

# Development of Cross Section Library and Application Programming Interface (API)

---

Nuclear Engineering Division

### **About Argonne National Laboratory**

Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC under contract DE-AC02-06CH11357. The Laboratory's main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne and its pioneering science and technology programs, see [www.anl.gov](http://www.anl.gov).

### **DOCUMENT AVAILABILITY**

**Online Access:** U.S. Department of Energy (DOE) reports produced after 1991 and a growing number of pre-1991 documents are available free via DOE's SciTech Connect (<http://www.osti.gov/scitech/>)

### **Reports not in digital format may be purchased by the public from the National Technical Information Service (NTIS):**

U.S. Department of Commerce  
National Technical Information Service  
5301 Shawnee Rd  
Alexandria, VA 22312  
**[www.ntis.gov](http://www.ntis.gov)**  
Phone: (800) 553-NTIS (6847) or (703) 605-6000  
Fax: (703) 605-6900  
Email: **[orders@ntis.gov](mailto:orders@ntis.gov)**

### **Reports not in digital format are available to DOE and DOE contractors from the Office of Scientific and Technical Information (OSTI):**

U.S. Department of Energy  
Office of Scientific and Technical Information  
P.O. Box 62  
Oak Ridge, TN 37831-0062  
**[www.osti.gov](http://www.osti.gov)**  
Phone: (865) 576-8401  
Fax: (865) 576-5728  
Email: **[reports@osti.gov](mailto:reports@osti.gov)**

### **Disclaimer**

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor UChicago Argonne, LLC, nor any of their employees or officers, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of document authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, Argonne National Laboratory, or UChicago Argonne, LLC.

# Development of Cross Section Library and Application Programming Interface (API)

---

Prepared by  
C. H. Lee, A. Marin-Lafleche, and M. A. Smith  
Nuclear Engineering Division, Argonne National Laboratory

September 30, 2013



## EXECUTIVE SUMMARY

The goal of NEAMS neutronics is to develop a high-fidelity deterministic neutron transport code termed PROTEUS for use on all reactor types of interest, but focused primarily on sodium-cooled fast reactors. While PROTEUS-SN has demonstrated good accuracy for homogeneous fast reactor problems and partially heterogeneous fast reactor problems, the simulation results were not satisfactory when applied on fully heterogeneous thermal problems like the Advanced Test Reactor (ATR). This is mainly attributed to the quality of cross section data for heterogeneous geometries since the conventional cross section generation approach does not work accurately for such irregular and complex geometries. Therefore, one of the NEAMS neutronics tasks since FY12 has been the development of a procedure to generate appropriate cross sections for a heterogeneous geometry core.

Several major cross section methodologies including the subgroup method, the resonance integral table method, and the direct resonance self-shielding method were reviewed to determine the best-fit cross section approach to a high-fidelity neutron transport simulation of various reactor types with fully heterogeneous geometry modeling. Investigations indicated that the conventional cross section libraries with at most a few hundred energy groups are limited to use on the specific reactor types for which the cross section library was generated. This is because the neutron spectra are very different between reactor types, and the importance of cross section characteristics in terms of resonance, scattering, and reaction is energy-dependent. Therefore, the reduced energy group libraries involve some degree of approximation and assumption which limits the accuracy of the cross section library for a targeted reactor type. This limitation is one of the known shortcomings in the deterministic method, compared to the stochastic method which directly uses the continuous energy cross section data.

To meet the NEAMS neutronics goal, a generalized cross section methodology and library was developed for application to various reactor types including light water reactor (LWR), very high temperature reactor (VHTR), and sodium-cooled fast reactor (SFR). The ultrafine group (2158 groups) cross section library including the resonance integral tables for resonance cross sections was produced by the GeneCS code using the cross section data generated from MC<sup>2</sup>-3 and NJOY. The resonance integral tables were formulated for absorption, nu-fission, and scattering cross sections. The ultrafine group cross section library can be condensed to a broad group library (< 300 groups) using GeneCS for specific use on a target reactor. This is accomplished using a group condensation optimization algorithm which uses a representative neutron spectrum and various homogeneous or pin cell compositions for the reactor type of interest. Note that this is a group condensation process from library to library such that the resulting broad group library can be directly used for any transport code via the cross section application programming interface (API) we have developed during this year. The number of the broad groups for the reduced library is determined by a group condensation error criterion. It should be emphasized that this cross section library generation procedure can be used for both the conventional (i.e. legacy codes) and high-fidelity multigroup cross section generation process.

Since the primary reactor system of interest to NEAMS has been a SFR, we additionally developed a rigorous cross section generation approach using the direct

resonance self-shielding method as a high-fidelity cross section generation approach for SFR, in which the MC<sup>2</sup>-3 methodology was extended from 1D geometries to large scale 3D geometries. This direct resonance self-shielding method can be used when more rigorous and accurate cross sections are required for fast reactor analysis. Unlike the generalized cross section approach, this method is at the moment limited to fast reactor analysis because the development of MC<sup>2</sup>-3 was focused exclusively on fast spectrum work and significant research and code modifications are necessary to incorporate thermal reactor analysis. We note that further research and development on the rigorous MC<sup>2</sup>-3 approach can improve the generalized cross section library that we developed this year.

Verification tests for the generalized cross section library were performed mostly using DeCART since the incorporation of the cross section API into PROTEUS was being done simultaneously and thus not ready during the testing of the cross section library. A new cross section library was generated in the DeCART format which does not include all of the necessary data that a general transport code needs. While a more general format containing the missing data should be developed in the future, we note that the DeCART format was useful to support the I-NERI collaboration on DeCART between Argonne and KAERI which has been conducted this year. Verification results of the new libraries indicated that the eigenvalues were estimated within 200-300 pcm for all LWR, VHTR, and SFR compositions depending on the resulting optimized group structures with 76 to 383 groups, compared to the Monte Carlo solutions.

The cross section application programming interface (API) was developed to make it easy to plug a set of cross section module or package into an existing neutron transport code. As part of this API, the transport code must provide a one-group fixed-source solver, the isotopic breakdown of each composition in the domain, and the mapping between those compositions and the geometry. In the API, the input and output arguments required are clearly defined so that a user can understand what the API needs from and provides to the transport code. The API was first developed with the subgroup method and PROTEUS-MOC was targeted as the initial test implementation where numerous interface subroutines were created to setup the API and retrieve the effective multigroup cross-section library from the API. Verification tests were performed using the VERA benchmark problems obtained from the Argonne and ORNL collaboration activities on the subgroup method.

In the future, detailed verification and validation tests for the generalized cross section library will be performed for various heterogeneous cases including whole-core problems. A complete set of cross section libraries with a more general data format will be provided by including all nuclides and reaction types. The cross section API will be implemented into PROTEUS-SN as well so that larger benchmark problems such as the ATR can be executed.

## TABLE OF CONTENTS

Executive Summary .....	i
Table of Contents .....	iii
List of Figures .....	iv
List of Tables.....	v
1 Introduction .....	1
2 Development of Cross Section Method and Library .....	4
2.1 Resonance Self-shielding Methods .....	4
2.1.1 The Conventional Subgroup Method .....	4
2.1.2 The Resonance Integral Table Method.....	6
2.1.3 The Direct Resonance Self-shielding Method.....	7
2.1.4 Other Method.....	13
2.2 Development of a Generalized Cross Section Library.....	15
2.2.1 Introduction .....	15
2.2.2 Methodology.....	17
2.2.3 Library Generation .....	20
2.3 Verification Tests .....	22
2.3.1 Preliminary Group Optimization Tests.....	22
2.3.2 Resonance Self-shielding Table Tests.....	24
2.3.3 Library Tests.....	26
3 The Cross Section API .....	37
3.1 Structure .....	38
3.1.1 Initialization.....	38
3.1.2 Cross Section Region Setup .....	39
3.1.3 Cross Section Calculation.....	40
3.1.4 Multigroup Cross Section Retrieval .....	42
3.1.5 Utilities .....	42
3.2 Implementation of the Cross Section API into PROTEUS.....	44
3.3 Verification Tests .....	45
3.3.1 Pin Cell Problems .....	47
3.3.2 Lattice Problems .....	49
4 Conclusions .....	51
References .....	53
Appendix A. Major Subroutine and Function Headers of the Subgroup API .....	55

## LIST OF FIGURES

Figure 1.1 A Generalized Cross Section Library and the Cross Section API Integrated in PROTEUS.....	3
Figure 2.1 Differences of Multigroup Total Cross Sections of Pu-239 between the MC <sup>2</sup> -3 and Bondarenko Approaches for the Monju Fuel Composition .....	6
Figure 2.2 Resonance Self-shielding for Whole Core Problems .....	10
Figure 2.3 2D Multi-Pin Problems with Fuel Pins and Reflector Compositions.....	11
Figure 2.4 1D Problem Equivalent to the 2D Problems .....	11
Figure 2.5 UFG Escape Cross Sections of Fe-56 Generated from the 1D Problem.....	12
Figure 2.6 UFG Escape Cross Sections of U-238 Generated from the 1D and 2D (Types A and B) Problems.....	12
Figure 2.7 230-Group Escape Cross Sections of U-238 Estimated by Three Different Methods for Two-region Pin-Cell Problems.....	14
Figure 2.8 A Procedure of Generating a Cross Section Library Using the GeneCS Code.....	16
Figure 2.9 Difference of 72-Group Absorption Cross Sections of U-238 between DeCART and MCNP5 for a Typical VHTR Pin-cell .....	17
Figure 2.10 Group Optimization Algorithms from UFG to BG Libraries.....	19
Figure 2.11 Lethargy Intervals of the Ultrafine Group Structure .....	21
Figure 2.12 Cross Section Library Generation Using the GeneCS Code.....	21
Figure 2.13 Multigroup Macroscopic Cross sections and Reaction Rate Errors [pcm] between the DeCART 190 Groups (red) and the New 215 Groups (blue).....	23
Figure 2.14 Absorption Resonance Integral versus Background Cross Section at a Resonance Group of U-238 .....	24
Figure 2.15 Resonance Cross Sections as a Function of Background Cross Section for U-238 at 1 keV (left) and 0.8 keV (right) .....	25
Figure 2.16 Neutron Spectra of Sodium Fast Reactor Compositions.....	27
Figure 2.17 Lethargy Intervals of the Broad Group Structures for Sodium Fast Reactor .....	29
Figure 2.18 U-238 Total Cross Sections in Sodium Fast Reactor (Case 5).....	29
Figure 2.19 Differences of Total Cross Sections between MC <sup>2</sup> -3 and the RI Table Method in Sodium Fast Reactor (Case 5) .....	30
Figure 2.20 Neutron Spectra of Light Water Reactor Compositions.....	31
Figure 2.21 Lethargy Intervals of the Broad Group Structures for Light Water Reactor.....	33
Figure 2.22 U-238 Total Cross Sections with Different Group Structures in Light Water Reactor (Case 1).....	33
Figure 2.23 Neutron Spectra of High Temperature Reactor Compositions.....	34
Figure 2.24 Lethargies of the Broad Group Structures for High Temperature Reactor .....	36
Figure 2.25 U-238 Total Cross Sections with Different Group Structures in High Temperature Reactor (Case 6) .....	36
Figure 3.1 Interaction between the Cross Section API and the Neutron Transport Solver .....	37

Figure 3.2 Subroutine Call from the Subgroup API to the Transport Solver .....	39
Figure 3.3 Composition Assignment (left), Element Indexing (middle), and Cross Section Region Mapping (right) for a 2x2 Pin-Cell Geometry .....	39
Figure 3.4 Flow Chart of the Computation Phase.....	41
Figure 3.5 Subgroup Interface Used to Retrieve Macroscopic Cross Section Data .....	42
Figure 3.6 Doxygen HTML Output for Composition Module .....	43
Figure 3.7 Cross Section Region Mapping to 4 Processors .....	44
Figure 3.8 Meshes of Benchmark Problems for the Cross Section API.....	47
Figure 3.9 Flux Distributions (at ~3 eV) for 2x2 Pin Cell Benchmark Problems .....	50

### LIST OF TABLES

Table 2.1 Comparison of Eigenvalues for 1D and 2D Problems with Different Escape Cross Sections .....	10
Table 2.2 VHTR Pin Cell Configurations for Group Structure Optimization .....	22
Table 2.3 Optimized Energy Group Structures.....	22
Table 2.4 Comparison of Eigenvalues of Various Homogeneous Compositions between MCNP5 and the RI Table Method .....	26
Table 2.5 Isotopes and Number Densities of Sodium Fast Reactor Compositions .....	27
Table 2.6 Eigenvalue Comparison for Sodium Fast Reactor Compositions.....	28
Table 2.7 Isotopes and Number Densities of Light Water Reactor Compositions .....	31
Table 2.8 Eigenvalue Comparison for Light Water Reactor Compositions .....	32
Table 2.9 Isotopes and Number Densities of High Temperature Reactor Compositions .....	34
Table 2.10 Eigenvalue Comparison for High Temperature Reactor Compositions .....	35
Table 3.1 Selected VERA PWR Core Physics Benchmark Problems <sup>*</sup> .....	46
Table 3.2 Eigenvalue Comparison of the Selected VERA Benchmark Problems.....	46
Table 3.3 Composition of PWR-VERA-1A Pin Cell Problem.....	48
Table 3.4 Eigenvalues with Change of Parameters for a Pin Cell Problem.....	48
Table 3.5 Composition of PWR-VERA-1E Pin Cell Problem .....	49
Table 3.6 Eigenvalues with Change of Parameters for a Pin Cell Problem.....	49
Table 3.7 Eigenvalues with Change of Parameters for 2x2 Pin Cell Benchmark Problems ....	50



## 1 Introduction

The goal of NEAMS neutronics work is to develop a high-fidelity deterministic neutron transport code termed PROTEUS [1] for use on all reactor types of interest. While PROTEUS-SN has demonstrated good accuracy for homogeneous fast reactor problems [2] and partially heterogeneous fast reactor problems [3], the simulation results were not satisfactory when applied to fully heterogeneous thermal problems such as the ATR [4]. This is attributed to the quality of cross section data for heterogeneous geometries since the conventional cross section generation approach does not work for such an irregular and complex geometry core. Therefore, one of the NEAMS neutronics tasks was directed to provide appropriate neutron cross section data for a heterogeneous geometry core starting from FY12 [5] and continuing in FY13.

Reviews were carried out on the subgroup method for heterogeneous geometry cross section generation since that method has been widely used in many physics codes such as HELIOS [6], DRAGON [7], WIMS [8], PARAGON [9], APOLLO [10], DeCART [11], and ECCO [12] to deal with the heterogeneity effect for generation of the effective cross sections. The method has been demonstrated for thermal systems, but not clearly for fast systems, even though ECCO adopts the method for fast reactor applications. In fast spectrum systems, the heterogeneity effect is relatively less important, whereas the resonance interaction and neutron spectrum effects are more important. Thus, it is worthwhile to quantify how well the subgroup method works for fast spectrum systems and to assess the best methodology for various spectrum systems.

Last year, we established the cross section library generation procedure using the GeneCS (pronounced as “genesis”) code [5] with the support of NJOY [13] and MCNP5 [14]. The goal of the cross section work this year was to continue developing the cross section library and the subgroup application programming interface (API). The PROTEUS code was to be the first demonstration code of the API. The research component of this work led to improvements on resonance self-shielding in order to accurately handle fast reactor problems. The subgroup API should be designed to include common interface routines such that it can be plugged into other neutron transport tools with minimal effort. For a peer review, we have had periodic (weekly or biweekly) meetings with the ORNL expert group that is interested in the subgroup method and is developing the embedded self-shielding method (ESSM) [15] as well as the API for accessing the AMPX library [16].

The goal of NEAMS neutronics is to develop a high-fidelity deterministic neutron transport code for use on general reactor types, but initially on sodium-cooled fast reactors. Previous experience indicated that an approximate treatment of the resonance interference effect, as done in the conventional subgroup method, is not adequate to achieve the desired accuracy in fast systems where resonance interferences exist between actinides and also between actinides and intermediate-weight isotopes (structural materials). As a high-fidelity cross section generation approach, we developed the direct resonance self-shielding method in which the MC<sup>2</sup>-3 methodology was extended from 1D to 3D large scale calculation. Since the method requires large fixed source transport calculations to determine the escape cross sections in every cross section region, significant computational effort is expected for actual reactor sized problems. To extend the approach to thermal reactor systems, significant research and development effort will be needed for updating the existing MC<sup>2</sup>-3 methodology

[17]. Based upon our previous research on thermal systems, generating cross sections in the complicated manner employed in the MC<sup>2</sup>-3 methodology may not be necessary for thermal systems. Therefore, we decided to develop a new resonance self-shielding methodology which can be applied to all reactor types.

The conventional process of generating a subgroup library typically is only applicable to a given specific reactor type. Historically, the energy group structures at each step of the generation procedure were specifically tuned to light water reactors (LWRs) and thus they cannot be directly used for very high temperature reactors (VHTRs) or sodium-cooled fast reactors (SFRs). The accuracy of the cross section library is often degraded when generating the library to accommodate a wider range of neutron spectra for use in other reactor cores. In the new generation procedure we researched, one starts with a single ultrafine group library (~2000 groups) and tailors the broad group library (< 300 groups) specifically to the reactor type of interest. This is obviously a much simpler procedure and thus less prone to user error and avoids the typical loss of accuracy associated with making an all-purpose library.

In this year, the base ultrafine group cross section library was developed for use on different reactor types including LWR, VHTR, and SFR. The ultrafine group cross section library including the resonance integral tables for resonance cross sections is produced by the GeneCS code using the cross section data generated from MC<sup>2</sup>-3 and NJOY. The resonance integral tables are formulated for absorption, nu-fission, and scattering cross sections. The ultrafine group cross section library is condensed to the broad group library using GeneCS for specific use on a target reactor. This is accomplished using a group condensation optimization algorithm where a representative neutron spectrum and various homogeneous or pin cell compositions for the reactor type of interest. Note that this is a group condensation process from library to library such that the resulting broad group library can be directly used in any transport code. The number of broad groups for the reduced library can be determined by a group condensation error criterion.

Verification tests of the new cross section library were performed mostly using DeCART since the implementation of the cross section API into PROTEUS was being done at the same time. The new cross section library was generated in the DeCART library format which does not include all of the necessary data that a transport code needs and will have to be updated to hold a complete set of data in the future. However, we note that the library in the DeCART format was useful to support the I-NERI collaboration [19] between Argonne and KAERI, whose goal is to verify and validate DeCART and multi-physics simulation for advanced nuclear reactors.

The cross section API was developed to make it easy to plug the developed cross section module or package into an existing neutron transport code. To accomplish this task, the transport code is required to provide a one-group fixed-source transport solver, the isotopic breakdown of each composition in the domain, and the mapping between those compositions and the geometry. In the API, the input and output arguments required are clearly defined so that a user can understand what the API needs from and provides to the transport code. The API was developed first with the subgroup API and integrated with PROTEUS-MOC to demonstrate its application. Numerous interface subroutines in the neutron transport solver were created to setup the API and retrieve the effective multigroup cross-section library from the API. Figure 1.1 graphically shows the whole idea of the

NEAMS neutronics work that has been accomplished partly this fiscal year and is going to be completed in the following fiscal years.

Section 2 describes the resonance self-shielding methods as well as the library generation. The section includes preliminary verification tests for the new cross section library. Section 3 presents the cross section API and its verification tests. The final section contains the conclusions.

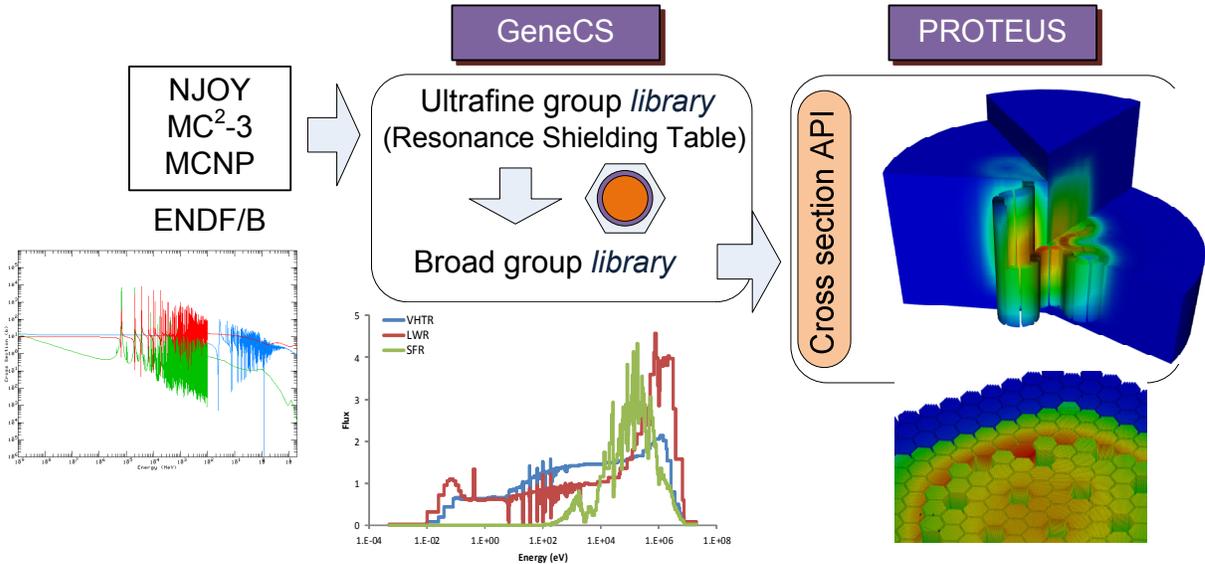


Figure 1.1 A Generalized Cross Section Library and the Cross Section API Integrated in PROTEUS

## 2 Development of Cross Section Method and Library

For resonance self-shielding in a complex heterogeneous system, the subgroup method has been employed by many neutron physics and core simulation codes due to its good performance. Since the method is based on equivalence theory, it requires the pre-calculated tables or parameters which are functionalized with the background cross section and temperature. Those subgroup parameters should be prepared for individual isotopes to satisfy the range of neutron spectrum, background cross section, and temperature in the reactor system of interest. Since the parameters are generated for individual isotopes, the resonance interference between different resonant isotopes needs to be accounted for in an approximate way in the neutron transport code. Since we are also targeting a fast spectrum system, the approximate way of treating the resonance interference effect may be not accurate enough to meet our desired criteria. Therefore, while implementing and testing the conventional subgroup method, we have developed new resonance self-shielding methods or approaches to improve accuracy and consistency between library generation and its use in the neutron transport code.

### 2.1 Resonance Self-shielding Methods

#### 2.1.1 The Conventional Subgroup Method

The subgroup method determines the effective resonance cross sections without the intermediate calculation of Dancoff factors, and thus it is useful for arbitrary geometry or direct whole core transport calculations in which spatially dependent self-shielding should be properly considered. The flux solution is represented as below in terms of the background cross section  $\sigma_b$  and the microscopic absorption cross section  $\sigma_a^r$  of resonance  $r$ .

$$\phi(u) = \frac{\sigma_b}{\sigma_a^r(u) + \sigma_b}. \quad (2.1)$$

The resonance integral is approximated by quadratures and thus the effective absorption cross section  $\bar{\sigma}_a^r$  of resonance  $r$  is determined in terms of the subgroup weight  $w_n$  and level  $\sigma_n$  and flux solution  $\phi_n$  for the broad group as

$$\bar{\sigma}_a^r = \frac{\sum_n w_n \phi_n \sigma_n}{\sum_n w_n \phi_n}. \quad (2.2)$$

The subgroup parameters  $w_n$  for individual isotopes are prepared as a function of background cross section and temperature. Therefore, the resonance interference effect due to the presence of other resonant isotopes in a mixture is accounted for using the Bondarenko iteration in which other isotope cross sections are treated as a constant over the energy group. In the conventional subgroup method, the parameters are determined using the background cross sections estimated from the solution of a fixed source problem (FSP). From there, the

Bondarenko iteration is performed only with the subgroup parameters already determined from the FSP,

$$\Sigma_a^k(T) = \frac{\sum_n w_{an}^k(T) \Sigma_{an}^k \frac{\Sigma_{bn}^k}{\Sigma_{an}^k + \hat{\Sigma}_a^k(T) + \Sigma_{bn}^k}}{1 - \sum_n w_{an}^k(T) \frac{\Sigma_{an}^k}{\Sigma_{an}^k + \hat{\Sigma}_a^k(T) + \Sigma_{bn}^k}}, \quad (2.3)$$

where  $\hat{\Sigma}_a =$  absorption cross section of other resonant isotopes,  $w_{an}^k$ ,  $\Sigma_{an}^k$ ,  $\Sigma_{bn}^k =$  weighting factor, absorption cross section, and background cross section, respectively, of subgroup level  $n$  of isotope  $k$ .

The subgroup method requires considerable effort to prepare the subgroup parameters using the least square method (LSM) and can sometimes produce unexpected results due to the interpolation of parameters rather than the interpolation of resulting values. Since many codes still use the conventional subgroup method, we have implemented and tested it.

It is known that the subgroup method is appropriate for systems with compositions in which there are only a few dominant resonant isotopes and there is minimal or no resonance overlapping between the dominant resonant isotopes. Accordingly, it is qualitatively noted that the subgroup method has worked well for thermal systems where U-238 is a dominant isotope, and the accurate treatment of the resonance cross sections of structural material is not that important. The subgroup method may not work well for fast systems where there are multiple dominant resonant isotopes and accurate estimation of resonance cross sections of structural material is important. However, it is interesting to note that the ECCO code has used the subgroup method for fast reactor analysis. In this report, we will discuss the performance of the subgroup method for fast reactor applications in the verification test section.

In order to understand how good the Bondarenko approach works for fast reactor compositions, multigroup cross sections were generated from MC<sup>2</sup>-3 and the Bondarenko approach. The Monju startup core compositions were selected for the test, in which the fuel composition contains 79 wt% U-238 and 13 wt% Pu-239 and the structure composition is composed of 67 wt% Fe, 18 wt% Cr, and 9 wt% Ni.

While MC<sup>2</sup>-3 directly accounts for the resonance interference by using the pointwise cross sections of all isotopes in the composition, the Bondarenko approach considers the resonance interference for an individual isotope for which the other isotopes in the composition are treated as a constant background cross section. Thus, for a dominant isotope for which the background cross section is relatively small, multigroup cross sections generated from both MC<sup>2</sup>-3 and the Bondarenko approach would not have big differences. In addition, both methods produce similar cross sections for resonant isotopes whose resolved resonances do not overlap with each other (e.g., U-235 and Fe-56).

Including the Monju compositions, most of the typical fast reactor compositions contain a large amount of U-238 and Fe-56 compared to other isotopes. The resolved resonances of the two isotopes do not overlap each other and thus the test results showed that the cross sections produced from the MC<sup>2</sup>-3 approach and the Bondarenko approach were in good agreement even in the ultrafine groups. As can be expected, for isotopes with relatively

small concentrations such as Pu-239, Cr, and Ni, the differences of resulting ultrafine-group cross sections between the two approaches were larger than 30%. This large difference is caused by the differences in detailed (or energy-dependent) and constant background cross sections within each resonance group. Interestingly, however, those large differences that appear in the ultrafine group become smaller due to favorable error cancellation as the number of groups is reduced. Figure 2.1 shows percent differences of multigroup cross sections of Pu-239 between the two approaches. The percent differences are greater than 30% for ultrafine groups, but they become less than 5% for 230 groups and less than 2% for 33 groups. Similar results were obtained from other fast reactor core compositions.

The test results indicate that the subgroup method may work for fast reactor systems as long as a small number of cross section groups are used. Even for many groups, the impact of cross sections on eigenvalue with large errors would be cancelled out.

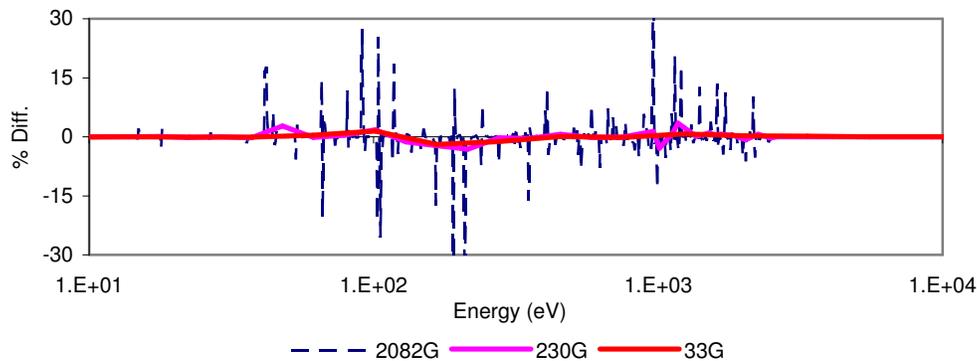


Figure 2.1 Differences of Multigroup Total Cross Sections of Pu-239 between the MC<sup>2</sup>-3 and Bondarenko Approaches for the Monju Fuel Composition

### 2.1.2 The Resonance Integral Table Method

As an alternative to the conventional subgroup method, the resonance integral (RI) table method has been proposed and tested. This method determines the resonance cross sections in an iterative manner, whereas the conventional subgroup method uses a quadrature representation with the subgroup cross sections and corresponding weighting factors and therefore no iteration is necessary. At every iteration, the resonance cross sections of resonant isotopes are updated based on its RI table where the resonance integral data is stored as a function of the background cross section and temperature. The RI table method uses the same basic library data necessary for the conventional subgroup method. While the conventional subgroup method solves the fixed-source neutron transport problem (FSP) for normally 4 subgroups (sub-levels), the iterative method repeatedly solves the FSP using updated cross sections until convergence is reached. From our previous work, we observed that the absorption cross sections converge very quickly within 2-3 iterations, thus the computational efforts for both methods should be comparable. As aforementioned, the RI table method is simpler than the conventional subgroup method in terms of data preparation and manipulation.

Like the conventional subgroup method, the RI table method uses the Bondarenko iteration to account for the resonance interference effect. The RI table method directly interpolates RI values with the background cross section while the subgroup method interpolates subgroup parameters and then calculates resonance cross sections by using the quadrature form. The resonance cross section is determined by

$$\sigma_{\alpha}^k(T) = \frac{RI_{\alpha}^k(\sigma_b^k, T)}{1 - RI_{\alpha}^k(\sigma_b^k, T) / \sigma_b^k}, \quad (2.4)$$

where the microscopic background cross section of the isotope  $k$  is the sum of the total cross sections of the other isotopes,  $\sigma_b^k = \sum_{i \neq k} N^i \sigma_t^i / N^k$ , and  $RI_{\alpha}^k(\sigma_b^k, T)$  is the resonance integral for type  $\alpha$  (absorption, nu-fission, or scattering) at the microscopic background cross section  $\sigma_b^k$  and temperature  $T$ . The background cross section is re-evaluated while the absorption cross sections of the other resonant isotopes are updated during the iteration.

### 2.1.3 The Direct Resonance Self-shielding Method

Both the RI table and subgroup methods rely on the Bondarenko iteration to account for the resonance interference effect. Since the Bondarenko method uses a constant background cross section over a unit energy group, there is a limitation in accurately considering the complex resonance interference between multiple resonant isotopes. Since the MC<sup>2</sup>-3 method has been proved to be accurate for fast system cross section generation [20], an improved resonance self-shielding method [21] has been developed for heterogeneous whole-core application by extending the MC<sup>2</sup>-3 method to the whole-core level. In this method, the resonance interference effect is accounted for through the narrow resonance (NR) approximation or slowing-down calculations for specific compositions, and the heterogeneity effect is accounted for by the use of isotopic escape cross sections. The isotopic escape cross sections are estimated from the fixed-source transport calculation for the whole-core problem domain, similarly to the subgroup method.

Due to the apparent problems with the subgroup method, Tone's method [22] was reviewed which is based on the collision probability method. By applying the narrow resonance approximation, the flux at region  $i$  can be expressed as

$$\phi_i(E) = \frac{1}{E} \frac{\sum_j P_{ji}(E) \Sigma_{pj} V_j}{\sum_j P_{ji}(E) \Sigma_{tj}(E) V_j}, \quad (2.5)$$

where  $P_{ji}(E)$  is the collision probability from region  $j$  to  $i$ ,  $\Sigma_{tj}(E)$  and  $\Sigma_{pj}$  is the total and potential cross sections of region  $j$ , respectively, and  $V_j$  is the volume of region  $j$ . Separating out the resonant isotope  $r$  of interest, Eq. (2.5) can be rewritten as

$$\phi_{r,i}(E) = \frac{1}{E} \frac{\sigma_{p,r} \sum_j P_{ji}(E) N_{r,j} V_j + \sum_j \sum_{k \neq r} P_{ji}(E) \sigma_{p,k,j} V_j}{\sigma_{t,r}(E) \sum_j P_{ji}(E) N_{r,j} V_j + \sum_j \sum_{k \neq r} P_{ji}(E) \sigma_{t,k,j}(E) V_j}, \quad (2.6)$$

which can be expressed as

$$\phi_{r,i}(E) = \frac{1}{E} \frac{\sigma_{p,r} + \sigma_{p,r,i}^0(E)}{\sigma_{t,r}(E) + \sigma_{t,r,i}^0(E)}, \quad (2.7)$$

$$\text{where } \sigma_{t,r,i}^0(E) = \frac{\sum_j \sum_{k \neq r} P_{ji}(E) \Sigma_{t,k,j}(E) V_j}{\sum_j P_{ji}(E) N_{r,j} V_j}.$$

Using Tone's approximation,  $P_{ji}(E)/\Sigma_{ti}(E) = \alpha_i(E) P_{ji}^g / \Sigma_{ti}^g$ , and the additional approximation of  $\Sigma_{t,k,j}(E)$  as  $\Sigma_{t,k,j}^g$ , the background cross section above becomes a constant over group  $g$  as

$$\sigma_{t,r,i}^{0g} = \frac{\sum_j \sum_{k \neq r} P_{ji}^g \Sigma_{t,k,j}^g V_j}{\sum_j P_{ji}^g N_{r,j} V_j}. \quad (2.8)$$

Then, the escape cross section of the resonant isotope  $r$  at region  $i$  can be simply determined by

$$\Sigma_{t,r,i}^{eg} = N_{r,i} \sigma_{t,r,i}^{0g} - \sum_{k \neq r} \Sigma_{t,k,i}^g. \quad (2.9)$$

The escape cross section is an isotope-dependent quantity but it becomes region-dependent when the same isotope is not present at other regions. In addition, it is insensitive to the resonance interference effect. The current formulation for the escape cross section in MC<sup>2</sup>-3 has been replaced by Eq. (2.9).

Since Eq. (2.8) is based on collision probability, it is very costly to apply it for a whole core calculation. However, it is possible to solve the following two fixed source problems for the resonant isotope  $r$  as proposed by Yu [23]

$$\Omega \cdot \nabla \psi_{n,r}^g(r, \Omega) + \Sigma_t^g(r) \psi_{n,r}^g(r, \Omega) = \sum_{k \neq r} \Sigma_{tk}^g(r), \quad (2.10)$$

$$\Omega \cdot \nabla \psi_{d,r}^g(r, \Omega) + \Sigma_t^g(r) \psi_{d,r}^g(r, \Omega) = N_r(r),$$

which allow the estimation of the background cross section in Eq. (2.9) without using collision probabilities as

$$\sigma_{0,i,r}^g = \frac{\int_{V_i} dV \int_{4\pi} d\Omega \psi_{n,r}^g(r, \Omega)}{\int_{V_i} dV \int_{4\pi} d\Omega \psi_{d,r}^g(r, \Omega)} = \frac{\phi_{n,r}^g}{\phi_{d,r}^g}. \quad (2.11)$$

Using Eqs. (2.9) and (2.10), the escape cross sections can be determined by performing  $2 \cdot N$  whole-core transport calculations where  $N$  is the number of resonant isotopes. One to two more iterations may be expected to converge the total cross sections. As illustrated in Figure 2.2, the fixed source transport calculation provides the UFG escape cross sections by iterations, with which MC<sup>2</sup>-3 finally determines the UFG or BG region-wise resonance cross sections to be used in the whole-core transport calculation.

Verification tests of the new resonance self-shielding method were performed using 2D multi-pin problems. The compositions for those problems were obtained from typical fast reactor systems: fuel, cladding, coolant, and reflector materials. For simplicity, only six

isotopes were used: U-235, U-238, O-16, Cr-52, Fe-56, and Na-23. Assemblies were composed of 3 fuel pins and 3 reflector material pins. The fuel pin locations were varied to make the pin boundary conditions different between assembly types, as shown in Figure 2.3. Since the combination of fuel and reflector material pins is kept the same for the assemblies, the 1D pin-cell model is equivalent to all of the 2D assemblies in terms of region and composition, illustrated in Figure 2.4.

There are different ways to make the 1D pin cell with cylindrical rings equivalent to the 2D problems. In this study, the 1D pin cell was made by adding two more regions for reflector and assembly gap regions to a single pin cell. The radii of regions were determined to preserve the volumes of each region. In the 2D problems, fuel and reflector pins are uniformly arranged in 120 degree symmetry for Type A, whereas fuel pins are concentrated in part of the assembly for Types B and C. Those variations were motivated to make local self-shielding conditions different from those of the standard 1D pin cell. Those types are not realistic assembly geometries but may represent some assembly configurations in a core.

The UFG cross sections were generated using the HFG cross sections and the NR approximation with the isotope and region dependent background cross sections discussed in Section 2. The escape cross sections for the 2D problems were calculated by solving the fixed source transport problems using a 2D MOC code (we modified DeCART for this purpose instead of PROTEUS-SN to obtain test results within the limited work schedule), whereas those for the 1D problem were determined using the MC<sup>2</sup>-3 code with the 1D collision probability method. The calculated escape cross sections were used to determine the UFG unresolved and resolved cross sections.

With the effective UFG cross sections, the eigenvalue of the 1D problem was calculated using the TWODANT code [24] with S<sub>16</sub> angular discretization and P<sub>3</sub> anisotropic scattering order. The resulting eigenvalue agreed within a few tens pcm with those obtained from MC<sup>2</sup>-3 with CPM. For the 2D MOC calculations, the ray tracing was performed with a ray spacing of 0.02 cm and 72 azimuthal and 8 polar angles. Reference calculations were performed using MCNP5 with 5 million particle histories so that the standard deviation of eigenvalue should be around 20 pcm or less. For the 1D problem, the white boundary condition was selected for both TWODANT and MCNP5.

As the first step of comparison, the eigenvalue calculations for the homogeneous mixture were conducted using MC<sup>2</sup>-3 and MCNP5. The eigenvalue of MC<sup>2</sup>-3 was 121 pcm  $\Delta k$  higher than that of MCNP5 for the homogenous mixture which is consistent with previous observations in the code assessment that MC<sup>2</sup>-3 tends to overestimate the eigenvalue in comparison with MCNP5. In the next step, the eigenvalues for the cylindrical and hexagonal pin cells were calculated using the two codes to compare the heterogeneity effect. As seen in Table 2.1, there is only a small difference of 30 pcm  $\Delta k$  in eigenvalue between the cylindrical and hexagonal geometries. However, the heterogeneity effect from MC<sup>2</sup>-3 or the new self-shielding method is underestimated compared to that from MCNP5.

The escape cross sections are basically region-dependent and thus they should be identical for all isotopes in a region unless the isotopes exist in the other regions. However, if the isotope of interest is present in more than one region, its escape cross sections are different from the region-dependent ones. Figure 2.5 shows how different the UFG escape cross sections of Fe-56 are between Regions 3 (coolant), 4 (reflector), and 5 (assembly gap) in its

resolved resonance energy range (up to 850 keV). The escape cross sections of Fe-56 are similar between Regions 3 and 5 (negative: neutron leak-in) which are quite different from Region 4 (positive: neutron leak-out). Since Fe-56 is present in three regions, its escape cross sections should be different from the region-dependent ones of each region.

The escape cross sections of U-238 for its resonance energy range (up to 149 keV) are plotted in Figure 2.6, which are noticeably different from each other between the 1D case and Types A and B of the 2D case. There is no noticeable difference in the escape cross section and eigenvalue between Types B and C. The large peak at 2.8 keV is produced due to the resonance of Na-23 in the surrounding regions. As shown in Table 2.1, the eigenvalues for Types B and C are more accurate when the escape cross sections are calculated with the explicit 2D geometry. It is noted that for Types B and C, the difference in the escape cross sections between 1D and 2D calculations makes about twice as much contribution to the eigenvalue variation than the difference in the spatial flux shape. The eigenvalue variation due to the change in energy self-shielding is mainly caused by the energy self-shielding of U-238 cross sections in the fuel region.

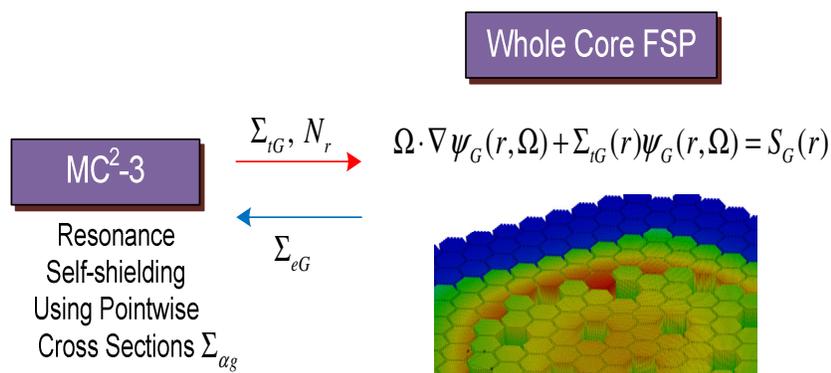


Figure 2.2 Resonance Self-shielding for Whole Core Problems

Table 2.1 Comparison of Eigenvalues for 1D and 2D Problems with Different Escape Cross Sections

Geometry		MCNP5	1D Code		2D Code ( $\Delta k$ , pcm)	
			( $\Delta k$ , pcm)	1D XS <sup>a</sup>	2D XS <sup>b</sup>	
Homogeneous		1.23019 ± 0.00020		121		
Heterogeneous Pin-cell	Cylinder	1.23610 ± 0.00019	-10			
	Hexagon	1.23640 ± 0.00020		-15		-19
Heterogeneous 2D Assembly	A	1.23663 ± 0.00020	-63	-22		66
	B	1.23836 ± 0.00019	-236	-148		25
	C	1.23813 ± 0.00022	-213	-138		25

a. The escape cross sections determined with the 1D pin-cell geometry

b. The escape cross sections determined with the 2D geometry

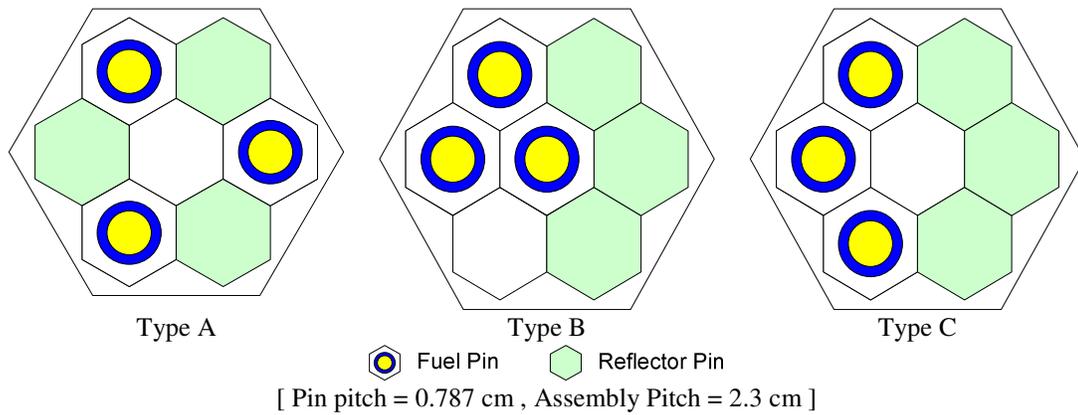


Figure 2.3 2D Multi-Pin Problems with Fuel Pins and Reflector Compositions

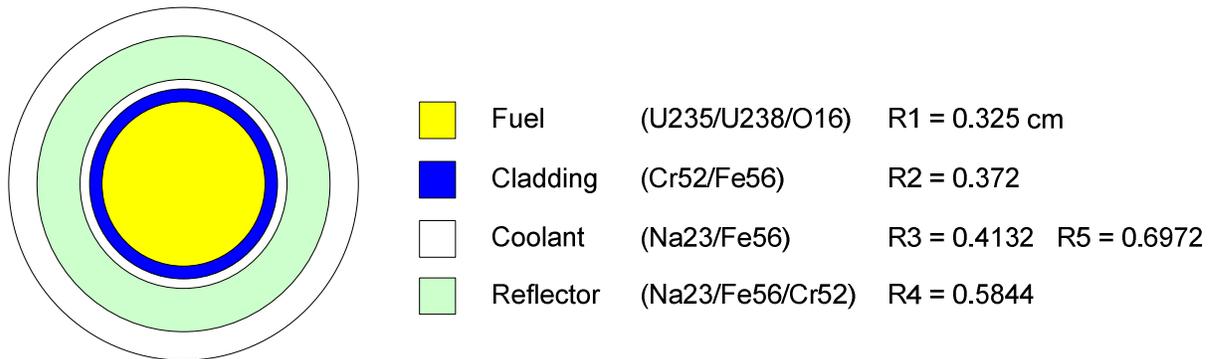


Figure 2.4 1D Problem Equivalent to the 2D Problems

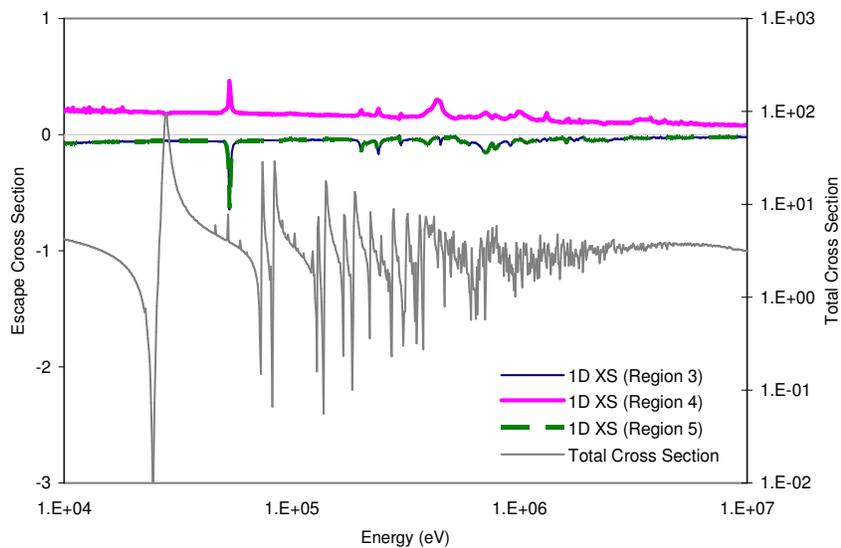


Figure 2.5 UFG Escape Cross Sections of Fe-56 Generated from the 1D Problem

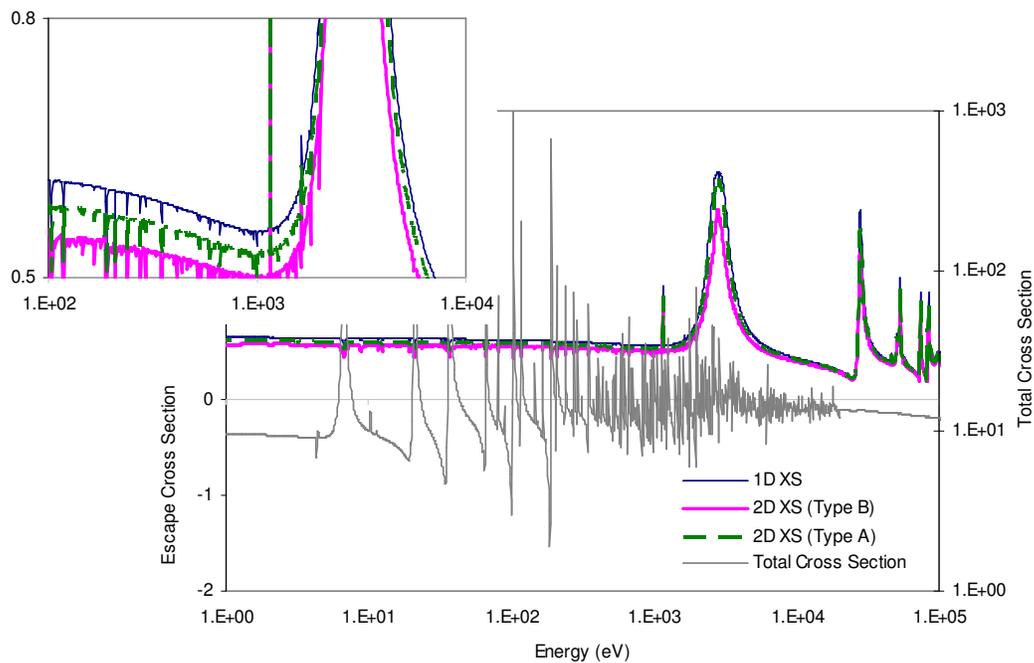


Figure 2.6 UFG Escape Cross Sections of U-238 Generated from the 1D and 2D (Types A and B) Problems

### 2.1.4 Other Method

The conventional subgroup method occasionally suffers from negative subgroup parameters when used in the non-uniform temperature distribution. Since the RI table method does not need the subgroup parameters, it has no such negative parameter or cross section problem. Both methods require the Bondarenko iteration to approximately account for the resonance interference effect. The embedded self-shielding method (ESSM) [15] proposed by ORNL is able to take into account the resonance interference effect simultaneously while solving the fixed source problem (FSP). While the subgroup method solves the FSP with given subgroup cross sections and sources for all subgroup levels, the RI table method and the ESSM requires iterations until the total cross sections in the FSP are converged.

In the RI table method, the following one-group fixed-source transport equation for an isotope  $i$  is solved with the infinite dilute cross section as a starter

$$\Omega \cdot \nabla \psi(r, \Omega) + (\Sigma_a^i(r) + \lambda \Sigma_p(r)) \psi(r, \Omega) = \lambda \Sigma_p(r), \quad (2.12)$$

where  $\lambda$ ,  $\Sigma_p$ , and  $\Sigma_a^i$  are the intermediate resonance parameter, potential cross section, absorption resonance cross section of the isotope of interest. Note that  $\lambda \Sigma_p = \sum_j \lambda \Sigma_p^i$ . The background cross section is evaluated as

$$\Sigma_b^i(r) = \frac{\Sigma_a^i(r) \phi(r)}{1 - \phi(r)}, \quad (2.13)$$

where  $\phi$  is the flux based on Eq. (2.1) which should be less than unity. Once the background cross section is determined, the absorption cross section is updated by the interpolation based upon a pre-calculated table available from the cross section library. In the iterative method, Eq. (2.12) is solved and followed by the interpolation until the absorption cross sections converge. Since the convergence is typically achieved within a few iterations, the computational load is similar to the conventional subgroup method which normally includes 4 to 7 subgroup levels.

On the other hand, the ESSM solves the following equation which includes the absorption cross sections for all isotopes in the second term of the left hand side

$$\Omega \cdot \nabla \psi(r, \Omega) + (\Sigma_a^i(r) + \Sigma_a^{other}(r) + \lambda \Sigma_p(r)) \psi(r, \Omega) = \lambda \Sigma_p(r), \quad (2.14)$$

where  $\Sigma_a^{other} = \sum_{j \neq i} \Sigma_a^j$ . The background cross section is calculated in the same manner as Eq. (2.13)

$$\Sigma_b(r) = \frac{(\Sigma_a^i(r) + \Sigma_a^o(r)) \phi(r)}{1 - \phi(r)}. \quad (2.15)$$

Note that the background cross section above represents the mixture for region  $r$  whereas the one in Eq. (2.13) is for a single isotope. The escape cross section  $\Sigma_e$  can be deduced as  $\Sigma_b - \lambda \Sigma_p$ . While the isotopic escape cross sections within the region can be different from each other in Eq. (2.13), they are identical for all isotopes of the region in Eq. (2.15). Thus, in

the ESSM it is difficult to separate the isotope-wise escape cross sections out from the region-wise escape cross section.

For comparison tests, the 230-group escape cross sections were calculated from two pin-cell problems: one with U-238/U-235 surrounded by Fe-56/C-12 and the other one with U-238/U-235 surrounded by Fe-56/U-238 so that both regions have the same isotope of U-238. The escape cross sections were estimated by three different approaches: the conventional subgroup method, the ESSM, and the MC<sup>2</sup>-3 method. The calculation by the MC<sup>2</sup>-3 method was added to derive the reference escape cross sections accounting for the resonance interaction effect, which can be different from those of the subgroup method and the ESSM.

As seen in Figure 2.7, for the case that U-238 is present only at the inner region, the escape cross sections of U-238 from all methods are similar to each other. However, the escape cross sections are noticeably different for the other case where U-238 is present at both regions. In particular, the escape cross sections of the ESSM are different from those of the other two methods. This is because the ESSM provides region-wise escape cross sections while the other methods produce isotope-wise ones. This suggests that it should be careful to use the escape cross sections generated from the ESSM when modeling multi-pin cells in which the same resonant isotope appears more than one region.

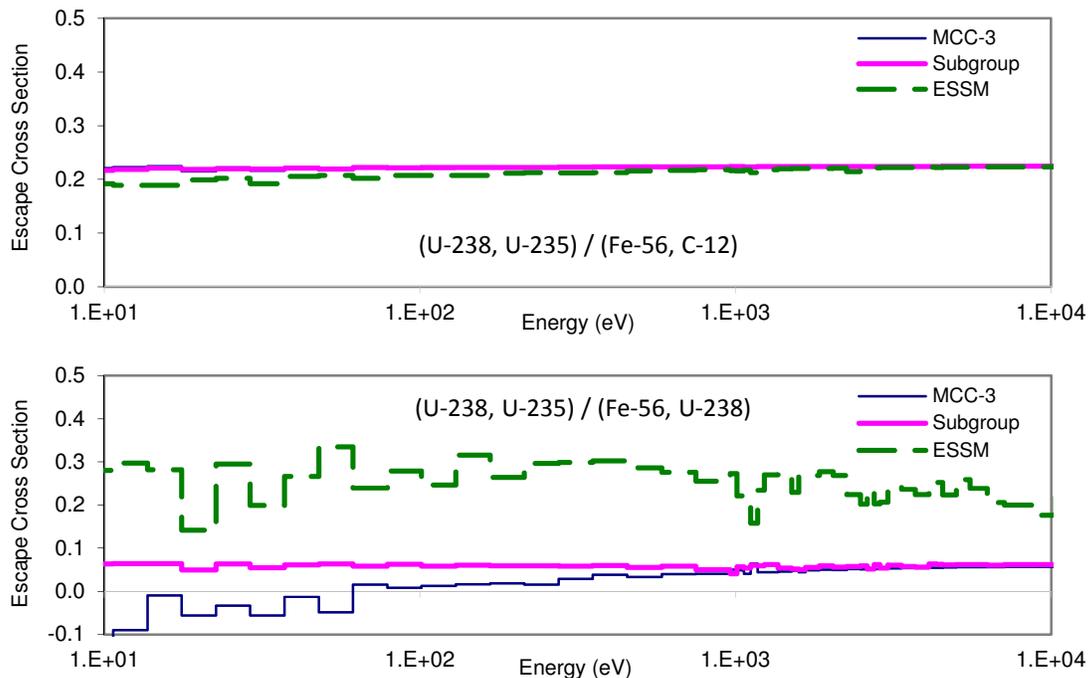


Figure 2.7 230-Group Escape Cross Sections of U-238 Estimated by Three Different Methods for Two-region Pin-Cell Problems

## 2.2 Development of a Generalized Cross Section Library

### 2.2.1 Introduction

The goal of NEAMS neutronics is to develop a high-fidelity deterministic neutron transport code for use on all reactor types of interest, primarily on sodium-cooled fast reactors. Since we targeted light water reactors in FY12, we started looking at the subgroup method for generating cross section libraries. However, the approximate treatment of the resonance interference effect in the conventional subgroup method was not acceptable to achieve the desired accuracy in fast reactor systems where resonance interferences exist between actinides and between actinides and intermediate-weight isotopes (structural material). As a consequence, we developed the direct resonance self-shielding method in which the MC<sup>2</sup>-3 methodology was extended from 1D to a whole-core application. However, the ultrafine group whole-core, fixed source transport calculations to determine the escape cross sections require significant computational effort for 2D and 3D problems with an actual reactor size. Therefore, we need to find a way to approximate the ultrafine group escape cross sections from the broad group ones. Extending the MC<sup>2</sup>-3 method and library to thermal reactor application requires significant effort as well. From our previous experience with thermal system cross sections, generating cross sections in the MC<sup>2</sup>-3 method may not be necessary for thermal systems where the detailed treatment of reactions in the high energy range is dispensable. This motivated us to develop a new resonance self-shielding methodology which is simple but accurate and can be applied to various reactor types.

Last year, we spent our effort on calculating subgroup parameters accurately by introducing MCNP5 as a reference resonance cross section calculation tool to the cross section library generation procedure. This is because we observed that the 238 group cross section library (the subgroup cross sections) for VHTR generated from the SCALE/CENTRM system [16] worked well for the NGNP/VHTR cores but produced large errors for HTTR and VHTRC [25]. This new procedure requires an additional set of MCNP5 ACE libraries with exclusion of absorption cross sections so that only a single isotope of interest in the mixture contains an absorption resonance cross section. For each isotope, many MCNP5 fixed-source calculations for a pin cell with different temperatures and number densities are necessary to obtain reference resonance cross sections as a function of temperature and background cross section. In the fixed-source calculations with MCNP5, isotopic absorption and nu-fission reaction rates as well as neutron fluxes are tallied to obtain absorption and nu-fission cross sections. The background cross sections for a pin cell at a given temperature and composition condition are estimated by solving the fixed-source neutron slowing-down equation in the deterministic way.

Figure 2.8 illustrates a procedure of generating a cross section library using the GeneCS code. The code produces the NJOY inputs to generate two different sets of ACE libraries for each isotope (a normal set and a set without absorption cross sections). The NJOY calculations are performed to produce the smooth multigroup cross sections as well as the resonance cross sections for different temperatures and background cross sections. The MCNP5 calculations are conducted to provide reference multigroup resonance cross sections for homogeneous mixture and pin cell geometry. Using those outputs from NJOY and MCNP5 calculations, the GeneCS code generates the multigroup cross sections and subgroup parameter tables as a function of temperature and background cross sections for each isotope.

The code also solves the 1D fixed-source neutron slowing-down equation and performs the least square fitting to determine subgroup parameters.

For tests, a set of the 72-group multi-group cross sections and parameters for U-238 were generated using the new procedure to update the existing subgroup cross section library. Then, DeCART was executed for a typical VHTR pin-cell problem (a hexagonal fuel pin with U-235/U-238/O/C/Si surrounded by graphite) using both the existing (old) and new cross section libraries. As a reference solution, the MCNP5 calculation was performed for the same pin-cell case, tallying the partial cross sections of U-238 as well as eigenvalue. Note that the old set of U-238 was generated via the CENTRM reference solutions.

Figure 2.9 shows the comparison of the 72-group absorption cross sections of U-238 between the old and new cross section libraries and MCNP5 cross sections. Although the new cross section of U-238 gives relatively better agreement with MCNP5 cross sections than the old one, the new eigenvalue was much worse, an underestimation of more than 500 pcm, than that from the old cross section library, compared to the MCNP5 solution. The large underestimation was caused by the larger absorption cross sections of U238 than those from the old library. This cannot be explained other than the existing (old) subgroup library has noticeable error cancellation between the resonance cross sections and the other cross sections. This suggests that a more rigorous cross section methodology and procedure is necessary to accurately estimate resonance cross sections and other cross sections for various reactor types with different neutron spectra.

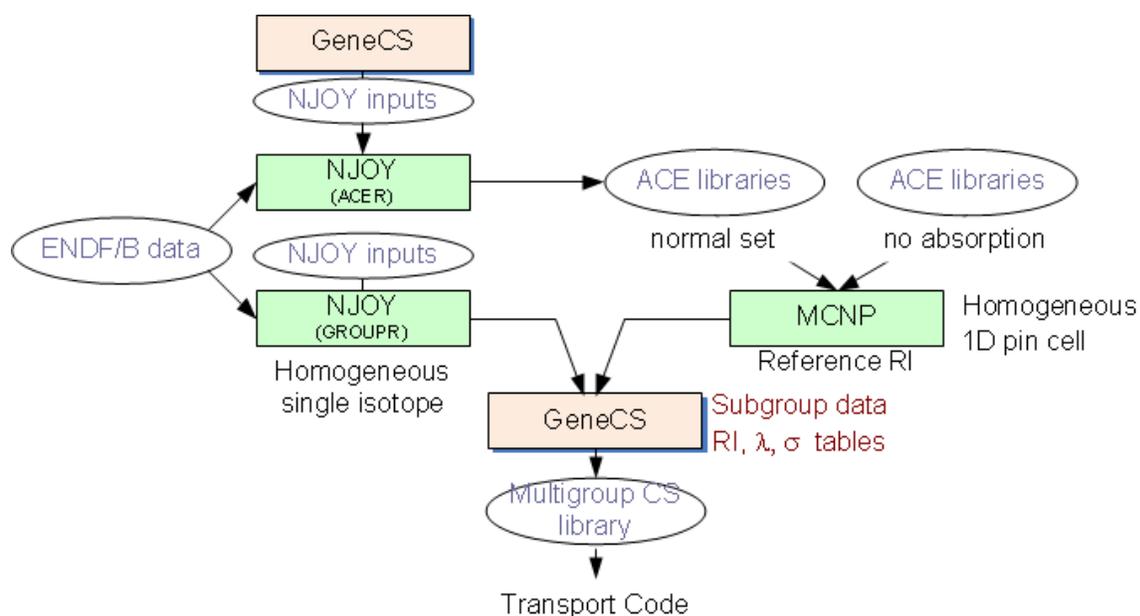


Figure 2.8 A Procedure of Generating a Cross Section Library Using the GeneCS Code

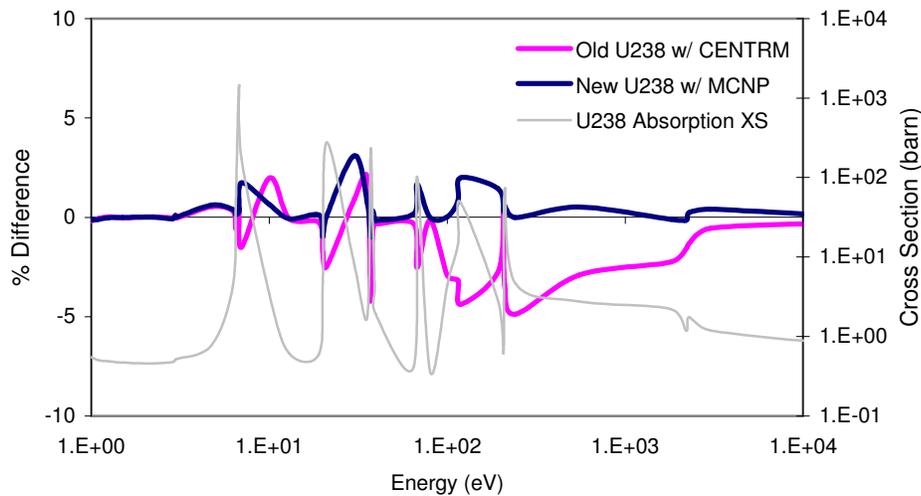


Figure 2.9 Difference of 72-Group Absorption Cross Sections of U-238 between DeCART and MCNP5 for a Typical VHTR Pin-cell

### 2.2.2 Methodology

From preliminary investigations on the resonance self-shielding methods, we decided to use the resonance integral (RI) table method which is more simple and robust to determine the resonance table rather than the subgroup method which needs the least square fitting to define subgroup parameters. However, there is no distinct superiority in accuracy between the two methods.

As discussed in the previous section, the resonance cross sections for absorption, nu-fission, and scattering are tabulated as a function of background cross section and temperature for the RI table method. It should be noted that the scattering resonance cross section which is often missed in the conventional subgroup library plays an important role in determining the neutron spectrum accurately. Those three resonance integrals are based on the flux approximated with the absorption cross section only as defined in Eq. (2.1). Although the flux approximation is different from the actual flux, it should be fine since the resonance versus background table is determined to reproduce reference solutions.

The observations discussed in the previous section indicated that the resulting multigroup cross sections with more accurate representation of resonance cross sections often showed better agreement on partial cross sections but rather worse agreement on eigenvalue with the reference solutions. This is partly because the group condensation was not accurately conducted for scattering cross sections and matrices.

To minimize errors due to the group condensation procedure, one can increase the number of groups to some extent. Based on the MC<sup>2</sup>-3 code development experience for fast reactor systems, the ultrafine groups (UFGs), ~2000 groups, should be a good starting point for generating a base cross section library. We understand that thermal reactor systems do not need that many groups, but fast reactor systems need the UFG cross sections to accurately capture severe spectrum transitions between distinctly different material regions.

Because the UFGs may be too many for a whole-core transport calculation, it needs to be reduced to a practical number of groups with an appropriate UFG neutron group spectrum. Thus, we propose a way of reducing an UFG library to a broad group (BG) library while minimizing the loss in accuracy. Once a reactor of interest is selected, the UFG transport calculations are first performed with a representative homogeneous composition to determine the UFG neutron spectrum. Next, various possible compositions are used to determine the BG boundaries to best approximate the solutions with the UFG cross section library in terms of partial multigroup cross sections and eigenvalue. This involves an optimization process for group condensation, which will be discussed later in this section. This would be similar to the way that the subgroup parameters are determined with many possible fuel compositions and pin cell geometries with various temperature and background cross section conditions.

Once the BG structure is determined, the UFG cross section library is reduced to the BG cross section library using the representative UFG neutron spectrum. This should be differentiated from the conventional cross section library generation practices in that it is not a process of generating a cross section library from a raw nuclide data but a process of reducing an UFG cross section library to a BG library. So, all the resonance tables should be reducible without losing accuracy. This is one of the reasons why we selected the RI table method instead of the subgroup method whose parameters cannot be reduced by a simple group condensation process. With the cross section table method, the UFG resonance cross sections are simply condensed as

$$\bar{\sigma}_G(\sigma_b, T) = \frac{\sum_{g \in G} \sigma_{\alpha g}(\sigma_b, T) \phi_g^{ref}}{\sum_{g \in G} \phi_g^{ref}}, \quad (2.16)$$

where  $\alpha$  is one of absorption, nu-fission, and scattering, and  $\phi_g^{ref}$  is the representative UFG neutron spectrum for a specific reactor or reactor type. To make the table condensation simple, the same background cross section knots are used over all resonance UFGs.

An optimized group condensation process is one of the key steps in the proposed cross section library generation methodology. An algorithm for the group structure optimization was devised to search for a BG structure which is insensitive to any change of UFG spectra within a BG. The allowance of sensitivity will be determined by a user as a group optimization criterion. There are two ways to check. One is to check the difference of production and absorption reaction rates based on the equation below.

$$\frac{\Delta k}{k} = \frac{\Delta P}{P} - \frac{\Delta A}{A}, \quad (2.17)$$

where  $P$  and  $A$  are the total production and absorption reaction rates, respectively and  $k$  is an eigenvalue. The optimized group condensation process checks the change rates of both production and absorption reaction rates with expanding a group boundary of broad group  $G$  from the first group.

Suppose that there are  $N$  compositions in association with a target reactor type, and the UFG cross sections and flux solutions for all compositions are given along with the representative neutron spectrum which can be the flux solution of one of the compositions or one that is provided independently of the compositions. Starting from the first group, merge

the neighboring fine group into the broad group  $G$  to calculate the BG reaction rates using the UFG fluxes for all  $N$  compositions, which are the reference reaction rates

$$R_{\alpha G} = \sum_{g \in G} \Sigma_{\alpha g} \phi_g . \quad (2.18)$$

Perform the same thing with the representative flux and calculate the BG cross sections this time which will be multiplied by the reference flux to produce the test reaction rates as

$$\bar{R}_{\alpha G} = \sum_{g \in G} \Sigma_{\alpha g} \phi_g^{rep} / \sum_{g \in G} \phi_g^{rep} \times \sum_{g \in G} \phi_g . \quad (2.19)$$

Check the absolute difference ratio between the reference and test reaction rates,  $\Delta = \left| \bar{R}_{\alpha G} - R_{\alpha G} \right| / R_{\alpha G}$  for absorption and production separately. If  $\Delta$  is less than the user-specified criterion, then move on to the next UFG and condense it into the same the broad group  $G$ . If  $\Delta$  is greater than the user-specified criterion at the fine group  $g$ , advance to the next BG evaluation,  $G+1$ , starting from the fine group  $g$ . Repeat the process until the last fine group is reached.

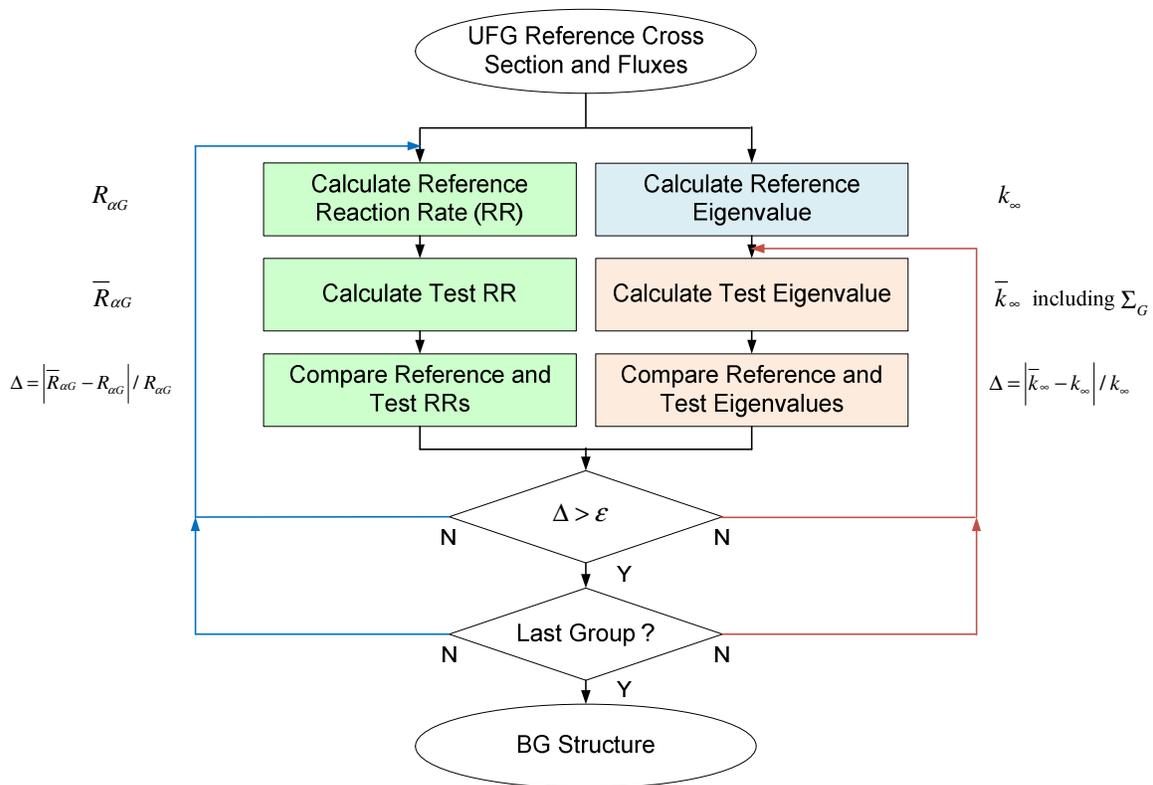


Figure 2.10 Group Optimization Algorithms from UFG to BG Libraries

This algorithm focuses on the change of two direct components which determine the eigenvalue with given neutron fluxes, but we found that it overlooks the contribution from the indirect component, scattering cross sections, which affects the neutron spectrum. The other

group optimization algorithm is to solve the eigenvalue problem and check  $\Delta k$  directly whenever adding a UFG to a broad group  $G$ . Since a focus of this algorithm is on the change of eigenvalue, this is more tolerant to the change of absorption and fission reaction rates when the change of both reaction rates moves toward the same direction but less tolerant for the other direction. Investigations indicated that the resulting broad groups from this algorithm are better than those from the first one, and therefore we selected the second algorithm for the group condensation in the work here. However, both algorithms could be utilized together to find more rigorous broad group boundaries. Figure 2.10 depicts the flow of the group optimization algorithms that have been discussed.

### 2.2.3 Library Generation

For generating the UFG resonance tables as a function of background cross section and temperature, the MC<sup>2</sup>-3 code was updated to specify new inputs and outputs and to allow a constant background cross section given from the input. The GeneCS code was also updated accordingly to read the new file output from MC<sup>2</sup>-3 and to store resonance cross sections as a table form.

Since the current version of MC<sup>2</sup>-3 is not capable of generating thermal cross sections, the NJOY code needs to produce UFG thermal cross sections and scattering matrices by adding THERMR to the inputs. The up-scattering cross sections are evaluated below 3 eV. The typical spectrum (Maxwellian + 1/E + fission spectrum) is used as a weighting function.

The GeneCS code reads the output files generated from both MC<sup>2</sup>-3 and NJOY to produce a complete set of the UFG cross section library in a DeCART library format (see Appendix A). The UFG structure is composed of total 2158 groups which include 2123 groups from 20 MeV to 0.414 eV divided into a constant lethargy interval of 1/120, and 35 groups from 0.414 to 10<sup>-4</sup> eV with variable lethargy intervals, as illustrated in Figure 2.11. Relatively large lethargies in the thermal energy range are assigned because there is no significant cross section variation like the resonance energy range.

Unlike ISOTXS, the current DeCART library has only total scattering cross sections which contain all of elastic, inelastic, (n,2n), and (n,3n) cross sections together. There are no separate (n,p), (n,d), (n,t), and (n, $\alpha$ ) reaction cross sections which are included in the absorption cross section. There is the transport cross section instead of the total cross section. The within-group terms of scattering matrices are corrected with total P<sub>1</sub> scattering cross sections. These need to be updated in the future to generalize the format.

$$\sigma'_a = \sigma_a - \sigma_{n2n} - 2\sigma_{n3n}, \quad (2.20)$$

$$\sigma'_{tr} = \sigma_t - \sigma_{s1},$$

$$\sigma'_s = \sigma_s + 2\sigma_{n2n} + 3\sigma_{n3n},$$

$$\sigma'_{sgg'} = \sigma_{sgg'} + 2\sigma_{n2n,gg'} + 3\sigma_{n3n,gg'} - \delta_{gg'} \sigma_{s1},$$

$$\sigma'_t = \sigma'_a + \sigma'_s = \sigma_a + \sigma_s + \sigma_{n2n} + \sigma_{n3n}.$$

where  $\sigma_{tr}$  is the transport cross section,  $\sigma_{s1}$  is the total P<sub>1</sub> scattering cross sections,  $\sigma'_s$  is the production-based scattering cross section, and  $\delta_{gg'}$  is the Dirac delta function. The total cross section can be retrieved by adding  $\sigma'_a$  and  $\sigma'_s$  as in Eq. (2.20).

The UFG cross section library is transformed to the BG cross section library using the group reduction and optimization process discussed in the previous section. The resulting BG structure depends upon the choice of the representative neutron spectrum and variable homogeneous or pin cell compositions. The user can provide the BG group structure and the representative neutron spectrum, just like the conventional approach. Figure 1.1 illustrates the flow of generating the generalized cross section library for the neutron transport code.

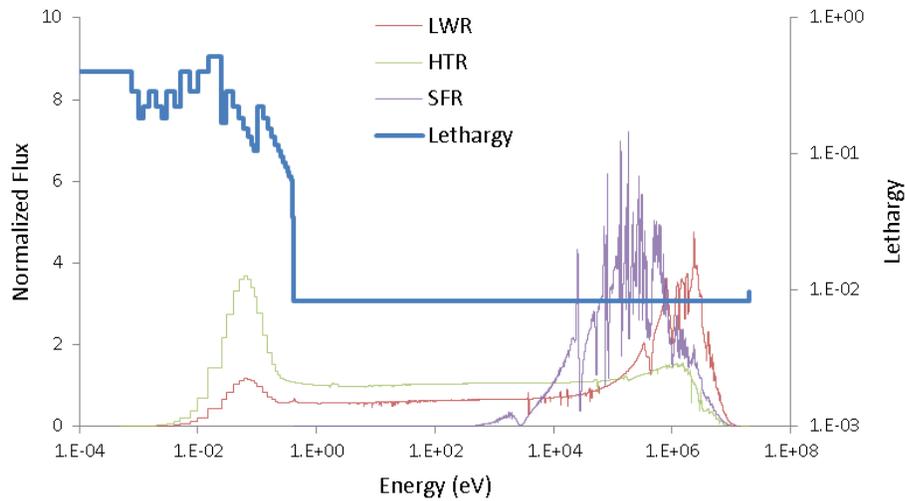


Figure 2.11 Lethargy Intervals of the Ultrafine Group Structure

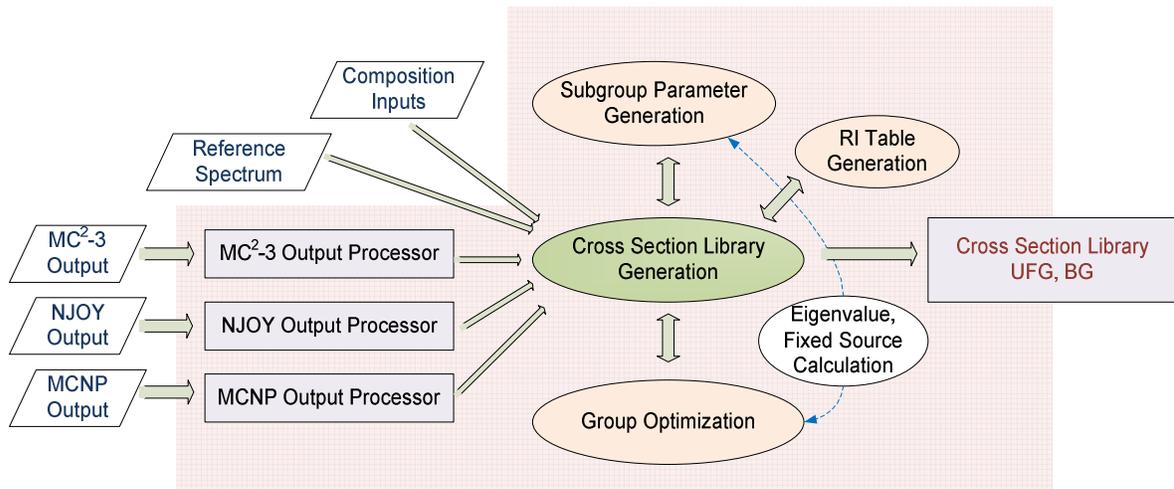


Figure 2.12 Cross Section Library Generation Using the GeneCS Code

## 2.3 Verification Tests

### 2.3.1 Preliminary Group Optimization Tests

Preliminary investigations were made for group optimization algorithms, discussed in the previous section. The reference calculations are performed using fine groups (950 groups) with the Monte Carlo code, McCARD [26], for various pin cell problems. As shown in Table 2.2, 8 VHTR pin cell configurations were selected by changing the enrichment from 5 to 30 wt%, the packing fraction from 10 to 50%, and the presence of burnable poison in the pin cells. For reaction rate comparison, absorption and nu-fission cross sections were tallied in the McCART calculation. Among the eight cases, Case 1 was selected as the reference fine group spectrum which is used for group collapsing. The double heterogeneity effect, which is one of the important factors in cross section generation, was not accounted for in the tests.

Table 2.2 VHTR Pin Cell Configurations for Group Structure Optimization

Case	Fuel	Enrichment (wt%)	Packing Fraction (%)
1	Fresh UO <sub>2</sub>	5	10
2	Fresh UO <sub>2</sub> with BP	5	10
3	Fresh UO <sub>2</sub>	5	50
4	Fresh UO <sub>2</sub> with BP	5	50
5	Fresh UO <sub>2</sub>	30	10
6	Fresh UO <sub>2</sub> with BP	30	10
7	Fresh UO <sub>2</sub>	30	50
8	Fresh UO <sub>2</sub> with BP	30	50

As aforementioned, the optimized groups are determined to satisfy the user-specified error criterion. Setting the error criteria to 1, 5, 10, and 20 pcm resulted in 215, 118, 85, and 65 groups, respectively. Note that the error in pcm is based on  $\Delta = \left| \bar{R}_{\alpha G} - R_{\alpha G} \right| / R_{\alpha G} \times 10^5$  per each broad group and thus it does not mean a difference in eigenvalue. Table 2.3 shows the group allocation of the energy group structures resulting from the group optimization process. Compared to the existing DeCART 190 groups (the last column) which were developed for its generic application to many different spectrum systems, the 215 groups ended up with much smaller number of fast groups and almost double resonance groups. This is because this optimization process was performed for VHTR pin cells and thus not many fast groups were necessary. The smallest group structure contains 65 groups, leading to only 20 pcm error from the reference solution with 950 groups.

Table 2.3 Optimized Energy Group Structures

Energy Range	215G	118G	85G	65G	DeCART 190G
Fast	16	8	7	5	63
Resonance	130	72	51	40	69
Thermal	69	38	27	20	58
Criterion [pcm] <sup>1)</sup>	1	5	10	20	N/A

1) Error criterion, for each broad group, from the McCARD reference solution with 950 groups

Figure 2.13 shows the group-wise macroscopic cross sections and the reaction rate errors in the resonance energy range for the conventional DeCART 190-group and new 215 group structure. One can observe from the figure that more energy groups are placed in the resonance peaks while less energy groups are placed in the resonance valleys between the peaks. Therefore, the total number of energy groups in the resonance energy range was almost doubled compared to the DeCART 190 groups. Note that the maximum error with the 190-group structure is as much as about 1200 pcm (red) while the maximum error with the new 215 group structure is less than 1 pcm (blue) for the eight test problems listed in Table 2.3.

Even though we used only the VHTR cases for preliminary group optimization tests, we can in principle find more generic group structures which can accurately cover a wide range of neutron spectrum if various fuel types (LWR, VHTR, and SFR) are used for test cases. As discussed previously, however, we had a problem in estimating eigenvalue accurately with the new broad group structures even though absorption and nu-fission cross sections in the resonance energy range agreed very well with those from McCARD. In other words, unlike the expectation from the within-group reaction rate sensitivity results, eigenvalues with the new broad group structures are underestimated by more than 500 pcm, compared to the Monte Carlo reference solutions.

This preliminary group optimization tests suggest the followings: 1) the 950 reference groups could be still insufficient to determine appropriate BG structures and 2) the group optimization algorithm based on the within-group reaction rate differences is not good enough to determine group boundaries which ensure accurate eigenvalue. This motivated us to propose the other group optimization algorithm, discussed in the previous section (see Figure 2.10), which performs an eigenvalue calculation at every group condensation to check eigenvalue changes by the group condensation. Test results will be discussed in details in the later section.

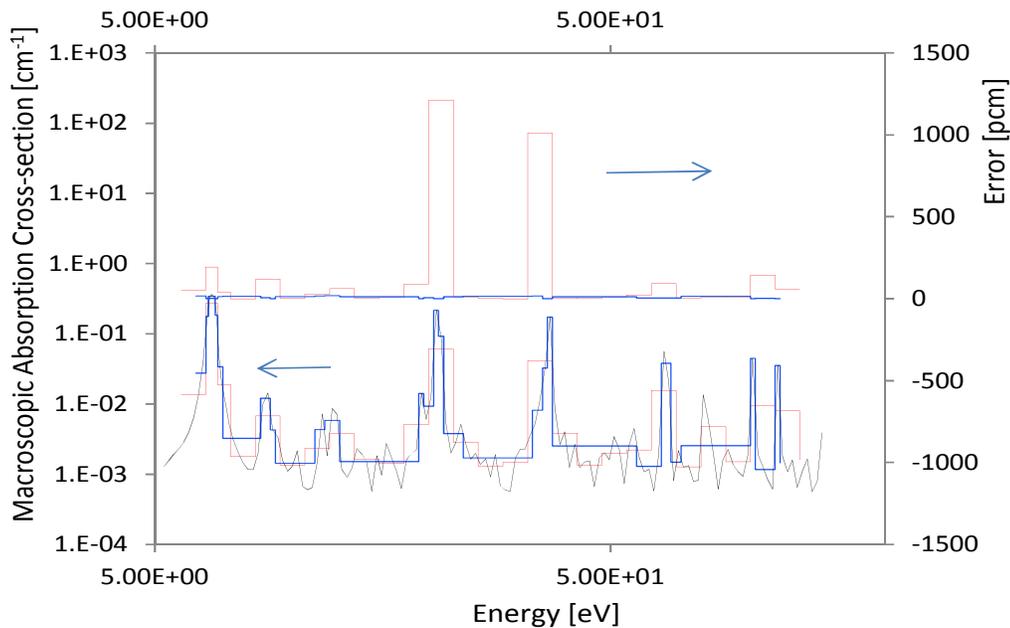


Figure 2.13 Multigroup Macroscopic Cross sections and Reaction Rate Errors [pcm] between the DeCART 190 Groups (red) and the New 215 Groups (blue)

### 2.3.2 Resonance Self-shielding Table Tests

Prior to generating cross section libraries, we spent significant efforts testing the cross section generation procedure to ensure that the selected cross section method is able to well reproduce the reference solutions, such as partial cross sections, reaction rates, and eigenvalue, for various reactor types and conditions. We have often observed that many of the existing cross section libraries give good results, ironically, due to large error cancellations. It is obvious that cross section methodologies which rely upon error cancellation will not be accurate for all possible reactor problems of interest.

Because of its simplicity and accuracy, we decided to use the resonance integral (RI) table method instead of the subgroup method which requires a significant effort for verification of the subgroup parameters. As mentioned previously, since the iteration for solving the fixed-source problem with the update of the resonance cross sections to determine the region-wise escape cross sections can be quickly converged in 2-3 iterations, there is no distinct difference in accuracy between the subgroup and RI table methods. Both methods require the Bondarenko iteration later on to account for the resonance interference effect.

A significant effort was made to investigate noticeable errors in the multiplication factors of test cases even though the new resonance self-shielding method and cross section library apparently estimated multigroup absorption and nu-fission cross sections better than the conventional subgroup approach. For more efficient investigation of the problem, we decided to look into a homogeneous composition rather than a heterogeneous pin cell.

As shown in Figure 2.14, the RI values from a homogeneous composition fall on the RI curve obtained from a heterogeneous pin cell. This suggests that tests can be carried out for homogeneous compositions and those results can be extended to the heterogeneous problem without losing accuracy.

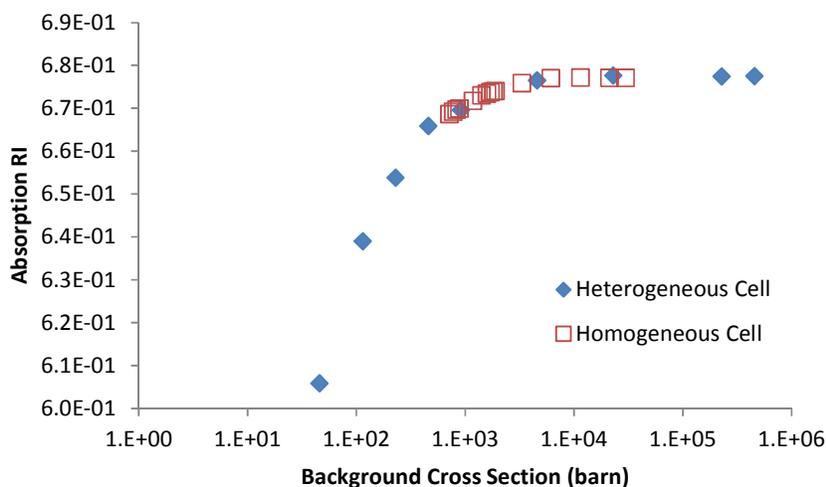


Figure 2.14 Absorption Resonance Integral versus Background Cross Section at a Resonance Group of U-238

Preliminary investigations on the resonance and background cross section table indicated that 16 background cross section points are necessary because of the steep gradient of the resonance versus the background cross section curve. Therefore, we produced the resonance tables of U-238 with 16 different background cross sections ranging from 0.1 to 1e10 barns at every resonance energy group so that the typical s-curve of resonance versus background cross section can be represented accurately. Currently, the same background cross sections were assigned for every resonance group to make the group condensation in the table form easy. Figure 2.15 shows examples of the background cross sections for U-238 at two different energies.

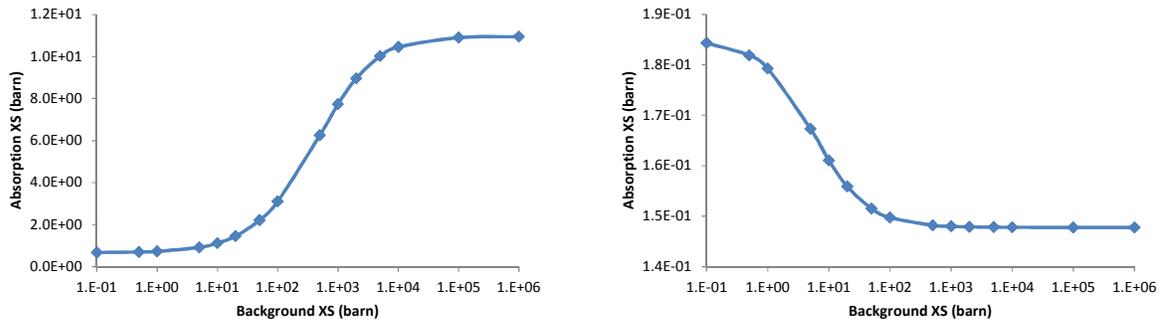


Figure 2.15 Resonance Cross Sections as a Function of Background Cross Section for U-238 at 1 keV (left) and 0.8 keV (right)

For an initial verification of the RI table, various homogeneous compositions were tested changing number densities such that the background cross sections of resonant isotopes vary from very small to almost infinite. As a test case, a typical fuel composition of VHTR was selected, in which only U-235 and U-238 were resonant isotopes. For the RI table of U-238, its number density was changed such that its background cross section can cover a wide range of the background cross sections (Case 1). A similar change in U-235 number density was made for the RI table of U-235 (Case 2).

According to previous observations, the resonance interference effect between U-235 and U-238 is dominant for thermal systems. To investigate the resonance interference effect between two major actinides, one cross section library was generated from MCNP5 by having resonance cross sections for only one resonant isotope of interest at a time for a mixture of multiple resonant isotopes, while another cross section library was generated by having the resonance cross sections of both U-235 and U-238 together in MCNP5 calculations. Note that cross section data other than resonances are prepared by NJOY calculations.

With the RI table generated with a single resonant isotope in a composition, the eigenvalues of all test cases agreed with the MCNP5 solutions within 254 pcm. When using the RI table produced with mixed resonant isotopes in a composition, the maximum eigenvalue error for most cases was reduced, except for a few cases (red). Table 2.4 shows the errors, in pcm, of the RI table method compared with the MCNP5 reference solutions. This is because the resonance cross section table already accounts for some of the detailed interference effect between the two actinides. However, it is not practical to consider the

interference effect in this way because one cannot consider numerous combinations of the number densities of two actinides. In addition, this approach will produce more errors when other actinides such as plutonium are present in the composition. As long as the Bondarenko method is used, this problem needs to be addressed to improve accuracy.

Table 2.4 Comparison of Eigenvalues of Various Homogeneous Compositions between MCNP5 and the RI Table Method

Case	Background Cross Section		MCNP5 <sup>1)</sup>	RI Table Method	
	U-235	U-238		Individual <sup>2)</sup>	Mixed <sup>3)</sup>
1	3490	58	0.44280	254	272
	3079	127	0.65519	-1	26
	2942	243	0.86047	-68	-10
	2873	474	1.06282	-165	-40
	2839	936	1.23987	-179	29
	2811	4636	1.53891	-59	132
	2806	23134	1.70068	80	67
	27947	230387	1.84899	39	38
	55880	460668	1.73383	14	18
2	298	474	1.61526	-30	11
	727	491	1.36542	-140	-52
	1442	480	1.19038	-191	-65
	5735	473	0.97151	-117	-2
	28630	472	0.74354	-91	-3
	57249	472	0.59630	-38	36
	560059	4617	0.51456	-17	35
	1118740	9222	0.36133	-24	-16

1) The standard deviations of MCNP5 solutions are within 55 pcm

2) The RI table is generated using a single resonant isotope in a composition

3) The RI table is generated using two resonant isotopes (U235 + U238) in a composition

### 2.3.3 Library Tests

Various compositions were selected for each reactor type of LWR, VHTR, and SFR to test the generalized cross section library. In order to have a wide range of background cross section for major actinide isotopes such as U-235, U-238, and Pu-239, the isotopic number densities in the compositions were changed. Details of test results for each reactor type will be discussed as follows.

#### Sodium Fast Reactor

For SFR, seven different compositions were selected as listed in Table 2.5. Among them, Cases 1, 3, and 7 came from the MC<sup>2</sup>-3 core benchmark problems: Cases 1 and 3 are composed of 21 and 16 wt% Pu contents, respectively, and structural material, while Case 7 is a typical blanket composition. For simplicity, only the isotope with the largest abundance ratio from elements, such as Fe, Cr, Ni, Mo, and Zr, was included in the compositions. Case 2 was fetched from the two-region benchmark problems of MC<sup>2</sup>-3 which were used to check the resonance interaction between U-238 and Fe-56. In Cases 4, 5, and 6, the number densities of U-238 and Pu-239 only were changed with maintaining the other isotopes' number densities (Fe-56, Na-23, and O-16) the same: the plutonium contents of the three

compositions are 16, 38, and 13 wt%, respectively. The neutron spectra for the compositions are shown in Figure 2.16. As seen in the figure, all cases show a typical neutron spectrum of fast reactors. Among them, Case 2 (red) shows an extreme change of neutron spectrum over energy due to a relatively large amount of Fe-56 included in the composition.

Table 2.5 Isotopes and Number Densities of Sodium Fast Reactor Compositions

Case	1		2		3		4		5		6		7	
Isotope & Number	U-235	1.11E-05	U-235	7.00E-04	U-235	1.44E-05	U-238	5.00E-03	U-238	7.00E-03	U-238	6.00E-03	U-235	1.59E-05
	U-238	5.49E-03	U-238	5.00E-03	U-236	9.20E-07	Pu-239	3.00E-03	Pu-239	1.00E-03	Pu-239	2.00E-03	U-238	7.53E-03
Density	Pu-239	9.11E-04	Pu-239	9.11E-04	U-238	9.05E-03	O-16	2.00E-02	O-16	1.60E-02	O-16	1.60E-02	Cr-50	1.67E-04
	Pu-240	3.52E-04	Fe-56	4.00E-01	Pu-239	1.57E-03	Fe-56	8.00E-03	Fe-56	2.00E-02	Fe-56	2.00E-02	Fe-56	1.26E-02
	Pu-241	1.03E-04			Pu-240	1.67E-04	Na-23	1.60E-02	Na-23	8.00E-03	Na-23	8.00E-03	Ni-58	1.84E-03
	Pu-242	5.79E-05			Pu-241	1.10E-05							Mo-92	4.54E-05
	Cr-50	3.22E-03			Pu-242	7.85E-07							Na-23	9.44E-03
	Fe-56	1.26E-02			Cr-50	1.05E-04							Mn-55	3.64E-04
	Ni-58	1.84E-03			Fe-56	1.49E-02							O-16	1.51E-02
	Mo-92	4.54E-05			Ni-58	6.80E-05								
	Na-23	9.44E-03			Mo-92	8.17E-05								
	O-16	1.37E-02			Na-23	7.15E-03								
					Mn-55	1.07E-04								
					Zr-90	1.68E-03								

Reference solutions were generated using MCNP5 and MC<sup>2</sup>-3. As shown in Table 2.6 the maximum difference in eigenvalue between the two codes is 144 pcm. The eigenvalue solutions from the RI table method with the 2158 group library (using the Bondarenko iteration for resonance self-shielding) agreed with MCNP5 solutions within 196 pcm. This accuracy in eigenvalue is much better than expected, but we noticed that the differences in the detailed UFG cross sections are not that small. This implies that the RI table method with the Bondarenko iteration resulted in good eigenvalue solutions due to some cancellation effect. This is similar to the observation that was made when testing the subgroup method for SFR compositions (see Section 2.1.1), with which the errors in the detailed broad group cross sections became reduced with a smaller number of energy groups.

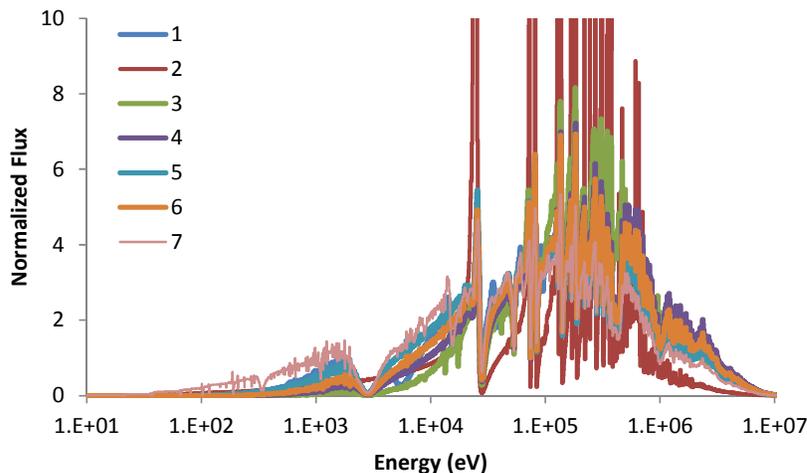


Figure 2.16 Neutron Spectra of Sodium Fast Reactor Compositions

The group condensation was performed using Cases 1, 2, and 4 as variable compositions and Case 5 as the representative UFG neutron spectrum composition. The criterion of  $\Delta k$  for each broad group was set to 5, 20, and 50 pcm in the group optimization process in GeneCS. The resulting broad group structures are 383, 219, and 142 groups for a  $\Delta k$  criterion of 5, 20, and 50 pcm, respectively. The eigenvalues from the broad group libraries agreed with the MCNP5 solutions within 225, 230, and 327 pcm  $\Delta k$ , as shown in Table 2.6. When using the existing group structures: the ANL 230 group structure, the DeCART 190 group structure, and the SCALE (ORNL) 238 group structure, most cases showed reasonably small differences in eigenvalue from the MCNP5 solutions but very large errors were observed in Case 2 (the most difficult case in terms of neutron spectrum transition) for all existing group structures and Case 3 for the ORNL group structure. The differences came mainly from the overestimation of Fe-56 scattering cross sections over 100 keV, which led to the softer neutron spectrum resulting in the large underestimation of eigenvalue.

Figure 2.17 compares the resulting broad group structures, from which the lethargy plots are shown from the smallest (top) to largest (bottom) number of groups. As seen in the figure, many groups were assigned to the energy ranges for actinide (tens – hundreds eV) and intermediate-weight (hundreds keV) isotope resonances. Figure 2.18 shows how the broad group structures are associated with U-238 resonances in the energy range between 1 eV and 1 keV. Group structures were determined to isolate large resonances.

Figure 2.19 shows the percent differences of total cross sections of U-238 between MC<sup>2</sup>-3 and the RI table method, which includes the results from 2158, 383, 219, and 142 group libraries. Note that while all the BG cross sections from MC<sup>2</sup>-3 are condensed from the 2158-group cross sections, those from the RI table method are the solutions from each BG transport calculation. Because of no overlapping of resolved resonances between different isotopes above 20 keV, the cross sections from the RI table method are in good agreement with those from MC<sup>2</sup>-3. However, large differences in the UFG cross sections between the two codes are observed below 2 keV for U-238 and Pu-238. The magnitudes of the differences become reduced significantly with smaller number of energy groups, which is apparently due to the cancellation effect.

Table 2.6 Eigenvalue Comparison for Sodium Fast Reactor Compositions

Case	MCNP5	MC <sup>2</sup> -3	UFG( $\Delta k$ )	BG ( $\Delta k$ )		
			2158G	383	219	142
1	1.35803	118	143	161	155	116
2	1.09527	-71	-169	-3	230	327
3	1.63860	-8	-196	-225	-230	-278
4	2.06932	28	-101	-95	-83	-113
5	1.24105	144	184	214	208	278
6	1.75072	64	-8	17	41	60
7	0.18888	54	73	88	32	-12

\* Standard deviation of MCNP5 solutions  $\leq$  30 pcm

\* The UFG and BG columns show differences in eigenvalue, pcm, from MCNP5

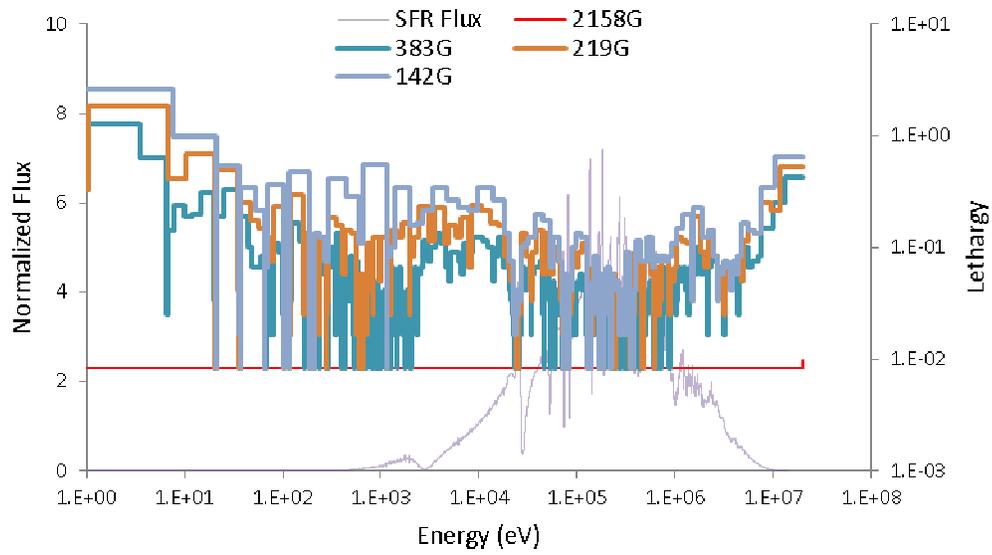


Figure 2.17 Lethargy Intervals of the Broad Group Structures for Sodium Fast Reactor

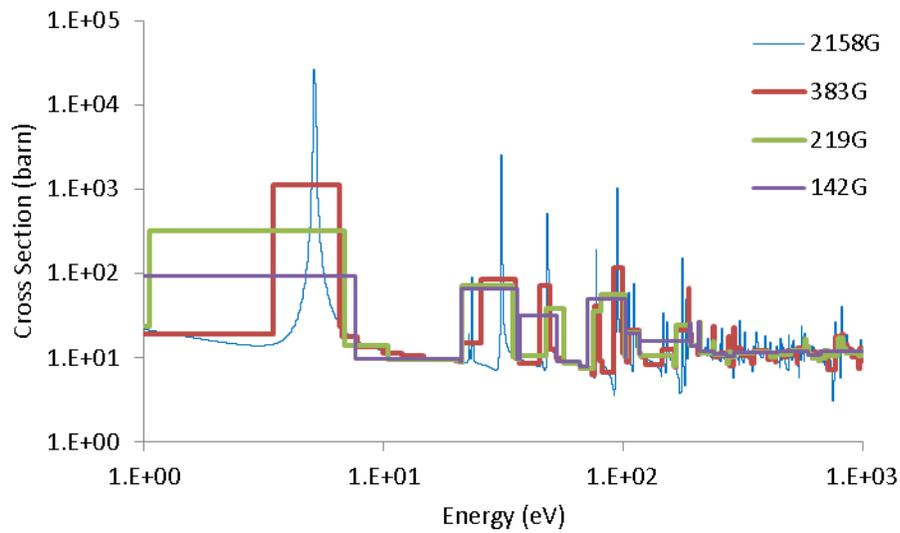


Figure 2.18 U-238 Total Cross Sections in Sodium Fast Reactor (Case 5)

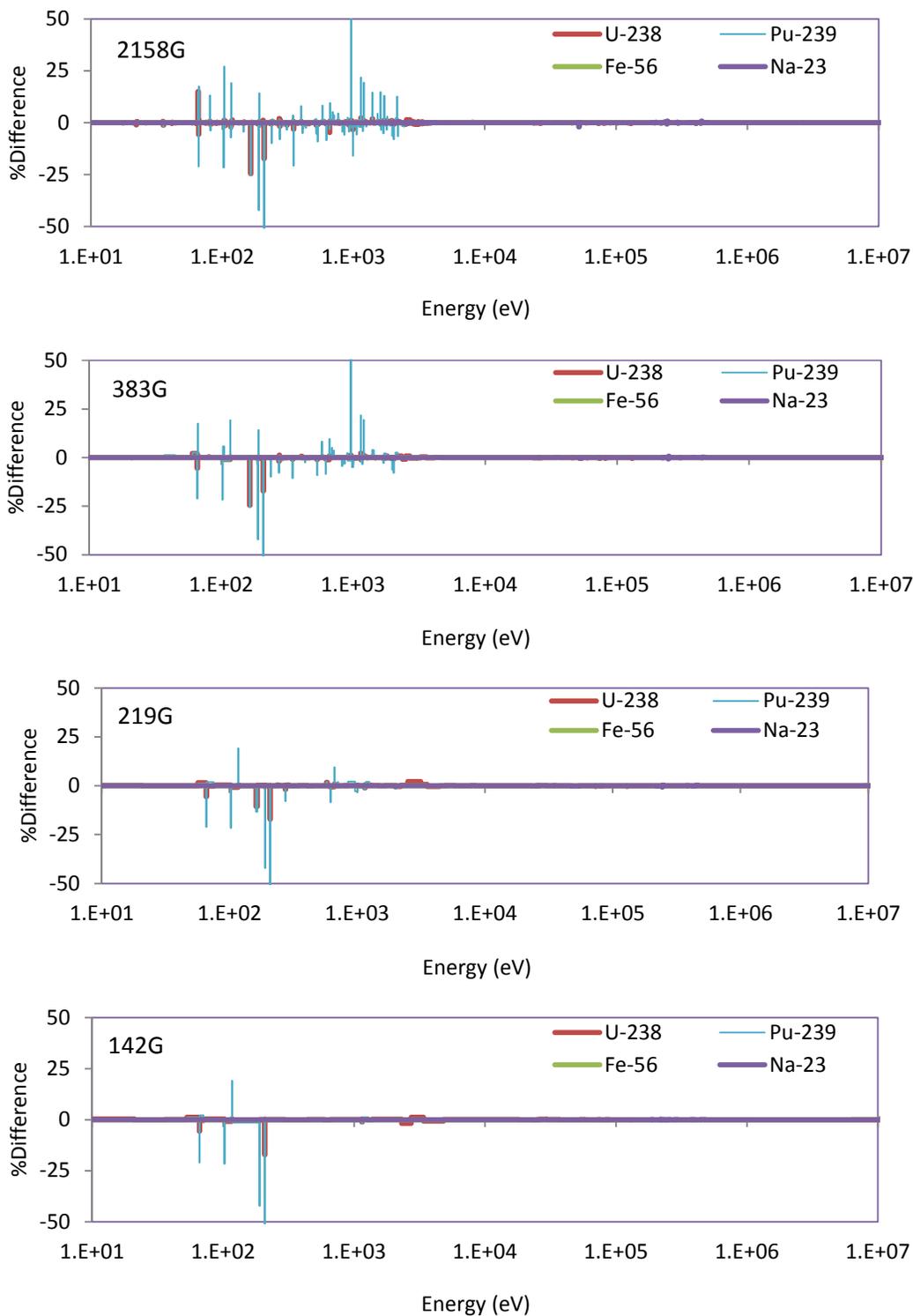


Figure 2.19 Differences of Total Cross Sections between MC<sup>2</sup>-3 and the RI Table Method in Sodium Fast Reactor (Case 5)

Light Water Reactor

For LWR, six different compositions were selected. The composition of Case 1 is a typical LWR fuel composition. The other cases were created by modifying the Case 1 number densities of U-235 and U-238. Consequently, U-235 enrichments were increased for Cases 2 (24 wt%), 3 (75 wt%), and 5 (96 wt%) and decreased for Cases 4 and 6 (almost natural uranium). These cases were created to provide different background cross sections to two major actinides, U-235 and U-238, in LWR, even though some of the compositions are unrealistic.

The compositions for the six test cases are listed in Table 2.7. Figure 2.20 depicts neutron spectra for those six cases. Cases 4 and 6, which have almost natural uranium, show higher thermal neutron fluxes while the other cases with U-235 enrichments of 3 to 96 wt% produces relatively higher fast neutron fluxes, as expected. The neutron flux wiggles are shown in the resonance energy range from a few eV to a few hundreds keV.

Table 2.7 Isotopes and Number Densities of Light Water Reactor Compositions

Case	1		2		3		4		5		6	
Isotope & Number Density	U-234	2.03E-06										
	U-235	2.38E-04	U-235	2.38E-04	U-235	2.38E-04	U-235	2.38E-05	U-235	2.38E-04	U-235	2.38E-06
	U-236	1.10E-06										
	U-238	7.36E-03	U-238	7.36E-04	U-238	7.36E-05	U-238	7.36E-03	U-238	7.36E-06	U-238	7.36E-03
	O-16	2.89E-02										
	Zr-90	4.28E-03										
	H-1	2.75E-02										

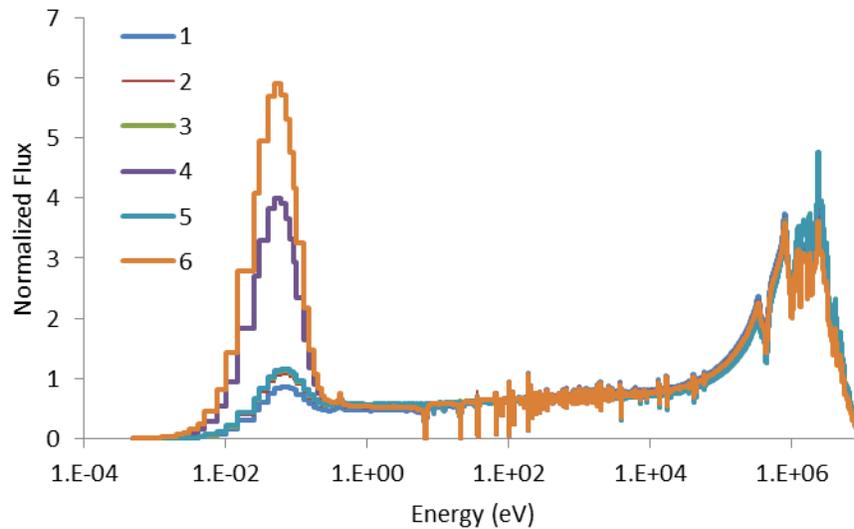


Figure 2.20 Neutron Spectra of Light Water Reactor Compositions

The group condensation was performed using Cases 1, 2, and 4 as variable compositions and Case 1 as the representative UFG neutron spectrum composition. In the same manner as SFR, the criterion of  $\Delta k$  for each broad group was set to 5, 20, and 50 pcm in the group optimization process of GeneCS. The resulting broad groups were 204, 120, and 78

groups for a  $\Delta k$  criterion of 5, 20, and 50 pcm, respectively. The resulting numbers of broad groups with the same stopping criteria for determining group boundaries are smaller than those for SFR. This is because coarser groups were determined in the high energy range which are more important to SFR but less important to LWR.

Reference solutions were generated using MCNP5. As shown in Table 2.8, the eigenvalue solutions from the RI table method with the 2158 group library (using the Bondarenko iteration for resonance self-shielding) agreed with MCNP5 solutions within 198 pcm. The eigenvalues from the broad group libraries with 204, 120, and 78 groups were off from the MCNP5 solutions by maximum 186, 144, and 183 pcm  $\Delta k$ , respectively. When using the existing group structures: the DeCART 190 group structure and the SCALE (ORNL) 238 group structure, Cases 1, 4, and 6 showed reasonably small differences in eigenvalue from the MCNP5 solutions, while very large errors were observed in Cases 2, 3, and 5 for all existing group structures, in which the U-238 number densities are relatively small and thus the U-235 enrichment becomes high. The differences came mainly from the large underestimation of U-238 absorption cross sections in the energy range between 6.7 and 100 eV which resulted in the large overestimation of eigenvalue. The underestimation of eigenvalue in Case 3 is smaller than in Case 2 since the number density of U-238 in Case 3 is smaller by an order of magnitude than in Case 2.

Similarly to the SFR cases, Figure 2.21 compares the resulting broad group structures, from which the lethargy plots are shown from the smallest (top) to largest (bottom) number of groups. Many groups were assigned to the energy ranges for actinides (tens – hundreds eV), as shown in the figure. Figure 2.22 shows how the broad group structures look like in association with U-238 resonances in the energy range between 1 and 100 eV. The broad group structures for LWR over the energy range have more groups than those for SFR. Group structures were determined to isolate large resonances.

Table 2.8 Eigenvalue Comparison for Light Water Reactor Compositions

Case	MCNP5	UFG( $\Delta k$ ) 2158G	BG ( $\Delta k$ )		
			204	120	78
1	1.28101	-85	-75	-90	-62
2	1.70407	-91	-186	20	183
3	1.83689	11	-88	28	109
4	0.58602	198	160	144	61
5	1.87309	108	25	109	140
6	0.15768	127	165	144	131

\* Standard deviation of MCNP5 solutions  $\leq 30$  pcm

\* The UFG and BG columns show differences in eigenvalue, pcm, from MCNP5

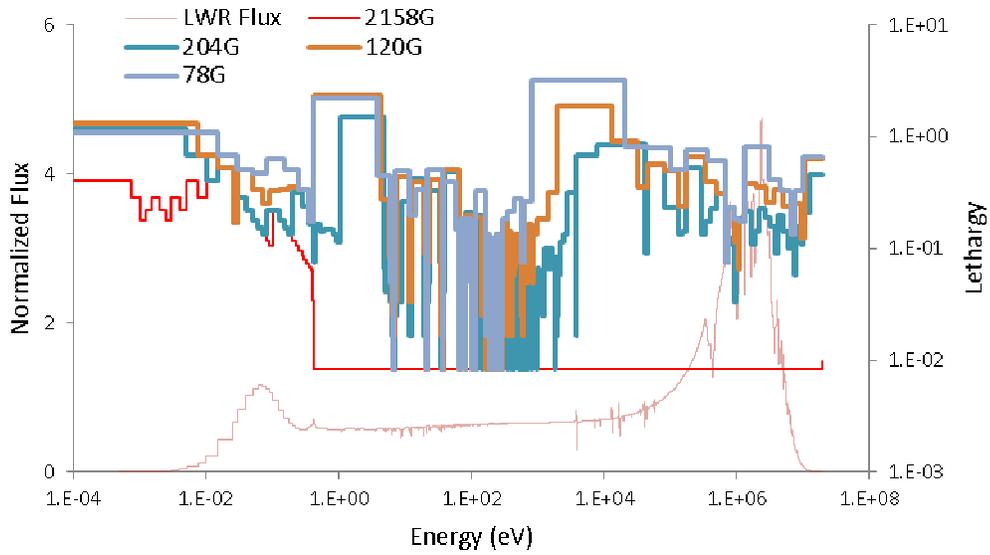


Figure 2.21 Lethargy Intervals of the Broad Group Structures for Light Water Reactor

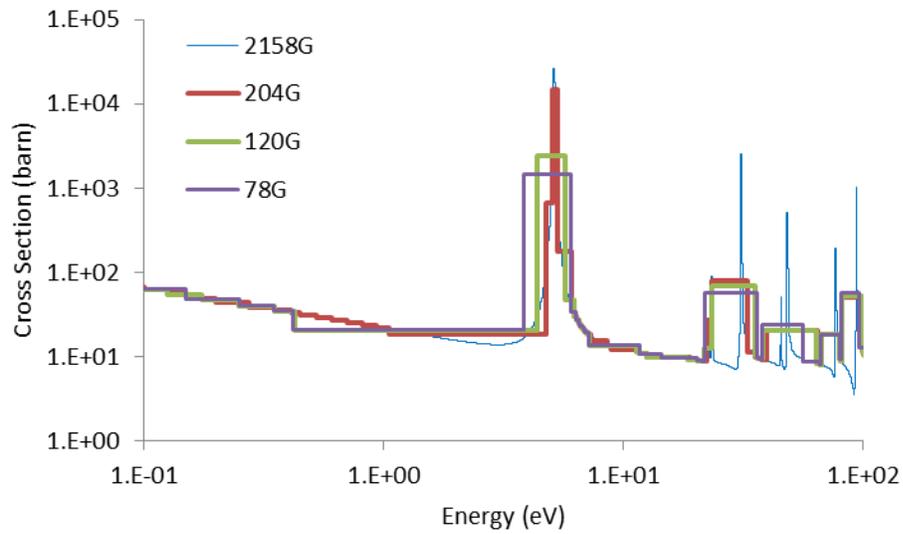


Figure 2.22 U-238 Total Cross Sections with Different Group Structures in Light Water Reactor (Case 1)

Very High Temperature Reactor

For VHTR, seven different compositions were selected as listed in Table 2.9. The composition of Case 1 (14 wt% U-235) is a typical VHTR fuel composition. The other cases were created by modifying the number densities of U-235, U-238, or other isotopes of the Case 1 composition. These changes increased U-235 enrichments up to 89 wt% for Cases 6 and 7. The carbon number density was increased to mimic the neutron spectrum for a fuel block or a whole core where a large thermal spectrum peak appears, as shown in Figure 2.23. These cases were created to provide various background cross sections to two major actinides, U-235 and U-238, in VHTR.

As shown in Figure 2.23, the neutron spectra have relatively large neutron fluxes in the intermediate energy range, compared to the LWR spectrum. Therefore, the more number of energy groups in VHTR than in LWR are expected in the epi-thermal energy range.

Table 2.9 Isotopes and Number Densities of High Temperature Reactor Compositions

Case	1		2		4		4		5		6		7	
Isotope & Number	U-235	1.06E-04												
	U-238	6.45E-04	U-238	6.45E-04	U-238	1.29E-03	U-238	3.23E-04	U-238	6.45E-05	U-238	1.29E-05	U-238	1.29E-05
Density	O-16	1.13E-03	O-16	1.13E-02	O-16	2.26E-02								
	C-12	6.06E-02	C-12	1.31E-01	C-12	6.06E-02	C-12	6.06E-02	C-12	6.06E-02	C-12	6.06E-01	C-12	1.21E+00
	Si-28	2.52E-03	Si-28	2.52E-02	Si-28	5.03E-02								

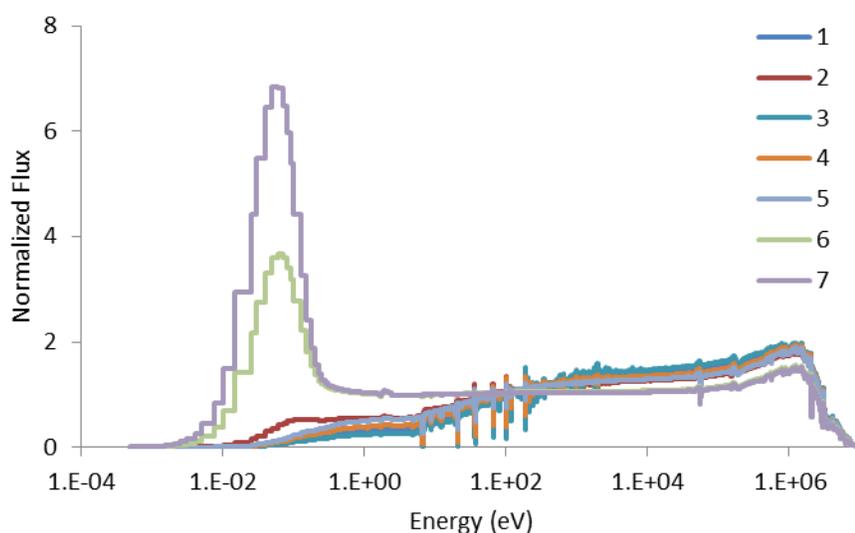


Figure 2.23 Neutron Spectra of High Temperature Reactor Compositions

The group condensation was performed using Cases 1, 2, and 5 as variable compositions and Case 6 as the representative UFG neutron spectrum composition. In the same manner as other reactor types, the criterion of  $\Delta k$  for each broad group was set to 5, 20, and 50 pcm in the group optimization process of GeneCS. The resulting broad groups were 198, 113, and 76 groups for a  $\Delta k$  criterion of 5, 20, and 50 pcm, respectively. Similar to LWR,

the resulting numbers of broad groups with the same stopping criteria for determining group boundaries are smaller than those for SFR. It is interesting to note that the number of broad groups for VHTR is similar to those for LWR, even though the allocation of group boundaries are different each other.

Reference solutions were generated using MCNP5. As shown in Table 2.10, the eigenvalue solutions from the RI table method with the 2158 group library (using the Bondarenko iteration for resonance self-shielding) agreed with MCNP5 solutions within 174 pcm. The eigenvalues from the broad group libraries with 198, 113, and 76 groups were off from the MCNP5 solutions by maximum 195, 244, and 317 pcm  $\Delta k$ , respectively. Note that all the largest errors came from Case 7 which is a kind of extreme case with very large amount of carbon. When using the existing group structures: the DeCART 190 group structure and the SCALE (ORNL) 238 group structure, Cases 1 and 7 showed reasonably small differences in eigenvalue from the MCNP5 solutions but very large errors were observed in most cases (Cases 2 through 6) for all exiting group structures. For the existing group structures, in fact, we switched the representative spectrum from Case 6 to Case 1 because Cases 1 to 4 resulted in more than 20% error in eigenvalue with the representative spectrum of Case 6. The results indicate that the existing group structures are very sensitive to the choice of the representative spectrum, whereas the broad group structures generated by GeneCS show no noticeable changes in eigenvalue no matter which representative spectrum was used. In the existing group structures, the large eigenvalue differences are caused in Cases 2 through 6 by over- and under-estimation of U-238 absorption cross sections in the energy range between 6.7 and 200 eV, similarly to the LWR cases.

Figure 2.24 compares the resulting broad group structures, from which the lethargy plots are shown from the smallest (top) to largest (bottom) number of groups. Figure 2.25 shows how the broad group structures look like in association with U-238 resonances in the energy range between 1 eV and 1 keV. Similarly to LWR, the broad group structures for VHTR over the energy range have more groups than those for SFR. However, finer groups are assigned to VHTR in the epi-thermal energy range, as expected.

Table 2.10 Eigenvalue Comparison for High Temperature Reactor Compositions

Case	MCNP5	UFG( $\Delta k$ ) 2158G	BG ( $\Delta k$ )		
			198	113	76
1	1.06282	111	57	109	168
2	1.21039	79	52	131	231
3	0.86047	102	-64	101	-28
4	1.23983	72	51	88	123
5	1.53891	-33	-106	-143	-200
6	1.84899	-99	-101	-72	-79
7	1.73383	-174	-195	-244	-317

\* Standard deviation of MCNP5 solutions  $\leq 30$  pcm

\* The UFG and BG columns show differences in eigenvalue, pcm, from MCNP5

As aforementioned, the GeneCS code produces the cross section library in a DeCART library format. Using the selected cases for LWR, VHTR, and SFR, the code generated the

BG cross section libraries as well as the UFG cross section libraries for each reactor type. We updated DeCART to use the RI table method in terms of absorption, nu-fission, and scattering resonance cross sections. The UFG or BG cross section library can be directly read by DeCART. All results produced by the GeneCS code have been reproduced in DeCART.

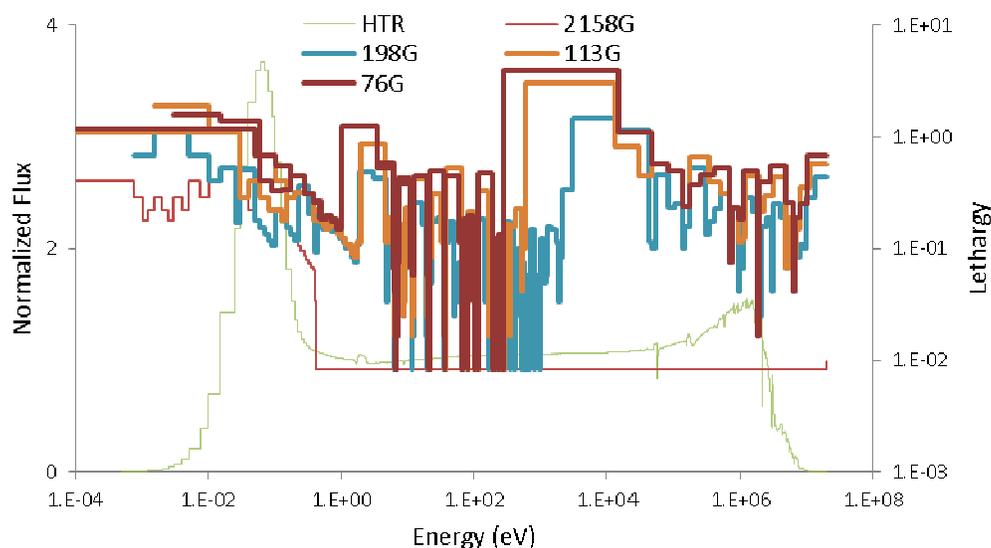


Figure 2.24 Lethargies of the Broad Group Structures for High Temperature Reactor

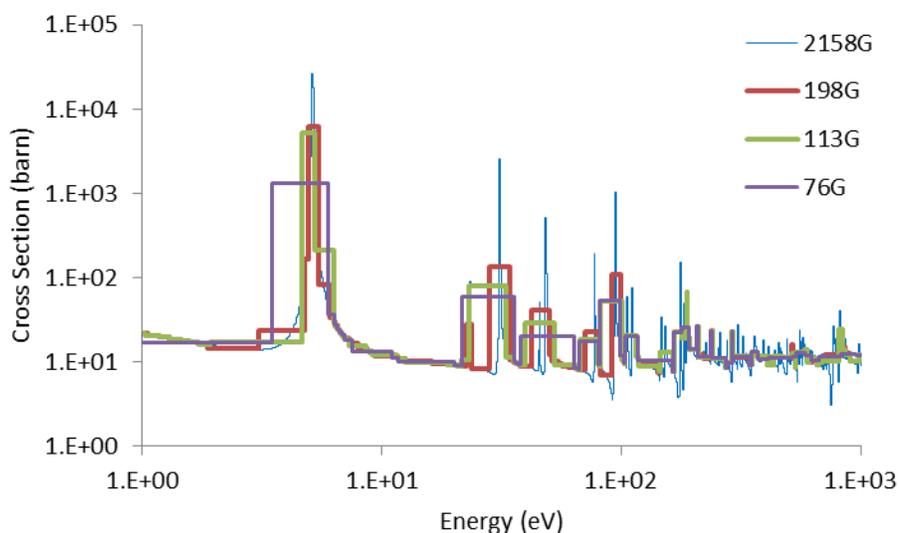


Figure 2.25 U-238 Total Cross Sections with Different Group Structures in High Temperature Reactor (Case 6)

### 3 The Cross Section API

The cross section application programming interface (API) was developed to make it easier to implement the cross section generation tool into any transport code. First, the subgroup method was selected as a cross section generation method. The subgroup API is divided into two parts: the subgroup-independent and -dependent ones. The subgroup-independent part is composed of setting-up, defining, reading & storing, and cleaning-up data, while the subgroup-dependent part contains the routines calling the fixed-source solver, handling the subgroup parameters, and conducting the Bondarenko iteration. When a different cross section method is used, only the method-dependent part needs to be updated.

The subgroup API reads isotopic data from the cross section library and the cross section region mapping information as well. Once all subgroup cross section data are specified and read from and written to necessary data files, the subgroup API generates the region-wise escape cross sections using the one-group (i.e., group-independent) whole-core fixed-source solver provided by the transport code. Therefore, one may expect to change a transport code to separate out a fixed-source transport solver in the manner as directed by the API. This would not be difficult since most transport codes should have an independent subroutine or module to solve for  $AX = B$ . To calculate the escape cross sections, the API sets up relevant cross sections and sources for the fixed-source solver. A do-loop is necessary to cover all energy groups and an iteration algorithm may be needed to converge cross sections and fixed sources depending upon the cross section method of choice. Figure 3.1 shows a developmental scheme on how the transport solver and the subgroup API interact.

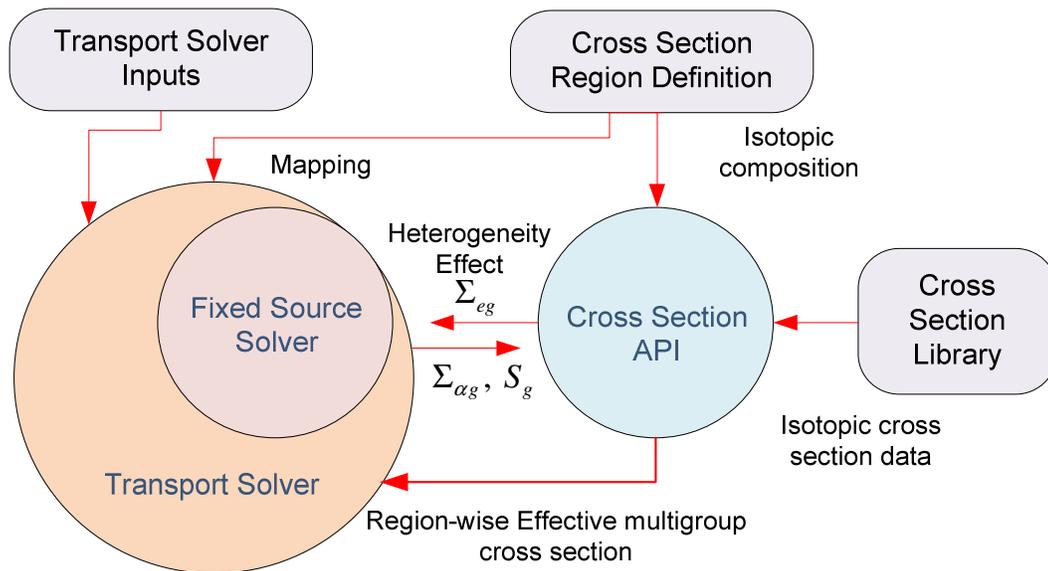


Figure 3.1 Interaction between the Cross Section API and the Neutron Transport Solver

### 3.1 Structure

The subgroup API was developed using the Fortran 2003 programming language. The API containing the C-language interfaces will be programmed to handle parallelism by using the MPI interface. Before programming, a design document was written in order to define the API and the interface functions. This design document defines three phases for the API: data setup, computation of the effective multi-group cross sections, and data cleanup. This document was internally reviewed at Argonne and peer-reviewed by the ORNL neutronics expert group with whom we have collaborated for the API development.

In the subgroup API design document, over 40 interface functions are defined and the interaction between the transport solver and the subgroup API is specified (see Appendix C for major subroutines and functions). Those functions can be gathered into three distinct types: 1) setup and clean up, 2) cross-section retrieval, and 3) statistics and error handling. The internal objects defined by the API are opaque to the transport solver and all the interaction between the API and the transport solver is made through basic interface arguments. No complex structures are used as arguments of the interface procedure.

A new library storage format was defined based on the HDF5 API. The HDF5 API enables the creation of binary files that can be transferred on different system and can be read efficiently in parallel. A specific structure is designed to store the cross-section library data in memory. Several steps are required between the subgroup API and the transport solver to generate the effective multigroup cross sections. Details will be discussed in this section. The header part of the major subroutines and functions for the subgroup API is listed in Appendix A, as well.

#### 3.1.1 Initialization

The initialization of the subgroup API is first performed to provide a path to a valid subgroup cross section library file. By doing this, the transport solver has access to the basic level data for the cross section library. The basic level data information includes the number of energy groups, the number of isotopes, their names, their atomic masses, and so on.

If parallelization with respect to the energy variable is requested, one should also provide a valid MPI communicator and the energy group range to the local process. Note that the subgroup method only performs a one-group calculation and therefore parallelization is almost trivial. However, a load imbalance may occur since the computation load for resonance energy groups is considerably higher than for non-resonance energy groups.

It is possible to select a cross section method to be used. The choice will be dictated by two parameters: the current status of the cross section API implementation and the cross-section library format. Different methods may use different data types (resonance integrals vs. subgroup parameters) or different tabulation parameters (temperature, levels, etc.). Thus, the cross section API should have data types in accordance with the method of choice.

The subgroup API was developed based on the assumption that the subgroup method relies on solving a one-group fixed source problem. The subgroup API only needs a function pointer to call the one-group fixed source solver when needed. Figure 3.2 shows the function interface expected by the subgroup API. The API sends the total and scattering cross sections

and sources and receives the volume-weighted scalar fluxes for all calculation meshes or cross section regions.

```

! Defines the interface for the FSS procedure pointer
! This function is not meant to be called

SUBROUTINE FSS_NoArg &
    (NumberOfRegions, TotalXS, ScatteringXS, Source, Flux, Error, IERR)
    IMPLICIT NONE

    INTEGER (SG_Int), INTENT(IN)           :: NumberOfRegions
    REAL (SG_Double), DIMENSION(:), INTENT(IN) :: TotalXS
    REAL (SG_Double), DIMENSION(:), INTENT(IN) :: ScatteringXS
    REAL (SG_Double), DIMENSION(:), INTENT(IN) :: Source
    REAL (SG_Double), DIMENSION(:), INTENT(OUT) :: Flux
    REAL (SG_Double), INTENT(IN)           :: Error
    INTEGER, INTENT(OUT)                    :: IERR
END SUBROUTINE FSS_NoArg
    
```

Figure 3.2 Subroutine Call from the Subgroup API to the Transport Solver

### 3.1.2 Cross Section Region Setup

There are three different meshes or regions in the cross section API: calculation mesh, cross section mesh, and composition mesh. The calculation mesh is a finite element mesh which is used for the transport calculation, the composition region, rather than mesh, is determined by material assignment and depletion, and the cross section mesh can be determined between the calculation mesh and the composition region. A group of contiguous elements can be defined as a cross section mesh. Even though the elements belong to the same composition region, they will be defined as different cross section meshes if they are not neighboring each other.

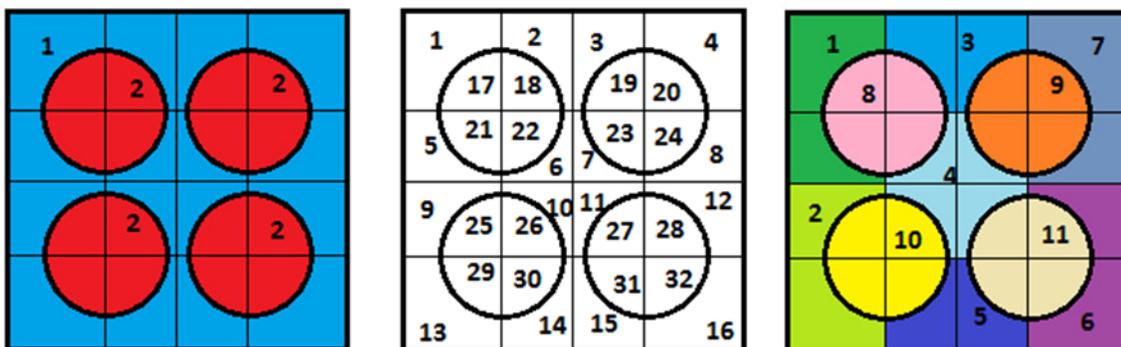


Figure 3.3 Composition Assignment (left), Element Indexing (middle), and Cross Section Region Mapping (right) for a 2x2 Pin-Cell Geometry

Figure 3.3 illustrates different mapping layers for a simple 2×2 pin-cell geometry. The first one shows the composition assignment with two different compositions and the second one illustrates 32 calculation meshes for which 8 calculation meshes are created for each pin. The last one shows an example of the cross section meshes for which 11 different regions are defined. The transport code is responsible for specifying those three different mesh types and constructing the mapping indices between those mesh types. The information should be passed from the transport code to the cross section API.

### 3.1.3 Cross Section Calculation

All the data concerning the cross-section regions should be provided to the subgroup API by the transport solver. The subgroup API needs to know the number of cross-section regions, their isotopic compositions and actual temperature before computing any cross section-related parameters. For each composition, the transport solver provides the number of isotopes, isotope names, and atom densities. The subgroup API needs to store this data separately. This may be a waste of computer resources because of duplicated memories for the transport code and the cross section API. However, this duplication is inevitable for the subgroup API to have all the data that it needs. If the available memory size is insufficient, the transport code can free up the memory occupied by the subgroup API once the cross sections are determined.

The transport code calls the subgroup API by passing the temperature information. Then, the subgroup API sets up all memories necessary for the cross section calculation and then determines smooth cross sections in terms of region and energy by reading data from the cross section library. The resonance-related data are provided directly from the library.

The API calls the fixed source solver to calculate the escape cross sections for a whole core. In the subgroup method, the fixed-source calculation produces the scalar fluxes to determine the escape cross sections for uniform temperatures, which are dependent on the compositions only. Therefore, this fixed-source calculation will be performed only when compositions are changed, for example, by control rod movement or depletion. Figure 3.4 illustrates the computational flow of the subgroup API in association with the transport code.

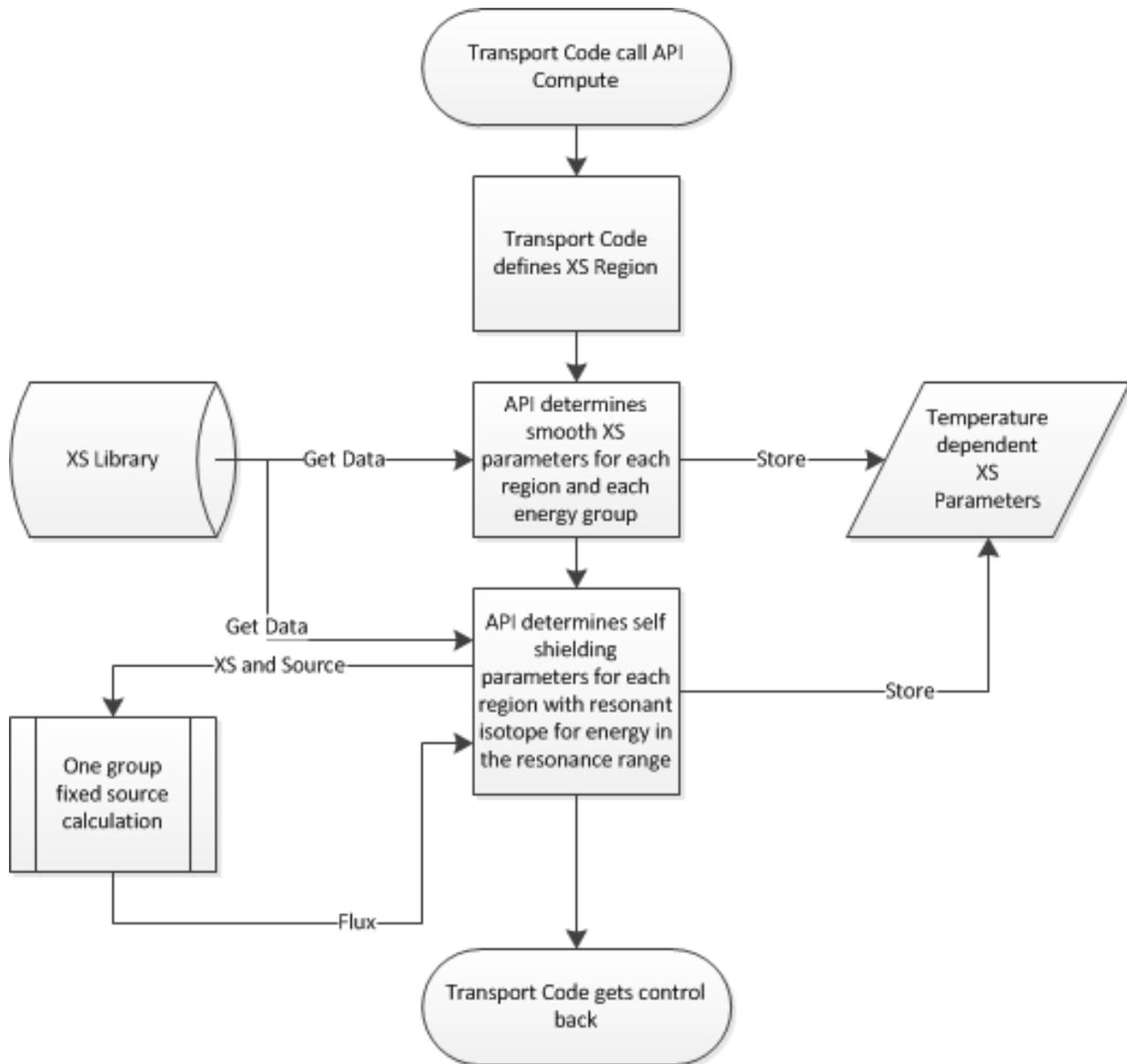


Figure 3.4 Flow Chart of the Computation Phase

### 3.1.4 Multigroup Cross Section Retrieval

The cross section API provides a capability of retrieving macroscopic cross-sections for regions and energy groups. Figure 3.5 shows the interface function used for this purpose. The API retrieves macroscopic cross section data for one cross section region at a time. Each function call returns absorption, chi times nu-fission, and power conversion factors, and scattering matrix for the energy group range specified. For example, it is possible to get cross section data for one group at a time by specifying only one group range. For parallelization in energy, retrieving the local group information only is possible.

```

!! @param NG Number of global energy groups
!! @param GStart Index of first energy group
!! @param GEnd Index of last energy group
!! @param LO Maximum legendre order to return for the scattering matrix
!! @param AbsorptionXS Macroscopic absorption cross-section in the region
!! @param ChiNuFissionXS Macroscopic ChiNuFission matrix in the region
!! @param ScatteringXS Macroscopic scattering matrix in the region
!! @param PowerConversionXS Macroscopic power conversion cross-section in the
!!      region
!! @param IERR Returned error code

SUBROUTINE SG_GetRegionMacroXS(RI,NG,GStart,GEnd,LO,AbsorptionXS, &
      ChiNuFissionXS,ScatteringXS,PowerConversionXS,IERR)
  IMPLICIT NONE

  INTEGER (SG_Int), INTENT(IN) :: RI
  INTEGER (SG_Int), INTENT(IN) :: NG
  INTEGER (SG_Int), INTENT(IN) :: GStart
  INTEGER (SG_Int), INTENT(IN) :: GEnd
  INTEGER (SG_Int), INTENT(IN) :: LO
  REAL (SG_Double), DIMENSION(:), INTENT(OUT) :: AbsorptionXS
  REAL (SG_Double), DIMENSION(:,,:), INTENT(OUT) :: ChiNuFissionXS
  REAL (SG_Double), DIMENSION(:, :, :), INTENT(OUT) :: ScatteringXS
  REAL (SG_Double), DIMENSION(:), INTENT(OUT) :: PowerConversionXS
  INTEGER, INTENT(OUT) :: IERR

END SUBROUTINE SG_GetRegionMacroXS

```

Figure 3.5 Subgroup Interface Used to Retrieve Macroscopic Cross Section Data

### 3.1.5 Utilities

The cross section API was documented using the Doxygen program [27]. This is an efficient way to provide the source code documentation at the same time as the code is being developed. Doxygen enables the production of documentation in the Html format and the standard print (pdf or Latex format). Doxygen parses the sources code and extracts special comments, subroutine interfaces and structure definitions to generate a complete documentation of the source code. A design document discussed in the previous section was

written and reviewed internally before programming. Figure 3.6 shows a web view produced by Doxygen.

The Fortran Unit Test Framework (FRUIT) [28] was installed. The Ruby wrapper was removed to keep only the main assertion subroutine definition. Unit tests are written for each module developed in the API. The role of unit tests is important to in advance identify failing components of a piece of the API. The unit tests are completed and well documented

A “Makefile” was created in order to facilitate the compilation of the subgroup API. The main target compiles all source code needed by the subgroup API and aggregates it into a static library file *libsubgroup.a*. Other useful target objects are “check” (compiles and run the unit tests), “doc” (generate the documentation via Doxygen), “clean” (remove all object and mod files from the compile directory), “clean\_check” (remove the output from the unit tests), and “tags” (generate a TAG table for the Emacs text editor)

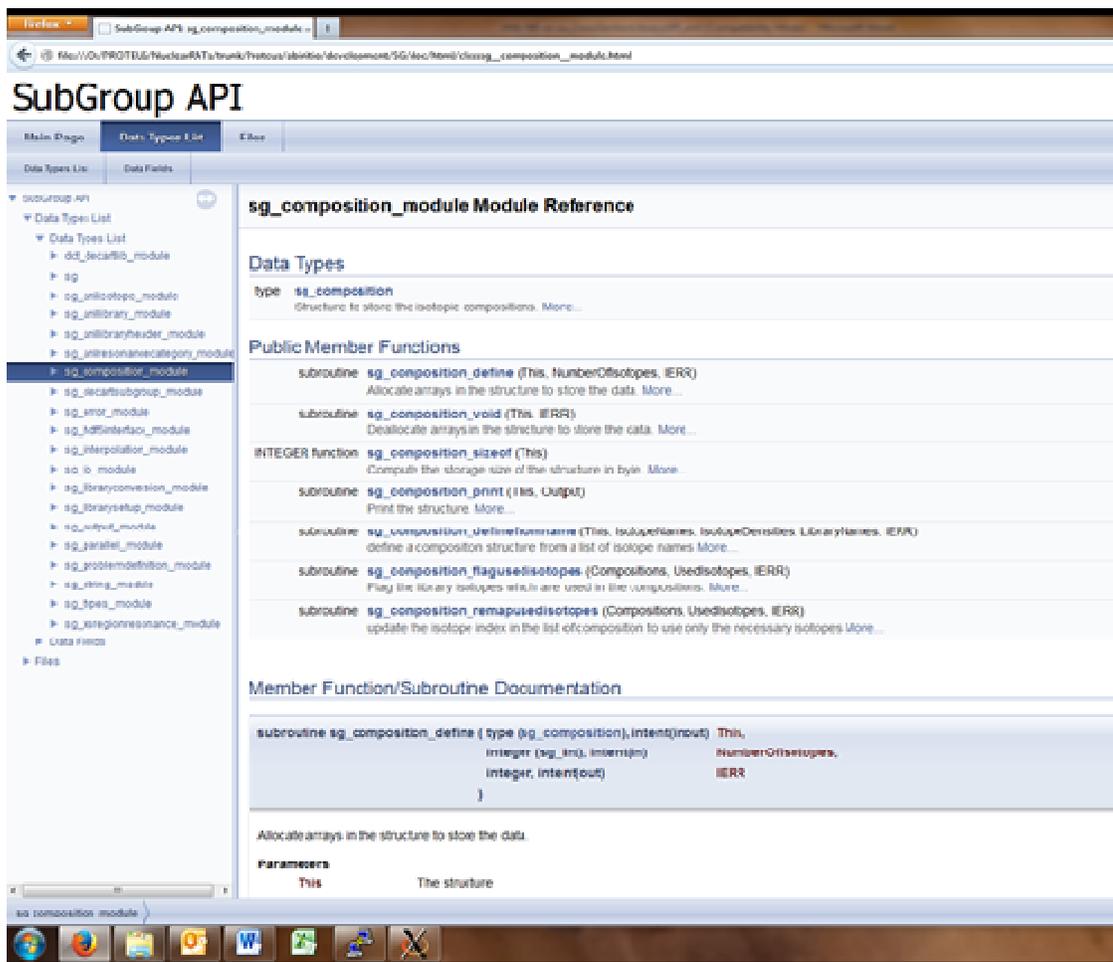


Figure 3.6 Doxygen HTML Output for Composition Module

### 3.2 Implementation of the Cross Section API into PROTEUS

The PROTEUS code is a continuous finite element mesh based high-fidelity neutron transport code developed at Argonne, containing two different neutronics solvers:  $S_N$  (a discrete ordinates solver based on the even-parity formulation of the transport equation) and MOC (a 2D MOC, a 3D MOC, and a 2D MOC combined with 1D FEM). The development of PROTEUS-SN is an ongoing research effort and has been closely related to the availability of a supercomputing machine. Given the cutting edge nature of its development, it has yet to prove its ability to routinely perform the desired heterogeneous full core calculation because of its requiring significant computational resources. The PROTEUS-MOC solver is intended to resolve the outstanding computational challenges by which the PROTEUS-SN is limited and to bridge the gap between PROTEUS-SN and existing homogenized structured grid solvers.

Since PROTEUS-SN requires substantial efforts to specify cross section regions and set up mapping between calculation meshes, composition regions, and cross section regions, we decided to first test the subgroup API on PROTEUS-MOC and then move on to PROTEUS-SN in the next fiscal year.

Parallelization with respect to radial mesh was recently implemented into PROTEUS-MOC. The PROTEUS-MOC code can now handle parallelization with respect to angle, energy, and radial and axial spaces, although parallelization in energy has not been thoroughly tested yet. The number of trajectories in a given element can vary substantially from direction to direction and thus a mesh partition balanced for one direction may not be balanced for another direction. Detailed scalability tests need to be performed. PROTEUS-MOC was updated to generate a HDF5-format output so that one can easily check the output using the visualization tool VisIt [29]. A one-group fixed-source calculation kernel was added to PROTEUS-MOC in order to support the development of the cross section API, in which the fixed sources are assigned to every mesh and resulting scalar fluxes are averaged for user-defined cross section regions.

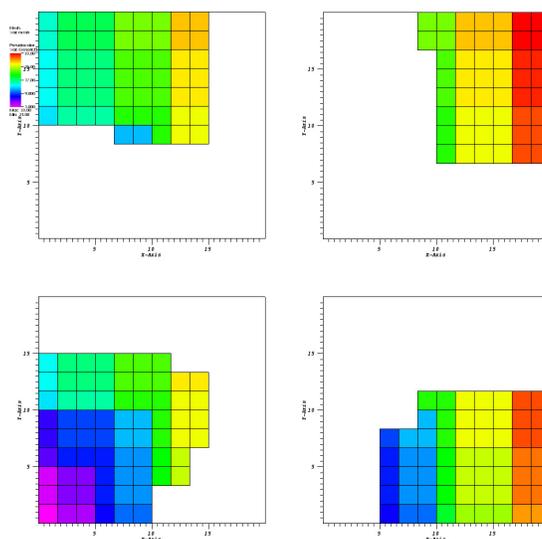


Figure 3.7 Cross Section Region Mapping to 4 Processors

The cross section mesh can in principle be the same as the calculation mesh, which will require huge memory and storage to store cross section data. To reduce the required resources, the cross section meshes are normally specified coarser than the calculation mesh by grouping contiguous mesh elements within compositions, called an averaged cross section region (ACR). The smallest ACR would be one calculation mesh. The number of elements within an ACR is determined by a user-input which specifies the target volume (or an area in 2D) of an ACR. The ACR allocation does not depend on the spatial decomposition. In Figure 3.7, one can see the cross section region mapping of each cell based on given ACRs. In this figure, the spatial meshes with a unit element size of  $2.78 \text{ cm}^2$  have been decomposed into 4 processors and at the same time the unit ACR size was set to  $20 \text{ cm}^2$ . The cross sections are assigned to the overlapped areas between 4 processors, based on the ACR allocations.

### 3.3 Verification Tests

The cross section API has been tested and verified for homogeneous, pin-cell, and  $2 \times 2$  lattice problems. Among the subgroup method and resonance integral (RI) table method implemented into the API, the former has first been tested for various problems in terms of geometry and compositions.

Test problems were created based on the existing LWR or VHTR configurations and the VERA (Virtual Environment for Reactor Analysis) PWR core physics benchmark problems [30]. Of the 21 benchmark cases with 1D pin cell and 2D lattice geometries, 9 cases were selected for our tests, which are composed of two 1D cases and seven 2D cases, as shown in Table 3.1. Specification of those cases are based on standard Westinghouse-type PWR components. One of the 1D pin cell cases contains IFBA coating which is difficult to simulate in terms of resonance self-shielding. The 2D cases are all  $17 \times 17$  assemblies with 3.1 wt% enriched fuels and 24 guide tubes and 1 instrument thimble. Variations are made with different types of burnable poisons (BAs) such as WABA, Pyrex, gadolinia, and IFBA or AgInCd control rods.

The DeCART 47-group library was used for initial tests of the subgroup API. The inputs of DeCART were prepared for the nine cases based on the specifications described in the VERA benchmark document. First of all, a convergence study was conducted to determine the ray tracing parameters such as ray spacing and the number of azimuthal and polar angles. We found that 0.015 cm ray tracing and 40 azimuthal angles and 2 polar angles per 90 degree, i.e. (0.015, 40, 2), are sufficient to obtain the converged solutions for most cases, but considering the computational efficiency we used a coarser refinement (0.030, 38, 2) for the 2D cases and a finer refinement (0.001, 20, 2) for the fuel pin with IFBA (Case 2E) which requires very fine ray spacing. Although we did not use the fine ray spacing this time, the 2D cases with IFBA (Cases 2M and 2N) need a fine refinement of ray spacing based on the observation in the pin cell with IFBA.

Table 3.1 Selected VERA PWR Core Physics Benchmark Problems\*

Case	Description
1A	3.1% unit fuel pin
1E	3.1% unit fuel pin with IFBA
2A	17×17 fuel assembly with 3.1 wt% fuel pins
2F	17×17 fuel assembly with 3.1 wt% fuel pins + 24 Pyrex BA
2G	17×17 fuel assembly with 3.1 wt% fuel pins + 24 AgInCd control rods
2K	17×17 fuel assembly with 3.1 wt% and 3.6 wt% fuel pins + 24 Pyrex BA
2M	17×17 fuel assembly with 3.1 wt% fuel pins and 128 IFBA pins
2N	17×17 fuel assembly with 3.1 wt% fuel pins and 104 IFBA pins + 20 WABA
2P	17×17 fuel assembly with 3.1 wt% fuel pins and 24 Gd pins

\* The temperatures were modified to 300K from the original ones.

Table 3.2 Eigenvalue Comparison of the Selected VERA Benchmark Problems

Case	Cont. Energy	252G	56G		47G	
	KENO-6	ESSM	ESSM	Subgroup	DeCART	
1A	1.18761	-74	49	26	276	0.015,40,2
1E	0.77237	31	59	55	335	0.001,20,2
2A	1.18251	-120	-15	-79	244	0.030,38,2 <sup>a)</sup>
2F	0.97690	-43	26	23	130	
2G	0.84924	-225	-75	188	-142	
2K	1.02100	-46	31	36	103	
2M	0.93946	137	193	140	11	
2N	0.87043	79	144	99	29	
2P	0.92800	-5	56	25	266	

a) (a,b,c) a: ray spacing in cm, b: number of azimuthal angles per 90°, c: number of polar angles per 90°

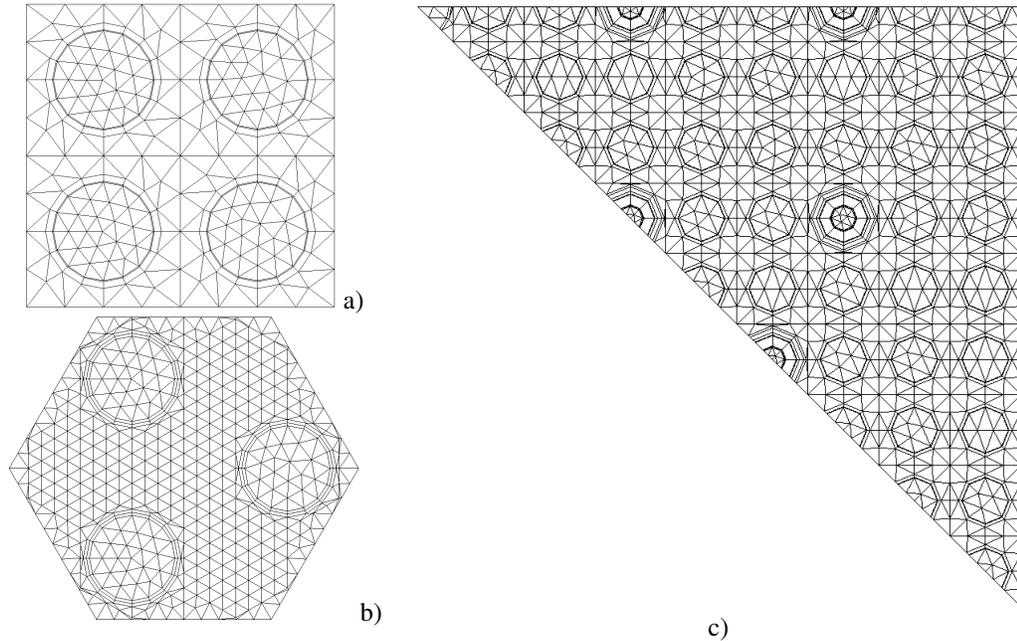
\* The multigroup columns show differences in eigenvalue, pcm, from KENO-6

Table 3.2 shows the eigenvalue comparison between different methods and libraries for the nine VERA benchmark problems. The results from KENO-6 [16], ESSM (the Embedded Self-Shielding Method), and Subgroup (the conventional subgroup method) were obtained from ORNL [31], and the last column results were produced by DeCART with the 47-group subgroup library. The 252-group ESSM results appear to be well optimized for these benchmark problems, showing a maximum difference of 225 pcm from KENO-6 for the assembly case with 24 control rods. The differences in the 252-group ESSM were rather reduced in the 56-group ESSM, which should be caused by error cancellation.

The subgroup API was successfully connected to PROTEUS-MOC as discussed in the previous section. The subgroup API is designed to read data from the cross section library, call a fixed source solver provided by a transport code, and provide multigroup cross sections for all cross section regions. It is divided into the resonance-dependent and -independent parts. Thus, the resonance-dependent part can easily be replaced by any other resonance self-shielding method as needed.

Several benchmark problems were created to verify the API routines, including homogeneous mixture, pin-cell, and multi-pin problems with rectangular and hexagonal

geometries. Figure 3.8 (a) and (b) illustrate meshes of the multi-pin benchmark problems. In addition, 9 benchmark problems were selected from the VERA benchmark problem set developed by ORNL [30], listed in Table 3.1, which include 2 pin-cell problems and 7 lattice problems. The meshes for those VERA benchmark problems were prepared as shown in Figure 3.8 (c).



(a) 2x2 Pin Cell Benchmark, (b) 7 Hexagonal Pin Cell Benchmark, (c) VERA 2F

Figure 3.8 Meshes of Benchmark Problems for the Cross Section API

### 3.3.1 Pin Cell Problems

#### PWR-VERA-1A

The first case is a single 2D PWR fuel pin cell which is composed of fuel, gap, cladding, and moderator in turn. The reflective boundary condition was applied at the pin cell boundaries. The fuel pin has 3.1 wt%  $\text{UO}_2$  fuel with a 0.4096 cm radius. It is surrounded by a 0.57 mm thick cladding and there is a 0.084 mm thick oxygen gap between fuel and cladding. The last region of the pin cell is filled with water and soluble boron. The cell pitch is 1.26 cm. The isotopic compositions are listed in Table 3.3.

A parametric study was performed with the VERA 1A pin-cell benchmark problem using the existing 47-group library of DeCART. As listed in Table 3.4, the eigenvalue varied largely with change of the number of angles. The mesh refinement also has a big impact on eigenvalue. A difference of 499 pcm in eigenvalue was made between the coarse (44 elements) and fine meshes (328 elements). The change of the trajectory area (ray spacing) had no effect on eigenvalue. This is because an algorithm in the code sets up trajectories to be at

least one trajectory per mesh element. The effect of refining the cross section region area was quite small, only 40pcm between 1.6 and 0.02 cm<sup>2</sup>.

The final eigenvalue ( $k_{\infty}$ ) with the most refined configuration is 1.19024 which is 263 pcm off from the KENO Monte Carlo solution ( $k_{\infty} = 1.18761$ ) and 13 pcm off from the DeCART solution ( $k_{\infty} = 1.19037$ ) with the same 47-group library. This good agreement in eigenvalue with DeCART indicates that the implementation of the subgroup API into PROTEUS-MOC was done correctly.

Table 3.3 Composition of PWR-VERA-1A Pin Cell Problem

Material	3.1 wt% UO <sub>2</sub> Fuel		Gap		Cladding		Moderator		
Isotope	U-234	6.119E-06	O-16	2.687E-05	Zr	4.254E-02	O-16	2.481E-02	
	U-235	7.181E-04			Sn	4.779E-04		H-1	4.962E-02
	U-236	3.299E-06			Fe	1.486E-04		B	5.380E-05
	U-238	2.215E-02			Cr	7.598E-05			
	O-16	4.576E-02			Hf	2.213E-06			

Table 3.4 Eigenvalues with Change of Parameters for a Pin Cell Problem

Angle a)	$k_{\infty}$	Element b)	$k_{\infty}$	Ray <sup>c)</sup> Spacing	$k_{\infty}$	XS <sup>d)</sup> Area	$k_{\infty}$
32	1.18306	44	1.18562	1	1.19061	1.6	1.19064
72	1.18919	84	1.19122	0.5	1.19061	0.5	1.19059
128	1.18749	152	1.19135	0.1	1.19061	0.1	1.19061
200	1.19061	328	1.19061	0.05	1.19061	0.02	1.19024

a) 0.1 cm ray spacing, 0.1 cm<sup>2</sup> cross section area, and 328 elements

b) 0.1 cm ray spacing, 0.1 cm<sup>2</sup> cross section area, and 200 angles

c) 0.1 cm<sup>2</sup> cross section area, 328 elements, and 200 angles

d) 0.1 cm ray spacing, 328 elements, and 200 angles

### Modified PWR-VERA-1E

The second case is a single 2D PWR fuel pin cell with IFBA coating to the fuel pellet. Since the PROTEUS-MOC with the subgroup API has shown a problem with a thin IFBA coating and may take time to investigate the problem, we decided to modify the fuel configuration slightly by smearing the IFBA into the fuel pellet. Since no IFBA coating is dealt with in this case, the geometry configuration is the same as the previous pin cell case 1A. For this test, the 47-group DeCART library was used as well.

As shown in Table 3.5, the results from the parametric study in terms of element, cubature, and angles showed trends similar to those seen in the previous case 1A. The change of the number of elements and angles with a Legendre-Tchebyshev order had a large influence on eigenvalue. The eigenvalue from DeCART for this case is 0.77426 while that obtained from PROTEUS-MOC with the subgroup API is 0.77331 which is different by only

95 pcm. Note that both codes used the DeCART 47-group library. This indicates that the implementation of the subgroup API and its integration into PROTEUS-MOC was successful.

Table 3.5 Composition of PWR-VERA-1E Pin Cell Problem

Material	3.1 wt% UO <sub>2</sub> Fuel		Gap		Cladding		Moderator	
Isotope	U-234	6.119E-06	O-16	2.687E-05	Zr	4.254E-02	O-16	2.481E-02
	U-235	7.181E-04			Sn	4.779E-04	H-1	4.962E-02
	U-236	3.299E-06			Fe	1.486E-04	B	5.380E-05
	U-238	2.215E-02			Cr	7.598E-05		
	O-16	4.576E-02			Hf	2.213E-06		
	B	2.020E-04						
	Zr	1.010E-04						

Table 3.6 Eigenvalues with Change of Parameters for a Pin Cell Problem

Element <sup>a)</sup>	k <sub>∞</sub>	Angle <sup>b)</sup>	k <sub>∞</sub>
200	0.77075	48	0.76790
728	0.76973	96	0.77075
1472	0.76999	192	0.77169
		384	0.77331

- a) 96 angles used
- b) 200 elements used

### 3.3.2 Lattice Problems

Attempts were made to simulate the 17×17 lattice cases. However, these attempts were unsuccessful on our Linux cluster “Janus” (Intel Xeon) due to memory constraints and long computation time. To resolve the problem, PROTEUS-MOC should be run on a supercomputer such as the BG/P machine available at the Argonne Leadership Computing Facility (ALCF). Instead of the 17×17 cases, therefore, we decided to develop 2×2 pin cell problems which can give more clear comparisons on verification tests.

Two 2×2 pin cell problems were constructed using the pin cells 1A and 1E. One case (Type A) is a checkerboard type and another (Type B) is composed of three 1A pin cells and one 1E pin cell. Flux distributions for group 24 (~3 eV) from Types A and B are shown in Figure 3.9.

Eigenvalue solutions from DeCART for the two cases are 1.05544 and 0.94136, respectively. The eigenvalues obtained from PROTEUS-MOC with the subgroup API are 257 pcm and 289 pcm off from those DeCART solutions when the code used 800 elements and 384 angles. The differences are much larger than those observed from the pin cell comparisons. There are two possible reasons for the large difference: 1) the PROTEUS-MOC solution was not completely converged since we still observed about 100 pcm change between the last two refinement steps, as shown in Table 3.7. 2) These 2×2 pin cell problems

are difficult to solve because of a large gradient between two distinctly different pins ( $k_{\infty} = 1.19037$  vs.  $0.77296$ ) and therefore the DeCART solutions should be checked as well. However, DeCART has limitations in the spatial discretization. Further investigation should be continued as future work.

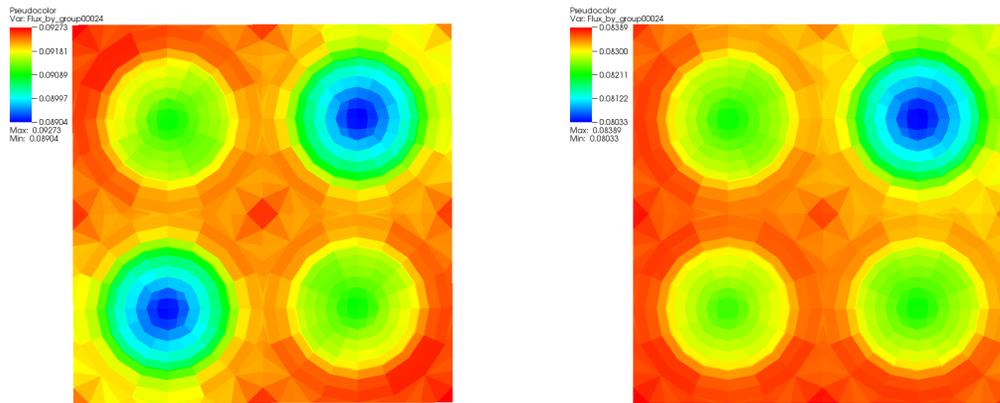


Figure 3.9 Flux Distributions (at ~3 eV) for 2x2 Pin Cell Benchmark Problems

Table 3.7 Eigenvalues with Change of Parameters for 2x2 Pin Cell Benchmark Problems

Angle <sup>a)</sup>	Type A	Type B
48	1.04402	0.93187
96	1.04884	0.93575
192	1.05104	0.93773
384	1.05216	0.93879

a) 800 elements used

## 4 Conclusions

A generalized cross section library was developed for application to various reactor types including LWR, VHTR, and SFR. The ultrafine group (2158 groups) cross section library including the resonance integral tables for resonance cross sections was produced by the GeneCS code using cross section data generated from MC<sup>2</sup>-3 and NJOY. The resonance integral tables are formulated for absorption, nu-fission, and scattering cross sections. The escape cross sections for considering the heterogeneity effect are estimated from the fixed-source transport calculation in the iterative manner with updating the resonance cross sections. The ultrafine group cross section library is condensed to the broad group library using the optimized group condensation algorithm with the representative neutron spectrum and various homogeneous or pin cell compositions of the reactor type of interest. Note that this is a group condensation process from library to library such that the resulting broad group library can be directly used for any transport code. The number of the broad groups for the reduced library is determined by a group condensation error criterion.

Verification tests of the new cross section libraries were performed mostly using DeCART since the cross section API was being implemented into PROTEUS at the same time. The new cross section library was generated in the DeCART library format which does not include all necessary data needed by a transport code. A more general library format is needed in the future. We note that producing the library in the DeCART format was useful to support the I-NERI collaboration between Argonne and KAERI for verification and validation of DeCART and high-fidelity multi-physics simulation for advanced nuclear reactors. Verification results of the new libraries indicated that the eigenvalues were estimated within 200-300 pcm for all LWR, VHTR, and SFR compositions depending on the resulting optimized group structures with 76 to 383 groups, compared to the Monte Carlo solutions.

Since the primary focus of NEAMS neutronics is a sodium cooled fast reactor, we developed the direct resonance self-shielding method as a high-fidelity cross section generation approach for SFRs, in which the MC<sup>2</sup>-3 methodology was extended from 1D to a 3D large scale basis. To achieve the best accuracy, the direct resonance self-shielding methodology over the whole core should be used. However, the generalized cross section library that we developed this year should satisfy most needs with reasonable accuracy for fast reactor analysis.

The cross section application programming interface (API) was developed to make it easy to plug the developed cross section module or package into an existing neutron transport code. The transport code must provide a one-group fixed-source transport solver, the isotopic breakdown of compositions, and their mapping to the physical domain. In the API, the input and output arguments required are clearly defined so that a user can understand what the API needs from and provides to the transport code. The API was developed first with the subgroup method and was integrated to PROTEUS-MOC. Twenty-one interface subroutines in the neutron transport solver were created to setup the API and retrieve the effective multigroup cross-section library.

As future work, detailed verification and validation tests for the generalized cross section library should be performed for numerous heterogeneous cases including whole-core problems with various reactor types. The cross section library should be completed to include

all nuclides and reactions by updating the library data format. The cross section API should be implemented into PROTEUS-SN so that larger benchmark problems such as the ATR can be simulated.

## References

1. M. A. Smith, A. Mohamed, A. Marin-Lafleche, E. E. Lewis, K. Derstine, C. H. Lee, A. Wollaber and W. S. Yang. "FY2010 Status Report on Advanced Neutronics Modeling and Validation," ANL-GenIV-149, September 2010.
2. M. A. Smith, A. Mohamed, C. H. Lee, A. Marin-Lafleche, and W. S. Yang, "Zero Power Fast Reactor Deterministic Modeling with Heterogeneous Geometry," American Nuclear Society, Hollywood, FL, June 26-30, 2011.
3. A. Mohamed, W. S. Yang, M. A. Smith, and C. H. Lee, "Spectrum Interaction Effects on Calculating Foil Reaction Rates in ZPR Assemblies," M&C 2011, Rio de Janeiro, Brazil, in May 8-12, 2011.
4. E. R. Wolters and M. A. Smith, "Scoping Analysis of the Advanced Test Reactor Using SN2ND", ANL-12-31, June, 2012.
5. M. A. Smith, C. H. Lee, and G. Yesilyurt, "FY2012 Status Report of Subgroup Library Development," ANL/NE-12-45, September 2012.
6. HELIOS Methods, Studsvik Scandpower, November 20, 2003.
7. G. Marleau, A. Hébert, R. Roy, A User's Guide for DRAGON, Ecole Polytechnique de Montréal, December 1997.
8. J. R. Askew, F. J. Fayers, and P. B. Kemshell, "A General Description of the Code WIMS," *J. British Nucl. Energy Soc.*, Vol. **5**, No. 4, p. 564, October 1966.
9. M. Ouisloumen, et al., "The New Lattice Code Paragon and its Qualification for PWR Core", M&C 2003, Gatlinburg, TN, April 6-10, 2003.
10. R. Sanches and J. Mondot, "APOLLO-2: A User-friendly Code for Multigroup Transport Calculations," Top. Meeting on Advances in Nuclear Engineering, Computation and Radiation Shielding, Santa Fe, 1989.
11. J. Y. Cho, et al., "Whole Core Transport Calculation Employing Hexagonal Modular Ray Tracing and CMFD Formulation," *J. Nucl. Sci. Tech.*, **45**, 745-751, 2008.
12. G. Rimpault, et al., "The ERANOS Code and Data System for Fast Reactor Neutronic Analyses," PHYSOR 2002, Seoul, Korea, October 2002.
13. R. E. MacFarlane, "NJOY 99/2001: New Capabilities in Data Processing," Presentation at the Workshop on Reactor Physics and Analysis Capabilities for Generation IV Nuclear Energy Systems, Argonne National Laboratory, Argonne, Illinois, February 18-19, 2003.
14. X-5 Monte Carlo Team, "MCNP - A General Monte Carlo N-Particle Transport Code, Version 5," LA-UR-03-1987, Los Alamos National Laboratory, 2003.
15. M. L. Williams and K. S. Kim, "The Embedded Self-Shielding Method," PHYSOR 2012, Knoxville, Tennessee, 2012.
16. SCALE: A Comprehensive Modeling and Simulation Suite for Nuclear Safety Analysis and Design, ORNL/TM-2005/39, Version 6.1, Oak Ridge National Laboratory, Oak Ridge, Tennessee, June 2011.

17. C. H. Lee and W. S. Yang, "MC<sup>2</sup>-3: Multigroup Cross Section Generation Code for Fast Reactor Analysis," ANL/NE-11-41, January, 2012.
18. A. Marin-Lafleche, M. A. Smith, E. E. Lewis, and C. H. Lee, "Development Status of PROTEUS-MOC," ANL/NE-12/46, September 2012.
19. C. H. Lee (ANL) and H. C. Lee (KAERI), et al., Argonne National Laboratory, Unpublished Information, February 2012.
20. C. H. Lee and W. S. Yang, "Verification and Validation of Multigroup Cross Section Code MC<sup>2</sup>-3 for Fast Reactor System," *Trans. Am. Nucl. Soc.*, **106**, 711-714, 2012.
21. C. H. Lee and W. S. Yang, "An Improved Resonance Self-shielding Method for Heterogeneous Fast Reactor Assembly and Core Calculation," M&C 2013, Sun Valley, ID, May 5-9, 2013.
22. T. Tone, "A Numerical Study of Heterogeneity Effects in Fast Reactor Critical Assemblies," *J. Nucl. Sci. Technol.*, **12**, 467, 1975.
23. H. Yu, T. Endo, and A. Yamamoto, "Resonance Calculation for Large and Complicated Geometry Using Tone's Method by Incorporating the Method of Characteristics," *J. Nucl. Sci. Technol.*, **48**, 3, 330-336, 2011.
24. R. E. Alcouffe, F. W. Brinkley, D. R. Marr, and R. D. O'Dell, "User's Guide for TWODANT: A Code Package for Two-Dimensional, Diffusion-Accelerated, Neutral-Particle Transport," LA-10049-M, Los Alamos National Laboratory, 1990.
25. C. H. Lee, Z. Zhong, W. D. Pointer, and W. S. Yang, Argonne National Laboratory, Unpublished Information, June 2010.
26. H. J. Shim and C. H. Kim, "Error propagation module implemented in the MC-CARD Monte Carlo code," *Trans. Am. Nucl. Soc.* **86**, 325, 2002.
27. Doxygen webpage: <http://www.stack.nl/~dimitri/doxygen/index.html>.
28. FRUIT webpage: <http://fortranxunit.sourceforge.net/>.
29. VisIt webpage: <http://visit.llnl.gov>.
30. A. Godfrey, "VERA Core Physics Benchmark Progression Problem Specifications," CASL-U-2012-0131-002, <http://www.casl.gov/publications.shtml>, 2013.
31. K. S. Kim, M. L. Williams, M. A. Jessee, and D. Wiarda, "Subgroup Data Generation for the Resonance Self-Shielding Calculation in the NEAMS Neutronics Simulation," ORNL/LTR-2013/427, September 2013.

## Appendix A. Major Subroutine and Function Headers of the Subgroup API

```
!> Initialize the subgroup API
!! - Duplicate the parallel communicator
!! - Initialize the HDF5 interface
!! - Setup the error output
!! - Setup the library path and retrieve library main info.
!! @param Communicator Parallel communicator (in energy)
!! @param LibraryPath Path to the library file
!! @param ErrorUnit Logical unit to use for error message output
!! @param IERR returned error
SUBROUTINE SG_Initialize(Communicator,LibraryPath,ErrorUnit,IERR)
  IMPLICIT NONE

  INTEGER, INTENT(IN) :: Communicator
  CHARACTER (*), INTENT(IN) :: LibraryPath
  INTEGER, INTENT(IN) :: ErrorUnit
  INTEGER, INTENT(OUT) :: IERR

END SUBROUTINE SG_Initialize
```

```
!> Retrieve the number of neutron energy groups in the library
!! @param NNG Number of neutron groups
!! @param IERR Returned error code
SUBROUTINE SG_GetNumberOfNeutronGroups(NNG,IERR)
  IMPLICIT NONE

  INTEGER, INTENT(OUT) :: NNG
  INTEGER, INTENT(OUT) :: IERR

END SUBROUTINE SG_GetNumberOfNeutronGroups
```

```
!> Retrieve the maximum legendre order used in the library
!! @param NL Maximum order for legendre scattering kernel
!! @param IERR Returned error code
SUBROUTINE SG_GetMaxLegendreScatteringOrder(NL,IERR)
  IMPLICIT NONE

  INTEGER, INTENT(OUT) :: NL
  INTEGER, INTENT(OUT) :: IERR

END SUBROUTINE SG_GetMaxLegendreScatteringOrder
```

```
!> Retrieve the neutron groups energy boundaries
!! @param NNG Number of neutron groups
!! @param NBoundaries Boundaries for the neutron energy groups
!! @param IERR Returned error code
SUBROUTINE SG_GetNeutronGroupBoundaries(NNG,NBoundaries,IERR)
  IMPLICIT NONE

  INTEGER, INTENT(IN) :: NNG
  REAL, DIMENSION(:), INTENT(OUT) :: NBoundaries
  INTEGER, INTENT(OUT) :: IERR
```

```
END SUBROUTINE SG_GetNeutronGroupBoundaries
```

```
!> Retrieve the average neutron velocities in each energy group
!! @param NNG Number of neutron groups
!! @param NVelocities Average velocity of the neutrons in each energy group
!! @param IERR Returned error code
SUBROUTINE SG_GetNeutronGroupVelocities(NNG,NVelocities,IERR)
```

```
  IMPLICIT NONE
```

```
  INTEGER, INTENT(IN) :: NNG
  REAL, DIMENSION(:), INTENT(OUT) :: NVelocities
  INTEGER, INTENT(OUT) :: IERR
```

```
END SUBROUTINE SG_GetNeutronGroupVelocities
```

```
!> Retrieve the number of isotopes in the structure
!! @param NI Number of isotopes
!! @param IERR Returned error code
SUBROUTINE SG_GetNumberOfIsotopes(NI,IERR)
```

```
  IMPLICIT NONE
```

```
  INTEGER, INTENT(OUT) :: NI
  INTEGER, INTENT(OUT) :: IERR
```

```
END SUBROUTINE SG_GetNumberOfIsotopes
```

```
!> Retrieve the isotope IDs
!! @param NI Number of isotopes
!! @param IIds Array of isotope identifiers
!! @param IERR Returned error code
SUBROUTINE SG_GetIsotopeIds(NI,IIds,IERR)
```

```
  IMPLICIT NONE
```

```
  INTEGER, INTENT(IN) :: NI
  CHARACTER (SG_ISOTOPEID), DIMENSION(:), INTENT(OUT) :: IIds
  INTEGER, INTENT(OUT) :: IERR
```

```
END SUBROUTINE SG_GetIsotopeIds
```

```
!> Retrieve the isotope WEIGHTS
!! @param NI Number of isotopes
!! @param IWeights Array of isotope weightentifiers
!! @param IERR Returned error code
SUBROUTINE SG_GetIsotopeWeights(NI,IWeights,IERR)
```

```
  IMPLICIT NONE
```

```
  INTEGER, INTENT(IN) :: NI
  REAL, DIMENSION(:), INTENT(OUT) :: IWeights
  INTEGER, INTENT(OUT) :: IERR
```

```
END SUBROUTINE SG_GetIsotopeWeights
```

```
!> Retrieve the number of subgroup methods
!! @param NM Number of subgroup method supported by the library
!! @param IERR Returned error code
```

```
SUBROUTINE SG_GetNumberOfMethods (NM, IERR)
```

```
  IMPLICIT NONE
```

```
  INTEGER, INTENT(OUT) :: NM
```

```
  INTEGER, INTENT(OUT) :: IERR
```

```
  IERR = SG_ERROR_NOERROR
```

```
END SUBROUTINE SG_GetNumberOfMethods
```

```
!> Retrieve the number of subgroup methods Ids
```

```
!! @param NM Number of subgroup method supported by the library
```

```
!! @param MIds Subgroup method Ids
```

```
!! @param IERR Returned error code
```

```
SUBROUTINE SG_GetMethodIds (NM, MIds, IERR)
```

```
  IMPLICIT NONE
```

```
  INTEGER, INTENT(IN) :: NM
```

```
  INTEGER, DIMENSION(:), INTENT(OUT) :: MIds
```

```
  INTEGER, INTENT(OUT) :: IERR
```

```
END SUBROUTINE SG_GetMethodIds
```

```
!> Set the subgroup method to be used
```

```
!! @param MIds Method identifier
```

```
!! @param IERR Returned error code
```

```
SUBROUTINE SG_SetMethod (MId, IERR)
```

```
  IMPLICIT NONE
```

```
  INTEGER, INTENT(IN) :: MId
```

```
  INTEGER, INTENT(OUT) :: IERR
```

```
END SUBROUTINE SG_SetMethod
```

```
!> Set the range of locally owned energy groups
```

```
!! @param GStart First energy group owned by the local process
```

```
!! @param GEnd Last energy group owned by the local process
```

```
!! @param IERR Returned error code
```

```
SUBROUTINE SG_SetLocalGroups (GStart, GEnd, IERR)
```

```
  IMPLICIT NONE
```

```
  INTEGER, INTENT(IN) :: GStart
```

```
  INTEGER, INTENT(IN) :: GEnd
```

```
  INTEGER, INTENT(OUT) :: IERR
```

```
END SUBROUTINE SG_SetLocalGroups
```

```
!> Set the number of compositions to be defined
```

```
!! @param NC Number of compositions
```

```
!! @param IERR Returned error code
```

```
SUBROUTINE SG_SetNumberOfCompositions (NC, IERR)
```

```
  IMPLICIT NONE
```

```
  INTEGER, INTENT(IN) :: NC
```

```
  INTEGER, INTENT(OUT) :: IERR
```

```
END SUBROUTINE SG_SetNumberOfCompositions
```

```
!> Define one composition
!! @param CId Index of composition being defined
!! @param NI Number of isotopes in the composition
!! @param IIds Isotope identifiers
!! @param IDensities Isotope densities in the composition (in a/o)
!! @param IERR Returned error code
SUBROUTINE SG_DefineComposition(CId,NI,IIds,IDensities,IERR)
```

```
  IMPLICIT NONE
```

```
  INTEGER, INTENT(IN) :: CId
```

```
  INTEGER, INTENT(IN) :: NI
```

```
  CHARACTER (SG_ISOTOPEID), DIMENSION(:), INTENT(IN) :: IIds
```

```
  REAL, DIMENSION(:), INTENT(IN) :: IDensities
```

```
  INTEGER, INTENT(OUT) :: IERR
```

```
END SUBROUTINE SG_DefineComposition
```

```
!> Setup the cross-section region to composition mapping
!! @param NR Number of cross-section region
!! @param RMap Mapping from cross-section region to composition
!! @param IERR
```

```
SUBROUTINE SG_SetXSRegionMap(NR,RMap,IERR)
```

```
  IMPLICIT NONE
```

```
  INTEGER, INTENT(IN) :: NR
```

```
  INTEGER, DIMENSION(:), INTENT(IN) :: RMap
```

```
  INTEGER, INTENT(OUT) :: IERR
```

```
END SUBROUTINE SG_SetXSRegionMap
```

```
!> Setup procedure pointer to the fixed source solver
!! @param FSS Pointer to the fixed source solver procedure
!! @param IERR Returned error code
```

```
SUBROUTINE SG_SetFixedSourceSolver(FSS,IERR)
```

```
  IMPLICIT NONE
```

```
  PROCEDURE (FSS_NoArg), POINTER :: FSS
```

```
  INTEGER, INTENT(OUT) :: IERR
```

```
END SUBROUTINE SG_SetFixedSourceSolver
```

```
!> Define temperature in each region
!! The first time this is done the fixed source calculation are run to compute
!! background cross-section levels
!! @param NR Number of cross-section regions
!! @param RegionTemperatures Array of temperatures in each region
!! @param IERR Returned error code
```

```
SUBROUTINE SG_SetRegionTemperature(NR,RegionTemperatures,IERR)
```

```
  IMPLICIT NONE
```

```
  INTEGER, INTENT(IN) :: NR
```

```
  REAL, DIMENSION(:), INTENT(IN) :: RegionTemperatures
```

```
INTEGER, INTENT(OUT) :: IERR
```

```
END SUBROUTINE SG_SetRegionTemperature
```





**Nuclear Engineering Division**

Argonne National Laboratory  
9700 South Cass Avenue, Bldg. 208  
Argonne, IL 60439

[www.anl.gov](http://www.anl.gov)



U.S. DEPARTMENT OF  
**ENERGY**

Argonne National Laboratory is a U.S. Department of Energy  
laboratory managed by UChicago Argonne, LLC