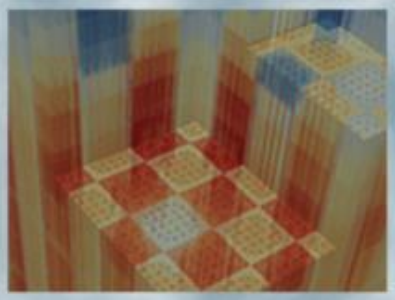
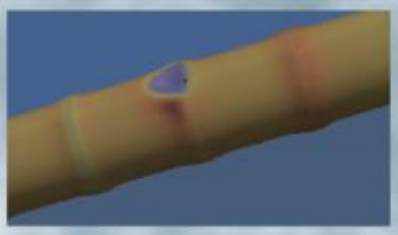
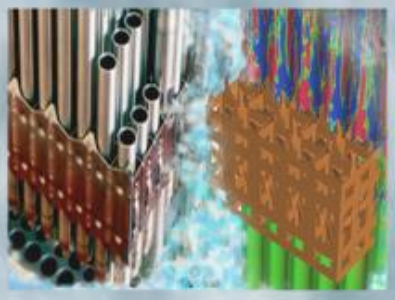
CASL-U-2016-1071-000
ORNL/TM-2016/154

SUBGR: A Program to Generate Subgroup Data for the Subgroup Resonance Self-Shielding Calculation

Revision 0
June 06, 2016

Kang Seog Kim

DOCUMENT AVAILABILITY

Reports produced after January 1, 1996, are generally available free via US Department of Energy (DOE) SciTech Connect.

Website http://www.osti.gov/scitech/

Reports produced before January 1, 1996, may be purchased by members of the public from the following source:

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
Telephone 703-605-6000 (1-800-553-6847)
TDD 703-487-4639
Fax 703-605-6900
E-mail info@ntis.gov
Website http://www.ntis.gov/help/ordermethods.aspx

Reports are available to DOE employees, DOE contractors, Energy Technology Data Exchange representatives, and International Nuclear Information System representatives from the following source:

Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831
Telephone 865-576-8401
Fax 865-576-5728
E-mail reports@osti.gov
Website http://www.osti.gov/contact.html

REVISION LOG

Revision	Date	Affected Pages	Revision Description
0	06/06/2016	All	Initial version

Document pages that are:

Export Controlled None

IP/Proprietary/NDA Controlled None

Sensitive Controlled None

Unlimited ALL PAGES

Requested Distribution:

To: N/A

Copy: N/A

Reviewed by:

Date:

Reviewers: Kevin T. Clarno, Mark L. Williams

EXECUTIVE SUMMARY

The Subgroup Data Generation (SUBGR) program generates subgroup data, including levels and weights from the resonance self-shielded cross section table as a function of background cross section. Depending on the nuclide and the energy range, these subgroup data can be generated by (a) narrow resonance approximation, (b) pointwise flux calculations for homogeneous media; and (c) pointwise flux calculations for heterogeneous lattice cells. The latter two options are performed by the AMPX module IRFFACTOR. These subgroup data are to be used in the Consortium for Advanced Simulation of Light Water Reactors (CASL) neutronic simulator MPACT, for which the primary resonance self-shielding method is the subgroup method.

TABLE OF CONTENTS

REVISION LOG iii
EXECUTIVE SUMMARY iv
LIST OF FIGURES vi
LIST OF TABLES vi
ACRONYMS vii
1. INTRODUCTION 1
2. THEORY 2
2.1 Subgroup Method 2
2.2 Subgroup Data 4
3. PROGRAMMING DETAILS 6
3.1 Algorithm 6
3.2 Description of the SUBGR Subroutines 9
3.3 Overlay Structure 10
3.4 Variable Descriptions 11
3.5 Programming Verification 14
4. INPUT INSTRUCTIONS 15
4.1 Program Run 15
4.2 General Rules of Input Preparation 15
4.3 Abbreviation 16
4.4 Input Card Description 17
4.5 File Structure 19
4.5.1 Structure of the subgroup level library 19
4.5.2 Structure of the RI (XS) table file 21
4.5.3 Structure of the subgroup data file (‘subgr.sub’) 23
5. SAMPLE PROBLEM 26
REFERENCES 28

LIST OF FIGURES

Fig. 2.1. Resonances and subgroups. 3
Fig. 3.1. Calculation flow of SUBGR. 7
Fig. 3.2. Procedure to generate the MPACT MG library 8
Fig. 3.3. Overlay structure of SUBGR. 10
Fig. 4.1. Initial subgroup level library. 20
Fig. 4.2. Initial subgroup level library 22
Fig. 4.3. RI and subgroup data file 25
Fig. 5.1. Sample input of SUBGR. 26
Fig. 5.2. Sample output of SUBGR. 27

LIST OF TABLES

Table 3.1. SUBGR subroutines 9
Table 3.2. Variables for input 11
Table 3.3. Variables for RI (XS) tables 12
Table 3.4. Variables for subgroup data 13
Table 3.5. Unit and regression tests 14
Table 4.1. Input and output files 15
Table 4.2. Definition of groups and blocks 16

ACRONYMS

CASL	Consortium for Advanced Simulation of Light Water Reactors
CMFD	Computational Multi-Fluid Dynamics
IAEA	International Atomic Energy Agency
KNS	Korea Nuclear Society
MG	multigroup (as in cross sections)
ORNL	Oak Ridge National Laboratory
PWR	pressurized water reactor
RI	resonance integral
RMS	root meat square
SUBGR	Subgroup Data Generation program
UKAEA	United Kingdom Atomic Energy Authority
XS	cross section

1. INTRODUCTION

The subgroup method was developed by Khairallah and Recolin [Kha72], Roth [Rot74], and Notari and Garraffo [Not87]. The essence of the subgroup method is that the effective resonance cross sections are approximated by quadrature sets in which the resonances are divided into the subgroup levels and the corresponding subgroup weights are assigned as probabilities.

The Subgroup Data Generation (SUBGR) program generates data including subgroup weights and levels for resonance treatment by using the resonance self-shielded cross section table as a function of the background cross section which could be generated by narrow resonance approximation, as well as homogeneous and heterogeneous IRFFACTORs [Kim15]. Subgroup data are composed of the subgroup levels for the absorption and ν*fission cross sections and the corresponding subgroup weights where ν is the total number of neutrons released from a fission process. These subgroup data are to be used in the Consortium for Advanced Simulation of Light Water Reactors (CASL) neutronic simulator MPACT [Koc15], for which primary resonance self-shielding method is the subgroup method.

This SUBGR report includes an associated theory, a program algorithm, program structure (including subroutine interface and variable description), the user's guide, and the structure of resonance integral and subgroup data file. The objective of this document is to provide directions for using or updating the SUBGR program.

2. THEORY

Reactor physics transport codes, such as MPACT, use a problem-independent cross section library to compute the neutron flux and reaction rate distribution for a large problem. The transport calculation is generally composed of two sequences: resonance processing and eigenvalue transport. The resonance processing portion of the calculation approximates the continuous-energy flux to weight the problem-independent cross sections and provide a problem-dependent cross section library.

2.1 Subgroup Method

Figure 2.1 shows a representative coarse energy group, including three resonances. The effective self-shielded cross section for this group can be obtained by the following flux weighting:

$$\sigma_{x,g} = \frac{\int_{\Delta u_g} \sigma_x(u)\phi(u)du}{\int_{\Delta u_g} \phi(u)du}, \tag{2.1}$$

where $\sigma_{x,g}$ denotes the effective cross section of reaction x at the energy group g, u is lethargy, and $\phi(u)$ is the continuous-energy scalar flux. The continuous-energy scalar flux can be approximated by solving the slowing down problem (a fixed source in the fuel that has a χ spectrum) with the point-wise cross sections. When using a multi-group, rather than subgroup, the SCALE/CENTRM code can solve this problem for a single pincell to generate a problem-dependent multi-group library for SCALE/TRITON, but this is very computationally expensive when many pincells are required. When using an intermediate resonance approximation the multi-group cross sections are a function of a "background cross section".

In the subgroup method, the resonances are divided by the subgroup levels, and the corresponding probability for each subgroup level (σ_{xn}) is defined as the subgroup weight (w_{xn}), as shown in Fig. 2.1 [Hel03]. Therefore, Eq. (2.1) can be approximated (removing the coarse group index for simplicity) as follows:

$$\sigma_x = \frac{\sum_n w_{xn}\sigma_{xn}\phi_n}{\sum_n w_{an}\phi_n}, \tag{2.2}$$

where the summation of the subgroup weights is unity.

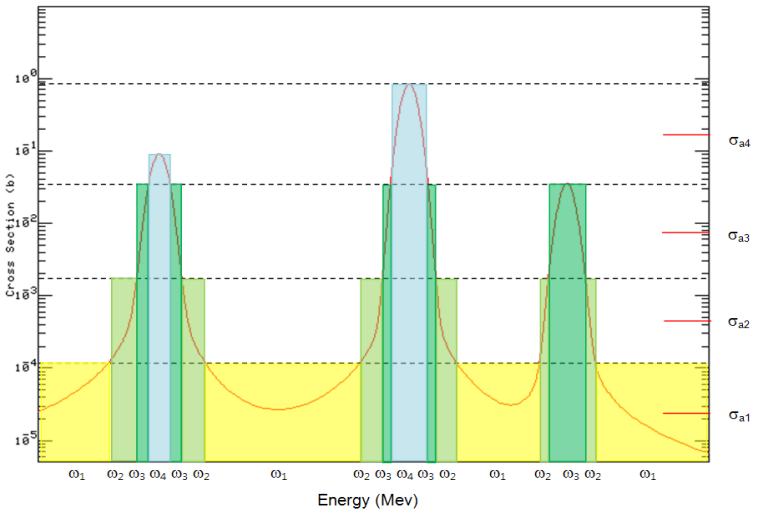


Fig. 2.1. Resonances and subgroups.

In the transport code that uses the subgroup method the final goal of the resonance treatment is to construct a procedure such that the self-shielded cross section estimated by Eq. (2.2) will be identical to the self-shielded cross section estimated by Eq. (2.1). The self-shielded scalar flux ϕ_n in Eq. (2.2) should be estimated from the fixed source transport calculation. If there is only one resonant nuclide, the fixed source transport equation will be

$$\hat{\Omega}\cdot\nabla\psi_n + (N_r\sigma_{an} + \sum_i \lambda_i \Sigma_p^i)\psi_n = \sum_i \lambda_i \Sigma_p^i \,, \tag{2.3}$$

and

$$\phi_n = \int \psi_n d\hat{\Omega}\,, \tag{2.4}$$

where N_r denotes the particle number density for the resonant nuclide, Σ_p is a potential cross section, and ψ_n is an angular flux. Equivalence theory enforces that the self-shielded scalar flux is expressed with the absorption (σ_{an}) and the background cross sections (σ_{bn}).

$$\phi_n = \frac{\sigma_{bn}}{\sigma_{an} + \sigma_{bn}}\,. \tag{2.5}$$

Typically the background cross section in the heterogeneous model is divided into two parts, as follows:

$$\sigma_{bn} = \lambda\sigma_p + \sigma_{en}(\sigma_{an})\,, \tag{2.6}$$

where λ denotes an intermediate resonance parameter, σ_p is a potential cross section, and σ_{en} is a parameter called the equivalence cross section which accounts for heterogeneous self-shielding effects. By using Eq. (2.5), Eq. (2.2) can be written as follows:

$$\sigma_x = \frac{\sum_n w_{xn}\sigma_{xn}\dfrac{\sigma_{bn}}{\sigma_{an}+\sigma_{bn}}}{1-\sum_n w_{an}\dfrac{\sigma_{an}}{\sigma_{an}+\sigma_{bn}}}. \tag{2.7}$$

Therefore, if the subgroup levels and weights are given for a certain nuclide, the effective self-shielded cross section can be obtained by estimating the corresponding background cross sections (σ_{bn}) in the lattice calculation. Since the equivalence cross section is not sensitive to the absorption cross section but to the geometrical configuration, σ_{bn} is often approximated by a single background cross section σ_b. [Hel03] Eq. (2.7) can be written as follows:

$$\sigma_x = \frac{\sum_n w_{xn}\sigma_{xn}\dfrac{\sigma_b}{\sigma_{an}+\sigma_b}}{1-\sum_n w_{an}\dfrac{\sigma_{an}}{\sigma_{an}+\sigma_b}}. \tag{2.8}$$

If there are several resonant nuclides, there will be a resonant interference between the resonant nuclides. Since this interference will influence the self-shielded scalar flux, Eq. (2.8) can be written as follows:

$$\sigma_x = \frac{\sum_n w_{xn}\sigma_{xn}\dfrac{\sigma_b}{\sigma_{an}+\hat{\sigma}_a+\sigma_b}}{1-\sum_n w_{an}\dfrac{\sigma_{an}}{\sigma_{an}+\hat{\sigma}_a+\sigma_b}}, \tag{2.9}$$

where $\hat{\sigma}_a$ is the total absorption cross section of the other resonant nuclides. The effective self-shielded cross sections are estimated iteratively.

2.2 Subgroup Data

The integrated resonance integral (RI_x) is defined as the numerator of Eq. (2.1), and the resonance integral divided by lethargy width is also called resonance integral (R_x). The resonance integral can be written by using the subgroup weights and levels from Eq. (2.7) as follows:

$$R_x = \sum_n w_n\sigma_{xn}\frac{\sigma_{bn}}{\sigma_{an}+\sigma_{bn}} \approx \sum_n w_n\sigma_{xn}\frac{\sigma_b}{\sigma_{an}+\sigma_b}. \tag{2.10}$$

Since the denominator of Eq. (2.1) can be understood as a scalar flux (ϕ) for the coarse energy group, Eq. (2.1) can be rewritten as follows:

$$\sigma_x = \frac{R_x}{\phi}. \tag{2.11}$$

As described in the "Development of a New 47-Group Library for the CASL Neutronics Simulators" [Kim15], many CENTRM pincell problems are defined with different constituent compositions and the geometrical configurations to span the anticipated range within the problem. Each CENTRM solution provides a resonance integral for each reaction ("x"), background cross section, and flux. With many pincells, a table of resonance integrals as a function of the background cross section can be defined. Index '*k*' for the variation cases can be added to Eq. (2.11).

$$\sigma_{x,k} = \frac{R_{x,k}}{\phi_k} \quad (k = 1, 2, \ldots, K). \tag{2.12}$$

The resonance integrals ($R_{x,k}$) can be obtained from the self-shielded cross sections and the corresponding background cross sections for the variation cases. Assuming there are more pincells (K) than subgroups N), the subgroup weights, wn, can be obtained from Eq. (2.10) by the least squares fitting to minimize the difference between the original resonance integral and the reconstructed resonance integral.

Subgroup levels are arbitrarily given, and the corresponding subgroup weights are obtained by minimizing the following function *f*.

$$f(w_1, \ldots, w_K) = \sum_k \frac{1}{R_{x,k}^2} \left(R_{x,k} - \sum_n w_{xn}\sigma_{xn} \frac{\sigma_{bn,k}}{\sigma_{an} + \sigma_{bn,k}}\right)^2. \tag{2.13}$$

$$f(w_1, \ldots, w_K) \cong \sum_k \frac{1}{R_{x,k}^2} \left(R_{x,k} - \sum_n w_{xn}\sigma_{xn} \frac{\sigma_{b,k}}{\sigma_{an} + \sigma_{b,k}}\right)^2. \tag{2.14}$$

Subgroup levels are to be automatically adjusted to minimize the relative squared-difference between the original resonance integral and the reconstructed resonance integral using the subgroup levels and weights. The resonance integral table from the IRFFACTOR calculation includes the background cross sections as a function of the absorption cross section at each variation *k* to cover all the subgroup levels. Although the subgroup levels are varied at iterations, the corresponding background cross section ($\sigma_{bn,k}$) can be obtained through the interpolation of the given table. SUBGR includes two options to generate the subgroup weights and levels by using Eqs. (2.13) and (2.14), which use the level dependent and constant background cross sections, respectively.

When the resonance interference is neglected, errors in estimating the effective self-shielded cross section result mainly from the subgroup levels and weights themselves, as well as the scalar flux estimation. In the real application, the self-shielded scalar flux is estimated by the following equation.

$$\hat{\phi}_k = 1 - \sum_n w_{an} \frac{\sigma_{an}}{\sigma_{an} + \sigma_{bn,k}}, \tag{2.15}$$

where the subgroup weights and levels are given, and the corresponding background cross sections ($\sigma_{bn,k}$) are obtained by the fixed source transport calculations of the transport lattice code using Eq. (2.3). The difference between the scalar flux (ϕ_k) in Eq. (2.12) and the scalar flux ($\hat{\phi}_k$) in Eq. (2.15) causes a difference in the effective self-shielded cross section.

In order to remove this error, a new method [Joo09] has been proposed recently to generate the subgroup weights. In the new method, the subgroup weights are to be estimated to conserve the self-shielded cross sections, as follows:

$$f(w_1, \ldots, w_K) = \sum_k \frac{1}{R_{x,k}^2} (\hat{R}_{x,k} - \sum_n w_{xn} \sigma_{xn} \frac{\sigma_{bn,k}}{\sigma_{an} + \sigma_{bn,k}})^2, \tag{2.16}$$

where

$$\hat{R}_{x,k} = \sigma_{x,k} \hat{\phi}_k = \sigma_{x,k} (1 - \sum_n w_{an} \frac{\sigma_{an}}{\sigma_{an} + \sigma_{bn,k}}). \tag{2.17}$$

3. PROGRAMMING DETAILS

3.1 Algorithm

Figure 3.1 shows the algorithm for the SUBGR code to generate the subgroup data, including subgroup levels and weights. The resonance integral and subgroup data file “subgr.sub” will be used in constructing the MPACT multigroup (MG) library by the DECLIB, as shown in Fig. 3.2.

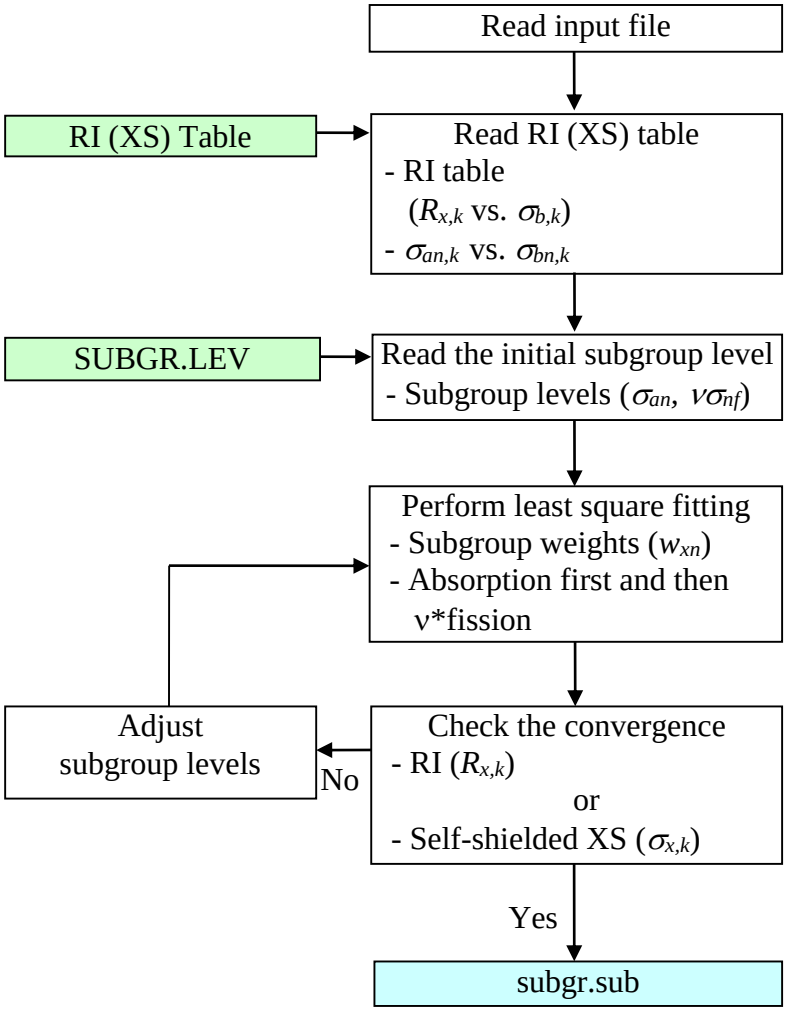


Fig. 3.1. Calculation flow of SUBGR.

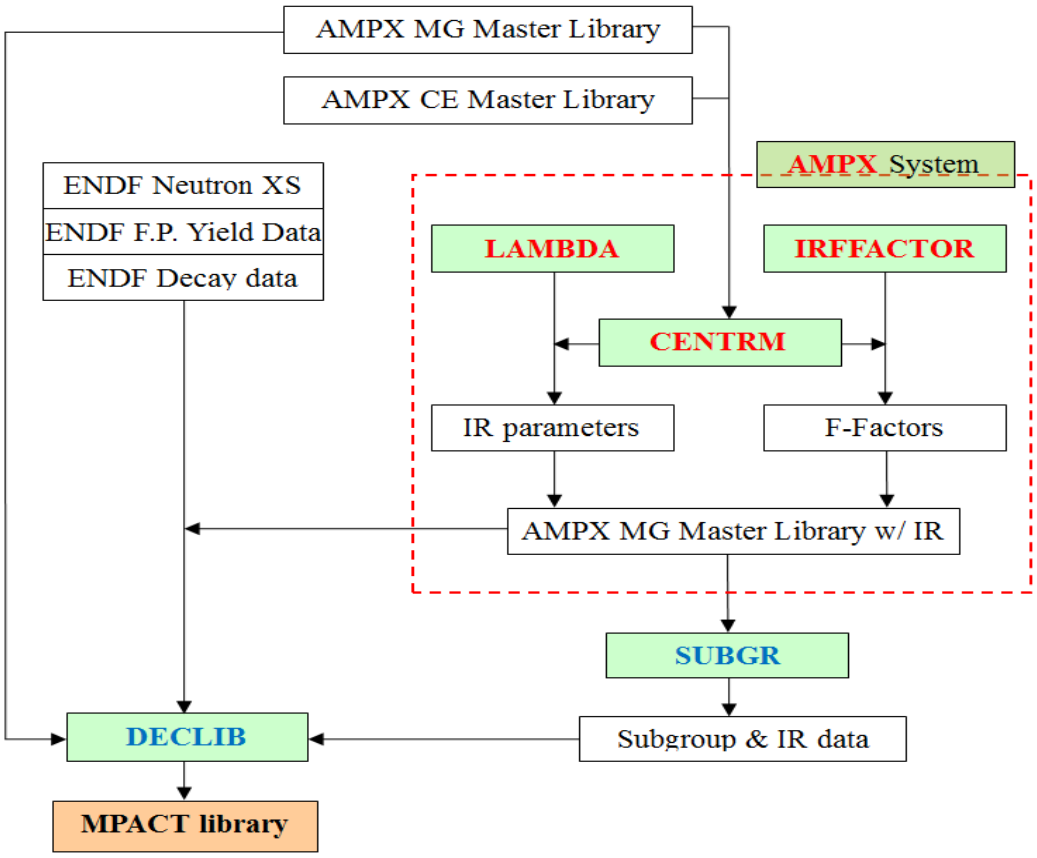


Fig. 3.2. Procedure to generate the MPACT MG library.

3.2 Description of the SUBGR Subroutines

Table 3.1 shows the subroutines in SUBGR.

Table 3.1. SUBGR subroutines

No.	File	Subroutines	Description
1	cubspline.f90	cubspline	Cubic spline from numerical recipe
		spline	Cubic spline from recipe
2	error.f90	error	Prints out error message
		warning	Prints out warning message
3	fitting0.f90	fitting0	Least square fitting based on RI conservation
		fitting2	Least square fitting based on XS conservation
		inverse	Inverse matrix from ‘numerical recipe’
		ludcmp	Implicit pivoting for inverse matrix from ‘numerical recipe’
		lubksb	Forward/backward substitution for inverse matrix from ‘numerical recipe’
4	levsigb.f90	levsigb	Obtains level dependent background XS
5	lsqfit.f90	lsqfit	Automatic search for subgroup levels and weights by least square fitting
6	param.f90	param	All variables and parameters are defined in this module
7	rdsublev.f90	rdsublev	Reads initial subgroup level file
8	readin.f90	readin	Reads input file
9	readri.f90	readri	Reads resonance integral table file
10	segev.f90	segev	Segev interpolation for the resonance integrals
11	subgr.f90	subgr	Program to generate subgroup weights from RI table
12	timedate.f90	timedate	Time and date
13	upsublev.f90	upsublev	Update subgroup level file
14	wrout.f90	wrout	Writes out RI table and subgroup data

3.3 Overlay Structure

Figure 3.3 shows the overlay structure of SUBGR.

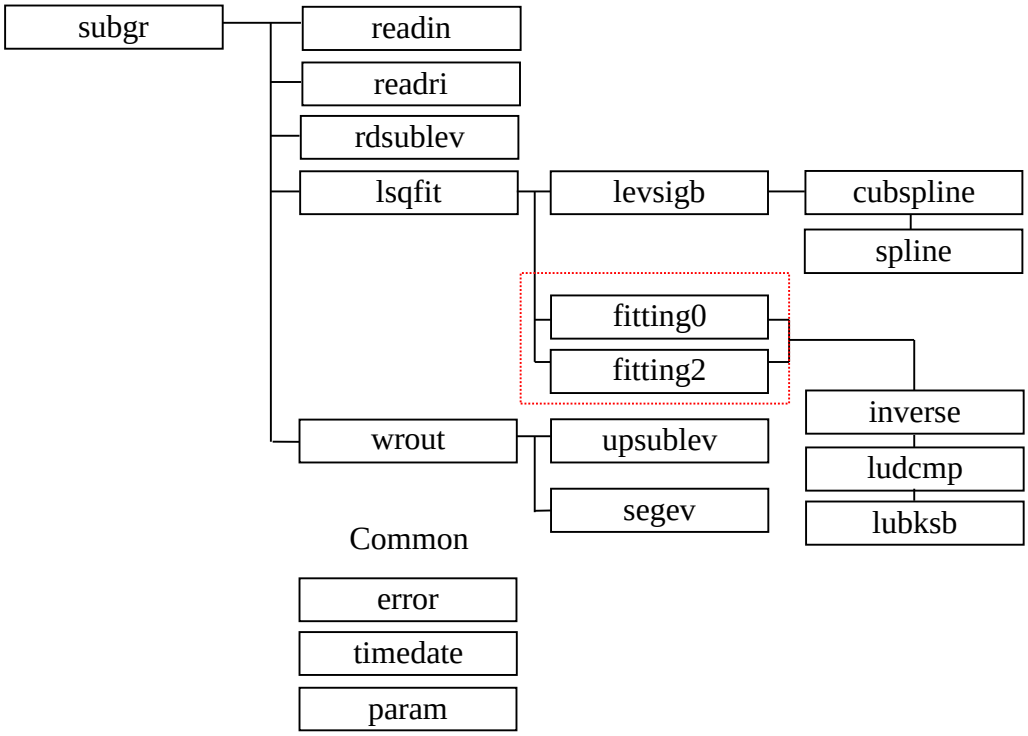


Fig. 3.3. Overlay structure of SUBGR.

3.4 Variable Descriptions

Tables 3.2–3.4 provide variables used in SUBGR. Table 3.2 includes the variables associated with the user input. Table 3.3 provides variables related to the resonance integral table and the initial subgroup level libraries. Table 3.4 provides the variables related with the generation of the subgroup data.

Table 3.2. Variables for input

Variables	Type	Description
titcard	A300	Title card
rilib	A100	RI table library filename
slevlib	A100	Subgroup level library filename
ioptsigz	I	Option for the background XS 0 : constant (RI conserved) 1 : level dependent (RI conserved) 2 : level dependent (XS conserved)
ioptri	I	Option for data type 0 : RI 2 : XS
igresb0	I	The starting resonance group
igrese0	I	The ending resonance group
nuclid(i)	I	Numeric nuclide id. no. (i=1,nnuc)
ioptxs(i)	I	XS type option (1=sig-a; 2=sig-a + nu*sig-f) (i=1,nnuc)
conv1	R	Convergence criteria #1 (absolute root mean square [RMS] error)
conv2	R	Convergence criteria #2 (relative RMS error)
conv3	R	Convergence criteria #3 (maximum difference)

Table 3.3. Variables for RI (XS) tables

Variables	Type	Description
nnuc	I	The number of nuclides in the RI library
norg	I	The number of groups in the RI table
nsigz	I	The number of background XS
ntemp	I	The number of temperatures
igresb	I	The starting resonance group
igrese	I	The ending resonance group
labsxs	L	Logical for the existence of sig-a
lnfixs	L	Logical for the existence of nu*sig-f
lsctxs	L	Logical for the existence of sig-s
ioptri	I	Option for data type (0/2: RI/XS)
nflx0	I	The number of sig-a levels
norgx	I	The number of groups for nu
nsig	I	The number of XS types
nsubs	I	The number of subgroups
numsub(i)	I	The number of subgroups (i=1,numlev)
sublev0(i,ig,j)	R	Reference subgroup levels (i=1,numlev; ig=1,norg; j=1,numsub)
sublev(i,j)	R	Reference subgroup levels (i=1,numlev; j=1,numsub)
nu(ig)	R	The number of released neutrons per fission (ig=1,norgx)
temp(i)	R	Temperature (K) (i=1,ntemp)
sigz(i,j,k)	R	Background XS (i=1,ntemp; j=1,norg; k=1,nsigz)
ria(i,j,k)	R	Reference RI-absorption/du (i=1,ntemp; j=1,norg; k=1,nsigz)
ris(i,j,k)	R	Reference RI-scattering/du (i=1,ntemp; j=1,norg; k=1,nsigz)
rif(i,j,k)	R	Reference RI-nu*fission/du (i=1,ntemp; j=1,norg; k=1,nsigz)
abslev0(i)	R	Absorption XS levels (i=1,nflx0)
backxs0(i,ig,j,k)	R	Background XS (i=1,ntemp; ig=iresb,irese; j=1,nsigz; m=1,nflx0)

Table 3.4. Variables for subgroup data

Variables	Type	Description
rmse	R	Root mean square error
xdat(i)	R	Temporary data points (i=1,nsigz)
subl(i)	R	Temporary subgroup levels (i=1,7)
rixx(i)	R	Temporary RI's (i=1,nsigz)
xsgxx1(i)	R	Temporary subgroup levels (i=1,7)
xsgxx2(i)	R	Temporary subgroup levels (i=1,7)
wgtxx(i)	R	Temporary subgroup weights (i=1,7)
wgtx2(i,j)	R	Temporary subgroup weights (i=1,nteml; j=1,7)
xsga(i,j,k)	R	Subgroup level for absorption (i=1,numlev; j=1,norg; k=1,numsub)
xsgf(i,j,k)	R	Subgroup level for nu*fission (i=1,numlev; j=1,norg; k=1,numsub)
wgta(i,j,k,l)	R	Subgroup weight for absorption (i=1,numlev; j=1,ntemp; k=1,norg; l=1,numsub)
wgtf(i,j,k,l)	R	Subgroup weight for nu*fission (i=1,numlev; j=1,ntemp; k=1,norg; l=1,numsub)
riax(i,j,k)	R	RI-absorption/du from subgroup data (i=1,ntemp; j=1,norg; k=1,nsigz)
rifx(i,j,k)	R	RI-nu*fission/du from subgroup data (i=1,ntemp; j=1,norg; k=1,nsigz)
risx(i,j,k)	R	RI-scattering/du from subgroup data (i=1,ntemp; j=1,norg; k=1,nsigz)
tt1(i)	R	Temporary subgroup levels (i=1,7)
tt2(i)	R	Temporary subgroup levels (i=1,7)
rr2(i)	R	Temporary RI's (i=1,maxz)
sigz0(i,j)	R	Level dependent background XS (i=1,nsigz; j=1,7)
xdat0(i)	R	Temporary data points (i=1,nsigz)
lxstype	L	Cross section type (t/f: abs/nu*fis)
flux0(i)	R	Flux(denominator) (i=1,nsigz)

3.5 Programming Verification

Table 3.5 provides unit and regression tests for the SUBGR programming verification. In addition, the SUBGR program internally includes a verification procedure by comparing the reconstructed RIs or XSs from subgroup data to the original RIs or XSs.

Table 3.5. Unit and regression tests

Test type	Test name	Description
Unit	testSpline	Test for cubic spline in cubspline.f90
	testInverse	Test for inverse matrix in cubspline.f90
	testFitting0	Test for RI least square fitting in fitting0.f90
	testFitting2	Test for XS least square fitting in fitting0.f90
	testSegev	Test for Segev interpolation for segev.f90
Regression		Generates 47-group subgroup data for ^{91}Zr, ^{235}U and ^{238}U

4. INPUT INSTRUCTIONS

4.1 Program Run

Table 4.1 shows input and output files treated in SUBGR. The SUBGR program can be executed by typing

subgr.exe

where the input file "subgr.in" should be in the same directory.

Table 4.1. Input and output files

File description	Filename	File	Status
Input file	subgr.in	1	Required
Output file	subgr.out	2	
Resonance integral (RI) table (Segev interpolated) or Self-shielded XS table (not interpolated)		3	Required
RI and subgroup data		4	
Initial subgroup level library	Name?	11	Required
Input file		12	Scratch
Updated initial subgroup level library	subgr.lev	13	

4.2 General Rules of Input Preparation

Main input is composed of groups and constituent blocks. Groups are defined by % followed by four characters, and blocks are defined by three characters as shown in Table 4.2. The general instructions to prepare the input are as follows:

(a) Comment (!)
Bang (!) is used to designate a comment and can be placed anywhere in the line. If "!" is placed on the first column or if there are only blanks prior to "!," then this line will be treated as a comment, and everything followed by "!" will be treated as a comment.

(b) Blank line
A blank line can be placed anywhere, and it will be neglected.

(c) %FINE
This group should be included in the input deck. All the input lines followed by "%FINE" are neglected.

(d) Upper and lower case (no distinction)
Both upper and lower case can be used. The program automatically transfers lower case characters into the upper case characters except for the filenames.

(e) Line limit (300 characters per line)
The maximum number of characters per line is 300 for actual input parameters, but for comments there is no character limit.

Table 4.2. Definition of groups and blocks

Group	Block	Description
TITL		Job title
IOPT	IOP	Calculation option
	GRP	Groups to be treated
	CON	Convergence criteria
FILE	SUB	Initial subgroup level library
	RIT	RI table library
RESO	NUC	Nuclide information
FINE		End of the job

4.3 Abbreviation

For convenience, the following abbreviations are used.

(1) Abbreviations for comment or default value
   a. (C) - comment
   b. (D) - default

(2) Abbreviations for variable formats:
   a. I - integer
   b. R - real
   c. A - character
   d. {variable} - variable that can be omitted

4.4 Input Card Description

%TITL
[TITCARD]

TITCARD	A300	Job title (D) None

Example:

```
%TITL
  SUBGROUP DATA GENERATION FOR     72-Hf-178
```

%IOPT
['IOP', IOPTSIGZ]
{['GRP', IGRESB0, IGRESE0]}
{['CON', CONV1, CONV2, CONV3]}

IOPTSIGZ	I	Calculation option 0 : Constant background xs (RI is conserved.) 1 : Level-dependent background xs (RI is conserved.) 2 : Level-dependent background xs (Cross section is conserved.) (D) None
IGRESB0	I	Starting energy group (D) Read from the RI table
IGRESE0	I	Ending energy group (D) Read from the RI table
CONV1	I	Convergence criteria for absolute RMS error (D) CONV1=0.05
CONV2	I	Convergence criteria for relative RMS error (D) CONV2=0.005
CONV3	I	Convergence criteria for maximum difference (D) CONV3=0.005

Example:

```
%IOPT
  IOP  2    !Level dependent background xs
```

%FILE
['SUB', SLEVLIB]
['RIT', RILIB]

SLEVLIB	A100	Subgroup level library filename (D) None
RILIB	A100	RI table library filename (D) None

Example:

```
%FILE
   sub  ..\lib\subgr.lev
   rit  ..\lib\rilamb.rit
```

%RESO
['NUC', NX, NUCLID(I), IOPTRI(I) , IOPTXS(I)]$_{I=1,NNUC}$

NX	I	Nuclide ordering number (D) None
NUCLID	I	Nuclide identification for the resonant nuclide (D) None
IOPTRI	I	RI/XS data option 0 : RI 2 : XS + Level-dependent background XS (D) None
IOPTXS	I	XS type option 1 : Absorption xs 2 : Absorption xs + Nu*fission xs (D) None
NNUC	I	The number of nuclides (D) None

Example:

```
%RESO
    nuc  1  42095    0   1
    nuc  2  43099    0   1
    nuc  3  46108    0   1
    nuc  4  47107    0   1
    nuc  5  47109    0   1
    nuc  9  92235    2   2
```

4.5 File Structure

In this section, the structures for the subgroup level library and the resonance integral/subgroup data files are shown.

4.5.1 Structure of the subgroup level library

A sample of the subgroup level library is provided in Fig. 4.1.

[Structure]

```
$TIT:
      [subtitle]
$DIM:
      [nnucx0, numsub(1), numsub(2)]
$NUC:      [inx]
   [nuclx]
   ['L7A', ig, sublev0(1,ig,k) k=1,numsub(1) ]
   ['L4A', ig, sublev0(2,ig,k) k=1,numsub(2) ]
$END:
**Repeat $NUC:
```

[Variable Description]

subtitl	A	title of library
nnucx0	I	number of nuclides
numsub(1)	I	number of subgroup levels (first)
numsub(2)	I	number of subgroup levels (second)
nuclx	I	nuclide ID
ig	I	energy group: If ig=0, this data will be applied to the other groups for which ig is not defined.
L7A	A3	indicator for the first subgroup
L4A	A3	indicator for the second subgroup
sublev0	R	subgroup level

```
$TIT:
  AMPX 47-G LIBRARY
$DIM:
      54      7      4
$NUC:    0
  00000
  L7A    0  -2.0  2.0  4.800000E+00  4.000000E+01  2.500000E+02  7.500000E+02  2.000000E+03  7.000000E+03  2.000000E+04
  L4A    0  -5.0  5.0  1.000000E+01  5.000000E+02  2.000000E+03  1.000000E+04
$NUC:    1
  40090
  L7A    0  -2.0  2.0  4.800000E+00  4.000000E+01  2.500000E+02  7.500000E+02  2.000000E+03  7.000000E+03  2.000000E+04
  L4A    0  -5.0  5.0  1.000000E+01  5.000000E+02  2.000000E+03  1.000000E+04
$NUC:    2
  40091
  L7A    0  -2.0  2.0  4.800000E+00  4.000000E+01  2.500000E+02  7.500000E+02  2.000000E+03  7.000000E+03  2.000000E+04
  L4A    0  -5.0  5.0  1.000000E+01  5.000000E+02  2.000000E+03  1.000000E+04
$NUC:    3
  40092
  L7A    0  -2.0  2.0  4.800000E+00  4.000000E+01  2.500000E+02  7.500000E+02  2.000000E+03  7.000000E+03  2.000000E+04
  L4A    0  -5.0  5.0  1.000000E+01  5.000000E+02  2.000000E+03  1.000000E+04

  … (Skipped)

$NUC:    4
  40094
  L7A    0  -2.0  2.0  4.800000E+00  4.000000E+01  2.500000E+02  7.500000E+02  2.000000E+03  7.000000E+03  2.000000E+04
  L4A    0  -5.0  5.0  1.000000E+01  5.000000E+02  2.000000E+03  1.000000E+04
$NUC:   21
  63155
  L7A    0  -2.0  2.0  4.800000E+00  4.000000E+01  2.500000E+02  7.500000E+02  2.000000E+03  7.000000E+03  2.000000E+04
  L4A    0  -5.0  5.0  1.000000E+01  5.000000E+02  2.000000E+03  1.000000E+04
  L7A   35  -2.0  2.0  4.800000E+00  5.000000E+01  3.500000E+02  7.500000E+02  2.000000E+03  7.000000E+03  2.000000E+04
$NUC:   22
  64155
  L7A    0  -2.0  2.0  4.800000E+00  4.000000E+01  2.500000E+02  7.500000E+02  2.000000E+03  7.000000E+03  2.000000E+04
  L4A    0  -5.0  5.0  1.000000E+01  5.000000E+02  2.000000E+03  1.000000E+04
  L7A   44  -2.0  2.0  4.800000E+00  5.000000E+01  3.500000E+02  7.500000E+02  2.000000E+03  7.000000E+03  2.000000E+04

  … (Skipped)

$NUC:   52
  74183
  L7A    0  -2.0  2.0  4.800000E+00  4.000000E+01  2.500000E+02  7.500000E+02  2.000000E+03  7.000000E+03  2.000000E+04
  L4A    0  -5.0  5.0  1.000000E+01  5.000000E+02  2.000000E+03  1.000000E+04
$NUC:   53
  74184
  L7A    0  -2.0  2.0  4.800000E+00  4.000000E+01  2.500000E+02  7.500000E+02  2.000000E+03  7.000000E+03  2.000000E+04
  L4A    0  -5.0  5.0  1.000000E+01  5.000000E+02  2.000000E+03  1.000000E+04
$NUC:   54
  74186
  L7A    0  -2.0  2.0  4.800000E+00  4.000000E+01  2.500000E+02  7.500000E+02  2.000000E+03  7.000000E+03  2.000000E+04
  L4A    0  -5.0  5.0  1.000000E+01  5.000000E+02  2.000000E+03  1.000000E+04
$END:
```

Fig. 4.1. Initial subgroup level library.

4.5.2 Structure of the RI (XS) table file

Figure 4.2 provides a sample of the RI (XS) table file to be used in SUBGR as an input.

[Structure]

```
$DIM:
     [nnucx]
$NUC:  [inx]
  [nuclide,nsigz,ntemp,igresb,igrese,labsxs,lsctxs,lnfixs,ioptri,nflx0]
  [temp(it)]it=1,ntemp
if (nflx0=0){
   [sigzw0(j)]j=1,nsigz
elseif (nflx0>0)
   [abslev0(m)]m=1,nflx0
}
if (labsxs){
 RI-A :
 do ig=igresb,igrese
   do it=1,ntemp
     [ig,it],[ria(it,ig,j)j=1,nsigz]
   enddo
 enddo
}
if (lsctxs){
 RI-S :
 do ig=igresb,igrese
   do it=1,ntemp
     [ig,it],[ris(it,ig,j)j=1,nsigz]
   enddo
 enddo
}
if (lnfixs){
 RI-NF:
 do ig=igresb,igrese
   do it=1,ntemp
     [ig,it],[rinf(it,ig,j)j=1,nsigz]
   enddo
 enddo
}
if (ioptri>0){
 BK-XS:
 do ig=igresb,igrese
   do it=1,ntempx
     [ig,it],[sigz(it,ig,iz)iz=1,nsigz]
   enddo
 enddo
 BK-SG:
 do ig=igresb,igrese
   do it=1,ntempx
     do im=1,nflx0
       [ig,it,im],[backxs0(it,ig,iz,im)iz=1,nsigz]
     enddo
   enddo
 enddo
  }
```

```
**Repeat $NUC:
$END:
```

[Variable Description]

Table 3.3 provides a detailed description for the variables.

```
$NUC:      409
         92235     16      5      1     47      T      T      T      2     10
  2.930000E+02  6.000000E+02  9.000000E+02  1.200000E+03  2.400000E+03
  1.000000E+00  5.000000E+00  1.000000E+01  5.000000E+01  1.000000E+02  5.000000E+02  1.000000E+03  5.000000E+03  …
 RI-A :
      1      1  1.523837E+00  1.519290E+00  1.517347E+00  1.516621E+00  1.516297E+00  1.515799E+00  1.515975E+00  …
      1      2  1.524338E+00  1.519163E+00  1.517809E+00  1.517313E+00  1.516058E+00  1.516293E+00  1.515483E+00
      1      3  1.523790E+00  1.519053E+00  1.517405E+00  1.517133E+00  1.515861E+00  1.516027E+00  1.516287E+00
      1      4  1.524322E+00  1.519626E+00  1.516991E+00  1.516807E+00  1.516285E+00  1.516188E+00  1.515273E+00
      1      5  1.524267E+00  1.518931E+00  1.517904E+00  1.517019E+00  1.516199E+00  1.515811E+00  1.516058E+00
 …
     47      3  1.297180E+03  1.303678E+03  1.309205E+03  1.311947E+03  1.313793E+03  1.317516E+03  1.319853E+03  …
     47      4  1.297516E+03  1.303878E+03  1.309326E+03  1.312032E+03  1.313854E+03  1.317528E+03  1.319838E+03
     47      5  1.297220E+03  1.303178E+03  1.308354E+03  1.310908E+03  1.312625E+03  1.316097E+03  1.318320E+03
 RI-S :
      1      1  5.016735E+00  5.032171E+00  5.036129E+00  5.038506E+00  5.040006E+00  5.042933E+00  5.039018E+00  …
      1      2  5.017522E+00  5.031521E+00  5.035741E+00  5.038568E+00  5.040159E+00  5.041872E+00  5.038142E+00
…
 RI-NF:
      1      1  5.247211E+00  5.231492E+00  5.224781E+00  5.222270E+00  5.221145E+00  5.219431E+00  5.220035E+00  …
      1      2  5.248942E+00  5.231054E+00  5.226373E+00  5.224655E+00  5.220326E+00  5.221132E+00  5.218342E+00
…
 BK-XS:
…
     18      1  3.662899E+02  6.881702E+02  1.036605E+03  1.243842E+03  1.404358E+03  1.807808E+03  2.094512E+03  …
     18      2  3.663022E+02  6.881934E+02  1.036656E+03  1.243920E+03  1.404462E+03  1.807979E+03  2.094744E+03
     18      3  3.663140E+02  6.882180E+02  1.036713E+03  1.244007E+03  1.404576E+03  1.808170E+03  2.095005E+03
     18      4  3.663263E+02  6.882448E+02  1.036775E+03  1.244103E+03  1.404704E+03  1.808383E+03  2.095298E+03
     18      5  3.663796E+02  6.883600E+02  1.037042E+03  1.244521E+03  1.405255E+03  1.809308E+03  2.096566E+03
…
 BK-SG:
…
   18    1   1  3.660918E+02  6.877074E+02  1.035581E+03  1.242293E+03  1.402341E+03  1.804479E+03  2.089993E+03  …
   18    1   2  3.661816E+02  6.879431E+02  1.036102E+03  1.243077E+03  1.403361E+03  1.806161E+03  2.092274E+03
   18    1   3  3.662994E+02  6.881945E+02  1.036663E+03  1.243930E+03  1.404476E+03  1.808005E+03  2.094781E+03
   18    1   4  3.668443E+02  6.891593E+02  1.038949E+03  1.247698E+03  1.409540E+03  1.816444E+03  2.106492E+03
   18    1   5  3.671524E+02  6.892780E+02  1.039484E+03  1.248972E+03  1.411515E+03  1.819938E+03  2.111568E+03
   18    1   6  3.675709E+02  6.851768E+02  1.031052E+03  1.239566E+03  1.401452E+03  1.804184E+03  2.092661E+03
   18    1   7  3.675773E+02  6.804612E+02  1.019135E+03  1.223371E+03  1.381615E+03  1.771014E+03  2.049553E+03
   18    1   8  3.673929E+02  6.639514E+02  9.662784E+02  1.142143E+03  1.275313E+03  1.584799E+03  1.801002E+03
   18    1   9  3.672999E+02  6.574722E+02  9.437760E+02  1.106232E+03  1.227333E+03  1.498689E+03  1.686483E+03
   18    1  10  3.672324E+02  6.531862E+02  9.291186E+02  1.083020E+03  1.196505E+03  1.444686E+03  1.615465E+03
…
$NUC:      410
         94239     16      5      1     47      T      T      T      0      0
  2.930000E+02  6.000000E+02  9.000000E+02  1.200000E+03  2.400000E+03
```

Fig. 4.2. Initial subgroup level library

4.5.3 Structure of the subgroup data file ('subgr.sub')

Figure 4.3 provides a sample of the subgroup data file from the SUBGR run.

[Structure]

```
$DIM:
  [nnuc]
$NUC:   [inc]
  [nuclide,nsigz,ntemp,igresb,igrese,labsxs,lsctxs,lnfixs,ioptxs,numlev]
  [temp(it)it=1,ntemp]
  [sigz(iz)iz=1,nsigz]
if (labsxs){
 RI-A :
 do ig=igresb,igrese
   do it=1,ntemp
     [ig,it,ria(it,ig,iz)iz=1,nsigz]
   enddo
 enddo
}
if (lsctxs){
 RI-S :
 do ig=igresb,igrese
   do it=1,ntemp
     [ig,it,ris(it,ig,iz)iz=1,nsigz]
   enddo
 enddo
}
if (lnfixs){
 RI-NF:
 do ig=igresb,igrese
   do it=1,ntemp
     [ig,it,rif(it,ig,iz)iz=1,nsigz]
   enddo
 enddo
}
do isl=1,numlev
 SUBA : [numsub(isl)]
 do ig=igresb,igrese
   [ig,xsga(isl,ig,il)il=1,nvlx]
 enddo
if (ioptxs=2) {
 SUBNF: [numsub(isl)]
 do ig=igresb,igrese
   [ig,xsgf(isl,ig,il)il=1,nvlx]
 enddo
}
 WGTA : [numsub(isl)]
 do ig=igresb,igrese
   do it=1,ntemp
     [ig,it,wgta(isl,it,ig,il)il=1,nvlx]
   enddo
 enddo
if (ioptxs=2) {
 WGTNF: [numsub(isl)]
 do ig=igresb,igrese
```

```
   do it=1,ntemp
     [ig,it,wgtf(isl,it,ig,il)il=1,nvlx]
   enddo
 enddo
}
**Repeat $NUC:
$END:
```

[Variable Description]

nnuc	I	number of nuclides
nuclid	I	nuclide identification for the resonant nuclide
nsigz	I	# of background XS
ntemp	I	number of temperatures
igresb	I	starting group for resonance
igrese	I	ending group for resonance
labsxs	L	logical for the existence of sig-a
lnfixs	L	logical for the existence of nu*sig-f
lsctxs	L	logical for the existence of sig-s
ioptxs	I	XS type option 1 : Absorption xs 2 : Absorption xs + Nu*fission xs
numlev	I	number of subgroup level types
temp(i)	R	temperatures (i=1,ntemp)
sigz(i)	R	background xs (i=1,nsigz)
ig	I	energy group
it	I	temperature ordering number
ria(it,ig,iz)	R	absorption resonance integral
ris(it,ig,iz)	R	scattering resonance integral
rif(it,ig,iz)	R	Nu*fission resonance integral
numsub(i)	I	number of subgroup levels
xsga(isl,ig,il)	R	subgroup levels for absorption xs
xsgf(isl,ig,il)	R	subgroup levels for nu*fission xs
wgta(isl,it,ig,il)	R	subgroup weights for absorption xs
wgtf(isl,it,ig,il)	R	subgroup weights for nu*fission xs

```
$DIM:
      6
$NUC:        1
         64155     16      5      1     47      T      T      F      1      2
  2.930000E+02  6.000000E+02  9.000000E+02  1.200000E+03  2.400000E+03
  1.778279E+01  3.162278E+01  5.623413E+01  1.000000E+02  1.778279E+02  3.162278E+02  5.623413E+02 ....
 RI-A :
      1      1  3.684452E-03  3.683047E-03  3.682375E-03  3.682205E-03  3.682201E-03  3.682201E-03 ....
      1      2  3.684345E-03  3.682967E-03  3.682570E-03  3.682314E-03  3.682049E-03  3.682049E-03
      1      3  3.684277E-03  3.682759E-03  3.682411E-03  3.682234E-03  3.681984E-03  3.681984E-03
      1      4  3.684283E-03  3.683001E-03  3.682618E-03  3.682316E-03  3.682146E-03  3.682146E-03
      1      5  3.684731E-03  3.683262E-03  3.682913E-03  3.682422E-03  3.682195E-03  3.682195E-03
....
     18      1  1.526399E+01  2.506623E+01  3.929973E+01  5.991735E+01  8.755505E+01  1.226166E+02 ....
     18      2  1.540759E+01  2.540904E+01  4.011916E+01  6.158251E+01  9.060761E+01  1.275242E+02
     18      3  1.553843E+01  2.571945E+01  4.084600E+01  6.302876E+01  9.317815E+01  1.314719E+02
....
 RI-S :
      1      1  4.919047E+00  4.920558E+00  4.921016E+00  4.921009E+00  4.920829E+00  4.920829E+00 ....
      1      2  4.919562E+00  4.921222E+00  4.921320E+00  4.921145E+00  4.920960E+00  4.920960E+00
....

 SUBA :      7
      1         4.318122E+00  3.598435E+01  2.249022E+02  6.747065E+02  1.799217E+03  6.297261E+03 ....
      2         4.318122E+00  3.598435E+01  2.249022E+02  6.747065E+02  1.799217E+03  6.297261E+03
....

     18         5.327255E+00  4.439379E+01  2.249022E+02  8.200824E+02  1.799217E+03  6.297261E+03 ....
     19         5.327255E+00  4.439379E+01  2.249022E+02  8.323836E+02  1.799217E+03  6.297261E+03
     20         5.016998E+00  4.244499E+01  2.318041E+02  8.079629E+02  1.854433E+03  6.297261E+03
....
     47         7.954224E+00  6.529092E+01  4.080683E+02  1.224205E+03  3.216302E+03  4.935710E+03
 WGTA :      7
      1      1  1.366447E-03 -1.102321E-04  2.140993E-05 -8.778303E-06  2.280418E-06 -2.610314E-07 ....
      1      2  1.366347E-03 -1.102063E-04  2.141543E-05 -8.789342E-06  2.285073E-06 -2.614092E-07
      1      3  1.366504E-03 -1.102798E-04  2.144860E-05 -8.807466E-06  2.290370E-06 -2.620983E-07
      1      4  1.366239E-03 -1.101712E-04  2.139974E-05 -8.777957E-06  2.280587E-06 -2.606023E-07
      1      5  1.366756E-03 -1.103063E-04  2.144321E-05 -8.799980E-06  2.286159E-06 -2.612369E-07
....
     18      3 -3.241712E-02  3.305267E-01  3.771768E-01  2.973695E-01  1.061015E-02 -7.908521E-04
     18      4 -2.631036E-02  2.965540E-01  3.979778E-01  3.241057E-01 -6.943130E-03  4.239550E-04
     18      5 -9.661847E-03  1.857938E-01  4.904229E-01  3.749928E-01 -5.019732E-02  3.969248E-03
....
SUBA :      4
      1         5.981802E+00  8.294978E+02  3.317991E+03  5.981802E+03
      2         6.072896E+00  8.172392E+02  3.268957E+03  6.072896E+03
      3         6.072896E+00  8.172392E+02  3.268957E+03  6.072896E+03
....
     47         1.093197E+01  5.465986E+02  1.799217E+03  9.133083E+03
 WGTA :      4
      1      1  7.058938E-04  2.517759E-08 -1.262299E-06  6.218236E-07
      1      2  7.058975E-04  2.448341E-08 -1.262087E-06  6.218080E-07
      1      3  7.058842E-04  2.377959E-08 -1.261908E-06  6.218090E-07
      1      4  7.058949E-04  2.512045E-08 -1.262237E-06  6.218534E-07
      1      5  7.059702E-04  2.264590E-08 -1.261780E-06  6.218476E-07
....
     47      1 -3.210303E+00 -2.000000E+00  2.000000E+00  2.000000E+00
     47      2 -3.210518E+00 -2.000000E+00  2.000000E+00  2.000000E+00
     47      3 -3.210484E+00 -2.000000E+00  2.000000E+00  2.000000E+00
     47      4 -3.210508E+00 -2.000000E+00  2.000000E+00  2.000000E+00
     47      5 -3.210514E+00 -2.000000E+00  2.000000E+00  2.000000E+00
$NUC:        2
         64156     16      5      1     47      T      T      F      1      2
  2.930000E+02  6.000000E+02  9.000000E+02  1.200000E+03  2.400000E+03
  1.778279E+01  3.162278E+01  5.623413E+01  1.000000E+02  1.778279E+02  3.162278E+02  5.623413E+02
```

Fig. 4.3. RI and subgroup data file.

5. SAMPLE PROBLEM

Figure 5.1 provides a sample input of SUBGR to generate subgroup data for 6 nuclides. A sample output of SUBGR is shown in Fig. 5.2.

```
%TITL
  SUBGROUP DATA GENERATION
%IOPT
  iop  2          !0/1/2: constant/variable(RI)/variable(sig)
  con  0.05 0.005 0.005
  grp   1  47
%FILE
  sub  subgr_47g_70s.lev
  rit  ssxs_47g_70s_10212014.dat
%RESO
   nuc     1    64155   2   1  !
   nuc     2    64156   2   1  !
   nuc     3    64157   2   1  !
   nuc     4    64158   2   1  !
   nuc     5    92235   2   2  !
   nuc     6    92238   2   1  !
%FINE
```

Fig. 5.1. Sample input of SUBGR.

```
Nuclide=     92235
    Completed lev=1 xs=1 gr=  1 it=  7 err= 0.0059 0.0039 0.0028  …
    Completed lev=1 xs=1 gr=  2 it=  7 err= 0.0048 0.0043 0.0031
    Completed lev=1 xs=1 gr=  3 it=  7 err= 0.0051 0.0040 0.0029
    Completed lev=1 xs=1 gr=  4 it=  7 err= 0.0053 0.0040 0.0029
…
    Completed lev=1 xs=1 gr= 23 it=  7 err= 0.0011 0.0002 0.0001 …
    Completed lev=1 xs=1 gr= 24 it= 51 err= 0.1684 0.0051 0.0033
    Completed lev=1 xs=1 gr= 25 it=  7 err= 0.0123 0.0004 0.0003
…
    Completed lev=1 xs=1 gr= 47 it=  7 err= 0.1958 0.0001 0.0001
    Completed lev=1 xs=2 gr=  1 it=  7 err= 0.0204 0.0039 0.0028
    Completed lev=1 xs=2 gr=  2 it=  7 err= 0.0143 0.0043 0.0031
…
    Completed lev=2 xs=2 gr= 46 it=  7 err=76.6645 0.0491 0.0426
    Completed lev=2 xs=2 gr= 47 it= 10 err******* 0.0498 0.0429

@NUCLID=  92235


Absorption ri(xs)
-----------------

  group=   1
    1   xst:    1.523837E+00  1.519290E+00  1.517347E+00  1.516621E+00  1.516297E+00  1.515799E+00 …
        sub:    1.514735E+00  1.533997E+00  1.520292E+00  1.507884E+00  1.510793E+00  1.524706E+00
                error (max/rms/rms2) = 0.00968/ 0.00581/ 0.00383
    2   xst:    1.524338E+00  1.519163E+00  1.517809E+00  1.517313E+00  1.516058E+00  1.516293E+00
        sub:    1.515123E+00  1.534238E+00  1.520503E+00  1.508170E+00  1.511162E+00  1.524830E+00
                error (max/rms/rms2) = 0.00992/ 0.00585/ 0.00385
    3   xst:    1.523790E+00  1.519053E+00  1.517405E+00  1.517133E+00  1.515861E+00  1.516027E+00
        sub:    1.514590E+00  1.533990E+00  1.520346E+00  1.507932E+00  1.510799E+00  1.524783E+00
                error (max/rms/rms2) = 0.00983/ 0.00586/ 0.00386
    4   xst:    1.524322E+00  1.519626E+00  1.516991E+00  1.516807E+00  1.516285E+00  1.516188E+00
        sub:    1.515300E+00  1.534089E+00  1.520194E+00  1.507855E+00  1.510989E+00  1.524894E+00
                error (max/rms/rms2) = 0.00952/ 0.00578/ 0.00380
    5   xst:    1.524267E+00  1.518931E+00  1.517904E+00  1.517019E+00  1.516199E+00  1.515811E+00
        sub:    1.514931E+00  1.534219E+00  1.520515E+00  1.508072E+00  1.510912E+00  1.524677E+00
                error (max/rms/rms2) = 0.01006/ 0.00592/ 0.00390

…

  group=   4
    1   xst:    1.332250E+00  1.331825E+00  1.332089E+00  1.331961E+00  1.332032E+00  1.332450E+00 …
        sub:    1.323899E+00  1.345364E+00  1.334488E+00  1.324139E+00  1.327214E+00  1.340199E+00
                error (max/rms/rms2) = 0.01017/ 0.00528/ 0.00396
    2   xst:    1.332104E+00  1.332270E+00  1.332021E+00  1.332272E+00  1.332403E+00  1.332233E+00
        sub:    1.323801E+00  1.345557E+00  1.334794E+00  1.324390E+00  1.327320E+00  1.340349E+00
                error (max/rms/rms2) = 0.00997/ 0.00528/ 0.00396
    3   xst:    1.332111E+00  1.331962E+00  1.332275E+00  1.332033E+00  1.332436E+00  1.332298E+00
        sub:    1.323793E+00  1.345443E+00  1.334669E+00  1.324353E+00  1.327395E+00  1.340380E+00
                error (max/rms/rms2) = 0.01012/ 0.00526/ 0.00395
    4   xst:    1.332640E+00  1.331715E+00  1.332674E+00  1.331736E+00  1.331903E+00  1.332229E+00
        sub:    1.324190E+00  1.345601E+00  1.334642E+00  1.324144E+00  1.327060E+00  1.339892E+00
                error (max/rms/rms2) = 0.01043/ 0.00532/ 0.00399
    5   xst:    1.331652E+00  1.332125E+00  1.332256E+00  1.331736E+00  1.332045E+00  1.332199E+00
        sub:    1.323548E+00  1.345286E+00  1.334559E+00  1.324238E+00  1.327208E+00  1.339955E+00
                error (max/rms/rms2) = 0.00988/ 0.00512/ 0.00385

  group=   5
    1   xst:    1.284633E+00  1.284322E+00  1.285001E+00  1.284270E+00  1.284182E+00  1.283838E+00 …
```

Fig. 5.2. Sample output of SUBGR.

REFERENCES

[Cho01] J. Y. Cho and et al., "Cell Based [Computational Multi-Fluid Dynamics] CMFD Formulation for Acceleration of Whole-core Method of Characteristics Calculations," Proceedings of the Korea Nuclear Society (KNS) Fall Mtg., Suwon (2001).

[Hel03] *HELIOS Methods*, Studsvik Scandpower, November 20, 2003.

[Joo09] H. G. Joo and et al., "Subgroup Weight Generation Based on Shielded Pin-Cell Cross Section Conservation," *Ann. Nucl. Energy*, 36, 859 (2009).

[Kha72] A. Khairallah and J. Recolin, "Calcul de l'autoprotection résonnante dans les cellules complexes par la méthode des sous-groupes," *Proc. Seminar IAEA-SM-154 on Numerical Reactor Calculation*s, IAEA, Vienna (1972): 305–317.

[Kim15] Kang Seog Kim, Mark L. Williams, Dorothea Wiarda, and Andrew T. Godfrey, "Development of a New 47-Group Library for the CASL Neutronics Simulators," M&C 2015, Nashville, TN, USA, April 19–23, 2015 (2015).

[Koc15] Brendan Kochunas, Benjamin Collins, Daniel Jabaay, Shane Stimpson, Aaron Graham, Kang Seog Kim, William Wieselquist, Kevin Clarno, Scott Palmtag, Thomas Downar and Jess Gehin, "VERA Core Simulator Methodology for [Pressurized Water Reactor] PWR Cycle Depletion," M&C 2015, Nashville, TN, USA, April 19–23, 2015 (2015).

[Not87] C. Notari and Z. Garraffo, "Spatial self-shielding for heterogeneous cells," *Ann. Nucl. Energy* 14 (1987): 615–618.

[Rot74] M. J. Roth, "Resonance absorption in complicated geometries," United Kingdon Atomic Energy Authority (UKAEA) Report AEEW-R 921, Winfrith, England (1974).