing code and nuclear data used are NJOY21 and the ENDF/B-VIII.0 library [8], respectively. In this paper, this MATXS files were named MUST library.

For the initial test of the programs, the group structure of the MUST library adopted the 27n19g library structure of the ADVANTG code [9], which is widely used in the shielding analysis even if our programs are general in energy group structure. **Tables** I and II show the group structure of neutrons and photons in the library. MUST cross sections were processed at 300, 500, 700 and 1100 K. All nuclides were set to have 10 background cross-sections (1x10^{10}, 5x10^{8}, 5x10^{7}, 5x10^{6}, 5x10^{5}, 5x10^{4}, 5x10^{3}, 5x10^{2}, 50, 5 barns). For the weighting spectrum function of the neutron, a typical function of thermal + 1/E + fission + fusion (IWT=10) was used while for the photon, 1/E + rolloffs weighting function was applied using the (IWT=3) option. MATXS files were created for a total of 547 nuclides, and the size of the entire library is 360 MB.

Table I: Neutron Group Structure

Group	Upper Energy (MeV)	Group	Upper Energy (MeV)
1	2.00000E+01	15	3.05900E-06
2	6.37630E+00	16	1.85540E-06
3	3.01190E+00	17	1.30000E-06
4	1.82680E+00	18	1.12530E-06
5	1.42270E+00	19	1.00000E-06
6	9.07180E-01	20	8.00000E-07
7	4.07620E-01	21	4.13990E-07
8	1.11090E-01	22	3.25000E-07
9	1.50340E-02	23	2.25000E-07
10	3.03540E-03	24	1.00000E-07
11	5.82950E-04	25	5.00000E-08
12	1.01300E-04	26	3.00000E-08
13	2.90230E-05	27	1.00000E-08
14	1.06770E-05	-	1.00000E-11

Table II: Photon Group Structure

Group	Upper Energy (MeV)	Group	Upper Energy (MeV)
1	2.00000E+01	11	1.33000E+00
2	1.00000E+01	12	1.00000E+00
3	8.00000E+00	13	8.00000E-01
4	6.50000E+00	14	6.00000E-01
5	5.00000E+00	15	4.00000E-01
6	4.00000E+00	16	3.00000E-01
7	3.00000E+00	17	2.00000E-01
8	2.50000E+00	18	1.00000E-01
9	2.00000E+00	19	4.50000E-02
10	1.66000E+00	-	1.00000E-02

2.2 Development of MATXS Processing Code

The MATXST code was written in C++ and used only the standard library without using an external library, such as Boost, for easy compilation. **Table** III describes the pseudo code of MATXST. When the code is executed, the user's input and MATXS file are read, and the cross section balance check is performed first. The background cross section (σ_0) of the nuclides in the material is calculated by Bondarenko iteration. After that, cross sections are reconstructed by interpolation. If the user requests, transport correction is applied when constructing the transport table.

Table III: Pseudo Code of MATXST Program

```
read input file
for all nuclide in input
    read matxs file
end for
for all matxs file
    check_balance
end for
for all material
    for all nuclide in a material
        calculate $\sigma_0$ by Bondarenko iteration
    end for
end for
construct transport table with trans-correction
print transport table with MUST format
```

a. Balance check

After reading MATXS file, the first thing to do is to check whether neutron P0 total cross section and the sum of the sub-reactions that make up it are identical.

$$\sigma_{t,0} = \sigma_{el,0} + \sigma_{in} + \sigma_{ftot} + \sigma_{capt} + \sigma_{misc} \qquad (1)$$

where $\sigma_{t,0}$ is P$_0$ total, $\sigma_{el,0}$ is P$_0$ elastic scattering, σ_{in} is inelastic scattering, σ_{ftot} is total fission, σ_{capt} is all kinds of capture, and we defined σ_{misc} as the sum of the reactions that constitute $\sigma_{t,0}$ but do not belong to the previous classification.

A description of the reactions constituting the neutron P$_0$ total cross section and the corresponding Hollerith identifier are listed in **Table** IV. Note that neutron reactions and photon reactions not specified herein are described in the MATXSR section of the NJOY manual.

Table IV: MT Number and Identifier for Neutron Reactions

Symbol	MT	Description	Hollerith Identifier
$\sigma_{t,0}$	1	P0 total	ntot0
$\sigma_{el,0}$	2	P0 elastic scattering	nelas
σ_{in}	4	Inelastic scattering (Sum of the MT=51-91)	ninel
σ_{ftot}	18	total fission (Sum of MT 19, 20, 21 and 38).	nftot
σ_{capt}	102-109 111-117	Sum of capture reactions	ng, np, nd, nt, nh, na, n2a, n3a, n2p, npa, nt2a, nd2a, npd, npt, nda
σ_{misc}	5	Sum of all reactions not given explicitly in another MT number.	nx
	16	(n,2n)	n2n
	17	(n,3n)	n3n
	37	(n,4n)	n4n
	11 22-25 28-30 32-36 41, 42, 44, 45	Production cross section of neutrons + various particles	n2nd nna, nn3a, n2na, n3na, nnp, nn2a, n2n2a, nnd, nnt, nnhe3, nnd2a, nnt2a, n2np, n3np, nn2p, nnpa

There are two reasons for performing a balance check. One is to check whether the nuclear data has been properly made, and the other is to check whether the Hollerith identifier is properly handled.

b. Interpolation scheme

Since cross section data cannot be stored for all temperatures and background cross-sections, the cross section must be reconstructed by applying interpolation except for the values present in the MATXS file.

The MATXST uses a linear interpolation on temperature T for numerical stability. If a value is requested outside the range of values in the library, the program returns the value of the boundary. For example, when 300 K is the minimum temperature of the library, if the user requests a temperature below that, 300 K is returned. In the case of the background cross section (σ_0), the code uses linear interpolation based on $\log(\sigma_0)$.

Fig. 3. shows the result of interpolation using the MATXST. **Fig**. 3(a) shows the interpolation of the P_0 cross section with respect to the temperature of al27, and (b) shows the interpolation of the P_0 cross section of the Pu239 according to the background cross section. It can be observed that the cross section of al27 is affected by temperature in the low energy region, and that pu239 is greatly influenced by the background cross section in the part where the resonance occurs.

c. Transport Correction

Instead of taking into account the higher order terms to capture the effect of anisotropic scattering, in many cases, transport correction is applied to reduce computation time. Although many types of correction techniques have been developed, most of the formulas modify the total scattering to adjust the amount of neutron penetration through the medium. And this correction is applied equally to the with-in group of the scattering matrix to match the balance. The MATXST code provides Diagonal and BHS transport correction methods. If the user wants, correction is not performed, but BHS is currently set as default.

$$\begin{gathered}\tilde{\sigma}_{s,\ell,g\to g} = \sigma_{s,\ell,g\to g} - \left(\sigma_{t,\ell,g} - \sigma_{t,0,g}\right) - \Delta_g \\ \tilde{\sigma}_{t,0,g} = \sigma_{t,0,g} - \Delta_g\end{gathered} \tag{2}$$

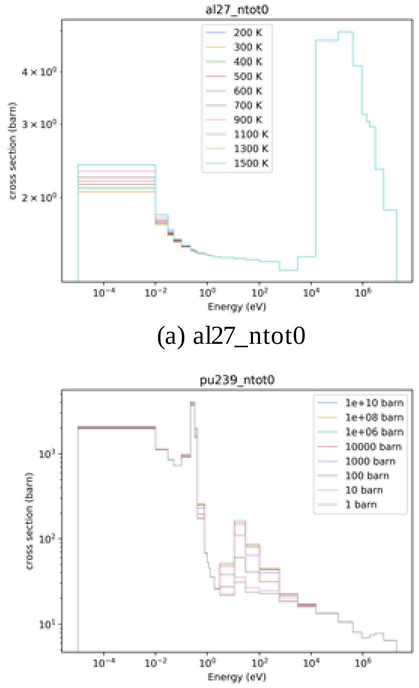
- *Diagonal*

$$\Delta_g = \sigma_{t,0,g} - \sigma_{t,N+1,g} + \sigma_{s,N+1,g\to g} \tag{3}$$

- *Bell-Hansen-Sandmeier (BHS)*

$$\Delta_g = \sigma_{t,0,g} - \sigma_{t,N+1,g} + \sum_{g'} \sigma_{N+1,g\to g'} \tag{4}$$

where g is energy group, Δ_g is transport correction, $\sigma_{t,0,g}$ is P_0 total cross section, $\sigma_{s,\ell,g'\to g}$ is a scattering cross section from g′ to g, $\tilde{\sigma}_{t,0,g}$ is corrected total cross section and $\tilde{\sigma}_{s,\ell,g\to g}$ is corrected scattering cross section.



(a) al27_ntot0

(b) pu239_ntot0

Fig. 3. Cross Sections Reconstructed by the Interpolation function of the MATXST.

d. Photon Production Matrix

MATXS files produced by NJOY code provide a neutron-to-photon production matrix only in infinite-diluted case. The MATXST code uses the self-shielding factor when calculating the production matrix as recommended in the NJOY manual.

3. Validation

3.1 Balance Check

In order to verify that the MATXST code correctly performs Hollerith identifier classification, a balance check was performed on all nuclides of the MATXS library. For this purpose, MUST library generated in this work and two MATXS files produced by KAERI were used. In the minor reactions in the KAERI library, it was found that "mt #" was used as the identifier instead of the proper Hollerith identifier listed in **Table** IV. For example, in the KASHIL-E70 library, reaction MT44 was named as "mt 44" rather than "nn2p". MUST cross sections do not have this case, so it seems to be a problem that occurs depending on the NJOY version. For compatibility with the MATXS file, the code has been modified so that the code can also classify the identifier of the "mt #" format.

The library information and the results of the balance check are shown in **Table** V. All nuclides in KAERI's libraries have passed the check. In MUST code, ag118m failed the balance check, but as a result of analysis, it was determined that it was not an identifier classification problem, so the nuclide was removed from the library.

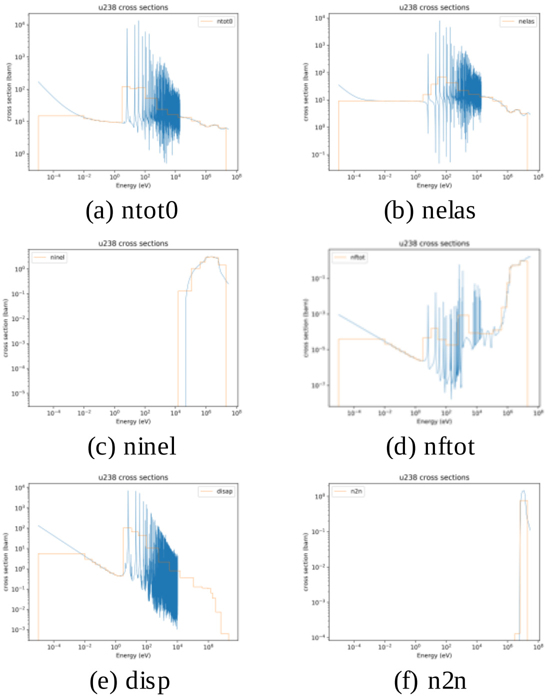
Table V: MATXS Libraries and Balance Check Results

Library	MUST	KASHIL-E70*	KAFAX-E71**
# of nuclide	547	204	144
Group Structure (n/g)	27/19	199/42	150/12
NJOY Code Version	NJOY21	NJOY99.296	NJOY99.396
ENDF/B Version	ENDF/B-VIII.0	ENDF/B-VII.0	ENDF/B-VII.1
Failed Nuclides	ag118m	None	None

* KASHIL-E70 [10], ** KAFAX-E71 [11]

3.2 Comparison with Continuous XS

The verification of MUST cross section created using the NIBGenMATXS was performed by comparing it with the continuous cross section. Since it is difficult to perform validation on all nuclides and reactions, comparisons were performed only on major nuclides such as h1, o16, al27, fe56, u238, pu239. As shown in **Fig**. 4., it can be seen that the MUST cross section of u238 has a similar shape to that of continuous cross section. However, since this approach has limitations in quantitative analysis, we are going to perform rigorous verification by solving the benchmark problem.



(a) ntot0 (b) nelas

(c) ninel (d) nftot

(e) disp (f) n2n

Fig. 4. Comparison of MUST Cross Section and ENDF/B-VIII.0 Continuous Cross Section of u238.

4. Conclusions

In this work, we developed the cross section processing program for MUST code. Using the developed code, the user can easily build the MATXS library, and the transport table required for transport calculation without an additional procedure.

Acknowledgments

This work was supported by the NRF (National Research Foundation of Korea) through Project No. NRF-2019M2D2A1A02057890.

REFERENCES

[1] S.G. Hong, J.W. Kim, and Y. Lee, "Development of MUST (Multi-group Unstructured Geometry S_N Transport) Code", Transaction of the Korean Nuclear Society Autumn Meeting, Gyeongju, Korea, 2009.

[2] S.G. Hong, "Two Subcell Balance Methods for Solving the Multigroup Discrete Ordinates Transport Equation with Tetrahedral Meshes", Nuclear Science and Engineering, 173, 101-107, 2013.

[3] R.E. MacFarlane, A. C. Kahler, and et. al., "The NJOY Nuclear Data Processing System, Version 2016". LA-UR-12-20093, Los Alamos National Laboratory, 2019.

[4] R.E. MacFarlance, and A.C. Kahler, "Methods for Processing ENDF/B-VII with NJOY", Nuclear Data Sheets, 111, 2739-2890, 2010.

[5] R.E. MacFarlane, "TRANSX 2: A Code for Interfacing MATXS Cross-Section Libraries to Nuclear Transport Codes", LA-12312-MS, Los Alamos National Laboratory, 1993.

[6] J.S. Song et al., "Development of Library Processing System for Neutron Transport Calculation", KAERI/RR-2947/2008.

[7] S.G. Hong et al., "Development of Multi-group Gamma Library Processing System for Lattice Transport Calculations", KAERI/RR-3311/2011.

[8] D.A. Brown, M.B. Chadwick, and et. al., "ENDF/B-VIII.0: The 8th Major Release of the Nuclear Reaction Data Library with CIELO-project Cross Sections, New Standards and Thermal Scattering Data", Nuclear Data Sheets, 148, 1-142, 2018.

[9] S.W. Mosher, el. al., "ADVANTG – An Automated Variance Reduction Parameter Generator", ORNL/TM-2013/416, Oak Ridge National Laboratory, 2015.

[10] Do Heon Kim, "Validation of an ENDF/B-VII.0-based Neutron and Photon Shielding Library in MATXS Format", Journal of the Korean Physical Society, 59(2), 1199-1202, 2011.

[11] Do Heon Kim, and el. al., "ZZ KAFAX-E70, 150 and 12 Groups Cross Section Library in MATXS Format based on ENDF/B-VII.0 for Fast Reactors", OECD-NEA, 2008.