

University of Tennessee, Knoxville

TRACE: Tennessee Research and Creative Exchange

Doctoral Dissertations

Graduate School

5-2004

Generalized Multigroup Method

Ibrahim Khalil Attieh
University of Tennessee, Knoxville

Follow this and additional works at: https://trace.tennessee.edu/utk_graddiss

 Part of the Nuclear Engineering Commons

Recommended Citation

Attieh, Ibrahim Khalil, "Generalized Multigroup Method. " PhD diss., University of Tennessee, 2004.
https://trace.tennessee.edu/utk_graddiss/4528

This Dissertation is brought to you for free and open access by the Graduate School at TRACE: Tennessee Research and Creative Exchange. It has been accepted for inclusion in Doctoral Dissertations by an authorized administrator of TRACE: Tennessee Research and Creative Exchange. For more information, please contact trace@utk.edu.

To the Graduate Council:

I am submitting herewith a dissertation written by Ibrahim Khalil Attieh entitled "Generalized Multigroup Method." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Nuclear Engineering.

Ronald E. Pevey, Major Professor

We have read this dissertation and recommend its acceptance:

Robert E. Uhrig, Lawrence Townsend, Ohannes Karakashian

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a dissertation written by Ibrahim Khalil Attieh entitled "Generalized Multigroup Method." I have examined the final paper copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Nuclear Engineering.

Ronald E. Pevey, Major Professor

We have read this dissertation
and recommend its acceptance:

Accepted for the Council:

Vice Chancellor and Dean of
Graduate Studies

Generalized Multigroup Method

A Dissertation
Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Ibrahim Khalil Attieh
May 2004

Dedication

To: My parents Salma and Khalil Attieh and my sisters Ghada and Reem

Acknowledgments

I would like to thank my dissertation supervisor Dr. Ronald E. Pevey for his unfailing support and guidance in my research for this dissertation. I would also like to thank Dr. Robert E. Uhrig for his financial support and encouragement to complete my studies expeditiously. My dissertation committee members, Dr. Lawrence Townsend and Dr. Ohannes Karakashian have contributed significantly to my studies. Their help and guidance are greatly appreciated. Special thanks to Mr. Maurice Greene who has been instrumental in helping me with issues related to cross section processing and understanding the structure of the ENDF/B data.

The completion of my dissertation would not have been possible without Ms. Pam Armentrout's instrumental role in my obtaining a teaching assistantship in the Mathematics Department. Her help and support are greatly appreciated. Also, I would like to thank Dr. Sam Jordan, from the mathematics department, for his support and accommodation of my busy schedule during the dissertation write up.

My life has been greatly enriched, intellectually and academically, through my interactions with Dr. Art Ruggles, Dr. Belle Upadhyaya, Dr. Lawrence Miller, and Dr. Lee Dodds.

I would like to thank Mr. Gary Graves, Ms. Ellen Fisher, and Mrs. Kristin England for their great support during my studies in the department.

I also would to acknowledge the mentoring and friendship of Jim Harb. Jim lent me great support during difficult times while I was pursuing my Ph.D. degree.

My friends Saeed Sheshehgar, and John Dattilo were generous in their support and encouragement in finishing my dissertation.

Finally, I would like to thank my parents for their boundless love and warmth. I would also thank my sisters who are also my best friends for their support and encouragement.

Abstract

Treatment of energy in the solution of the transport equation has been dominated by the multigroup method (MG). The multigroup method has the advantages of being simple and mathematically robust. However, it suffers from two disadvantages. When the multigroup transport equation is solved to find the group flux coefficients, the assumed spectrum becomes discontinuous at the energy boundaries, and the assumed spectrum maintains its original shape within the group. These characteristics are in contradiction with actual spectrum behavior. These disadvantages reduce the accuracy of the multigroup method, requiring a greater number of energy groups to converge to a solution with the desired accuracy.

These deficiencies are caused by the use of histogram basis functions to define group membership. These basis functions are orthogonal to each other and have only one degree of freedom. Hence, the calculated group flux coefficients shift the spectrum vertically to give the optimal balance between the sources and sinks in the transport problem.

To mitigate these inefficiencies of the multigroup method, a generalized multigroup method is proposed. In this method, an arbitrary set of basis functions is defined. By utilizing more appropriate basis functions, the spectrum is able to adapt within each energy group and be continuous at the boundaries. Improvement in the calculation of the spectrum will result in a more accurate solution of the equation.

In addition to defining the generalized multigroup method, this research also implements the hat basis function to allow the spectrum to adapt linearly within each energy group and be continuous at the boundaries. The hat basis function implementation is called the Linear Multigroup Method (LMG).

The LMG is then tested using two gamma ray spectrum calculation problems in an infinite oxygen medium. The results of these sample problems for the LMG are compared to the conventional MG method. This comparison shows that the LMG is superior to the MG in energy regions in which the flux spectrum is continuous, but less so for regions of discontinuous spectrum (e.g., near the 511 keV pair production spike or near discrete source energies). To deal with these special cases a hybrid LMG/MG method is developed and shown to give more accurate results.

TABLE OF CONTENTS

1. Introduction and Background **1**
1.1 Background 2
1.1.1 Boltzmann Transport Equation 2
1.1.2 Multigroup Method 7
1.1.3 Difficulties of Multigroup Method 10
1.2 Objectives of Dissertation 12
1.3 Original Contribution 12
1.4 Organization of the Dissertation 13

2. Review of the Literature **15**
2.1 Spatial Shape Improvements 15
2.2 Spatial Homogenization 16
2.3 Ultrafine Group Collapsing 16
2.4 Multiband Methods 19

3. ENDF Data and Codes Review **23**
3.1 ENDF Data 23
3.1.1 CSEWG 24
3.1.2 ENDF/B Format Overview 26
3.2 Data Processing Codes 29

4. Theory **31**
4.1 The Generalized Multigroup Theory 32
4.2 Multigroup Method from the GM Method 34
4.3 Linear Multigroup Method 37
4.4 Negative Fluxes and Hybrid LMG/MG Method 41

5. Modification of NJOY99 for LMG **43**
5.1 NJOY99 Modules 45
5.1.1 MODER Module 45
5.1.2 RECONR Module 45
5.1.3 BROADR Module 46
5.1.4 UNRESR Module 47
5.1.5 THERMR 50
5.1.6 GROUPR Module 51
5.2 Modifying PANEL and GETFF to Produce LMG Cross Sections 52
5.2.1 Average Group Cross sections 56
5.2.2 Scattering Matrices 60
5.2.3 Fission Spectrum 65
5.3 Data Preparation for Gamma Rays 67
5.4 Data Assembly 69

6. Sample Problems 71
6.1 First Sample Problem: Prompt Fission Gamma Source 72
6.1.1 Source Description 72
6.1.2 Comparison of Results 74
6.1.3 Issues of Pair Production and Discrete Energy Sources 75
6.1.4 Results for Fission Spectrum Gamma Source 76
6.1.5 Negative Fluxes Due to Spectrum Mismatch 80
6.1.6 Results with Adjusted Energy Boundaries 82
6.1.7 Summary of Results for Fission Gamma Source 82
6.2 Second Sample Problem: Monoenergetic Gamma Ray Source 87
6.2.1 Source Description 87
6.2.2 Comparison of Results 88
6.2.3 Use of MG Method for Discrete Source Group 91
6.2.4 Ultra Fine MG Source and Pair Production Groups 92
6.2.5 Summary of Discrete Source Results 95

7. Conclusions 99

8. Recommendations for Future Work 103

REFERENCES 107

APPENDICES 111

APPENDIX A 113

APPENDIX B 121

APPENDIX C 129

APPENDIX D 139

APPENDIX E 149

APPENDIX F 153

VITA 157

LIST OF TABLES

Table **Page**

TABLE 3.1: TYPICAL ENDF/B SUBLIBRARIES AND NAMES 28
TABLE 3.2: DEFINITIONS OF FILES TYPES (MF) 29
TABLE 6.1: ABSORPTION RATE AND THE % ERROR IN THE THIRD ENERGY GROUP AS FUNCTION OF NUMBER OF ENERGY GROUPS 78
TABLE 6.2: ABSORPTION RATE AND % ERROR IN THE FOURTH ENERGY GROUP AS FUNCTION OF NUMBER OF ENERGY GROUPS 78
TABLE 6.3: THE ENERGY BOUNDARIES FOR EIGHT GROUPS FOR EQUAL LOGARITHMIC INTERVAL AND ADJUSTED INTERVALS 81
TABLE 6.4: ABSORPTION IN THIRD ORIGINAL ENERGY GROUP FOR ADJUSTED ENERGY BOUNDARIES 83
TABLE 6.5: ABSORPTION IN THE FOURTH ORIGINAL ENERGY GROUP FOR ADJUSTED ENERGY BOUNDARIES 83
TABLE 6.6: ABSORPTION RATE IN THE FOURTH ENERGY GROUP 89
TABLE 6.7: ABSORPTION RATE IN THE THIRD ORIGINAL GROUP USING HAT BASIS FUNCTION AND MG SOURCE AND PAIR PRODUTION GROUPS 93
TABLE 6.8: ABSORPTION RATE IN THE FOURTH ORIGINAL GROUP USING HAT FUNCTIONS WITH MG SOURCE AND PAIR PRODUTION GROUPS 93
TABLE 6.9: ABSORPTION RATE IN THIRD ORIGINAL ENERGY GROUP AS FUNCTION OF NUMBER OF GROUPS FOR ULTRAFINE SOURCE AND PAIR PRODUCTION GROUPS 96
TABLE 6.10: ABSORPTION RATE IN THE FOURTH ENERGY GROUP AS FUNCTION OF NUMBER OF GROUPS FOR ULTRAFINE SOURCE AND PAIR PRODUCTION GROUPS 96

LIST OF FIGURES

Figure **Page**

FIGURE 1.1: TOTAL CROSS SECTION FOR ^{235}U 5
FIGURE 1.2: EFFECT OF CROSS SECTION ON PARTICLE FLUX AND REACTION RATE DISTRIBUTION. 8
FIGURE 1.3: PLOT OF THE ASSUMED, CALCULATED AND ACTUAL SPECTRUM FOR A THREE ENERGY GROUPS PROBLEM (REPRINTED WITH PERMISSION FROM REFERENCE 1). 11
FIGURE 2.1: MULTIBAND SUBDIVISIONS WITHIN AN ENERGY GROUP. (PRINTED WITH PERMISSION REFERENCE 2) 20
FIGURE 4.1: HISTOGRAM BASIS FUNCTIONS FOR CONVENTIONAL MULTIGROUP 36
FIGURE 4.2: LINEAR INTERPOLATION BETWEEN TWO POINTS 38
FIGURE 4.3: LINEAR BASIS FUNCTIONS FOR AN INTERVAL 38
FIGURE 4.4: HAT BASIS FUNCTIONS FOR LINEAR MULTIGROUP METHOD 40
FIGURE 5.1: TWO HALF SEGMENTS OF ADJACENT HAT BASIS FUNCTIONS THAT OVERLAP ORIGNAL MG ENERGY GROUP 53
FIGURE 5.2: SEGMENTS OF THE HAT FUNCTIONS THAT MAKE UP THE FIRST AND LAST ENERGY GROUPS 54
FIGURE 5.3: HAT BASIS FUNCTIONS UTILIZED TO CALCULATE THE SCATTERING CROSS SECTIONS FROM GROUP G TO GROUP K 61
FIGURE 5.4: ASSUMED SPECTRUM SPECIFIED BY NJOY99. IT CONSIST OF 1/E AND ROLL-OFF REGION TO ACCOUNT FOR PHOTOELECTRIC EFFECT 68
FIGURE 6.1: PROMPT FISSION GAMMA PHOTON SPECTRUM FOR U-235 73
FIGURE 6.2: SPECTRUM BECOMES NEGATIVE WHEN HAT FUNCTIONS ARE USED DUE TO DISCONTINUITIES CAUSED BY PAIR PRODUCTION GAMMAS AND DISCRETE ENERGY SOURCES. 77
FIGURE 6.3: THE ERROR IN ABSORPTION RATE IN THE THIRD ENERGY GROUP AS FUNCTION OF NUMBER OF ENERGY GROUPS. 79
FIGURE 6.4: ERROR IN ABSORPTION RATE IN THE FOURTH ENERGY GROUP AS FUNCTION OF NUMBER OF ENERGY GROUPS 79
FIGURE 6.5: NEGATIVE SPECTRUM DUE TO MISMATCH BETWEEN ASSUMED AND ACTUAL SPECTRUM 81
FIGURE 6.6: ABSORPTION CALCULATION ERROR IN THIRD ENERGY GROUP FOR ADJUSTED BOUNDARIES 84
FIGURE 6.7: ABSORPTION CALCULATION ERROR IN FOURTH ENERGY BY GROUP FOR ADJUSTED BOUNDARIES 84
FIGURE 6.8: ELIMINATION OF NEGATIVE OF FLUXES DUE TO MISMATCH BETWEEN ASSUMED AND ACTUAL SPECTRUM BY ADJUSTING GROUP ENERGY BOUNDARIES. THE SPECTRUM IS FOR EIGHT ENERGY GROUPS SOLUTION USING HYBRID LMG/MG METHOD 85

FIGURE 6.9: CALCULATED SPECTRUM USING 256 ENERGY GROUPS AND HAT/MG BASIS FUNCTIONS 85
FIGURE 6.10: COMPARISON OF SPECTRUM SHAPES FOR MG AND LMG/MG METHODS 87
FIGURE 6.11: ERROR IN ABSORPTION RATE IN FOURTH ENERGY GROUP AS FUNCTION OF NUMBER OF ENERGY GROUPS FOR MG METHOD AND LMG METHOD 90
FIGURE 6.12: SPECTRUM FOR A DISCRETE SOURCE USING 64 GROUPS AND HAT BASIS FUNCTIONS 90
FIGURE 6.13: AMPLIFICATION OF FIGURE 6.12 TO HIGHLIGHT THE OSCILLATORY BEHAVIOR OF THE SPECTRUM DUE TO DISCRRETE SOURCE AND PAIR PRODUCTION ENERGY RANGE. 91
FIGURE 6.14: ERRORS FOR MG AND LMG METHODS IN ABSORPTION PREDICTION OF DISCRETE SOURCE FOR CONVENTIONAL ENERGY STRUCTURE. 94
FIGURE 6.15 ERROR IN ABSORPTION RATE IN THIRD ORIGINAL GROUP AS FUNCTION OF NUMBER OF ENERGY GROUPS FOR ULTRAFINE SOURCE AND PAIR PRODUCTION GROUPS 97
FIGURE 6.16: ERROR OF ABSORPTION RATE IN THE ORIGINAL FOURTH ENERGY GROUP AS FUNCTION OF NUMBER OF ENERGY GROUPS 97

1. Introduction and Background

The reaction rate of neutral particles with the atoms of the medium through which they are traveling is an essential element of nuclear engineering. Every important aspect of design, operation, and safety analysis in nuclear engineering is dependent on knowledge of the different types of reactions occuring: fission, scattering, absorption, etc. The individual and the combined effects of these reactions hold the answers to the problems that nuclear engineers are trying to solve.

In the design of a nuclear reactor, it is important to calculate the multiplication factor of the neutrons in the core. This factor indicates whether the reactor is subcritical, critical, or supercritical. This information, among other things, allows the engineer to design a reactor that operates within safe limits. This factor can only be calculated if the fission, absorption, leakage, and scattering rates are well known. It is also important to know the reaction rates to keep track of fuel burn up.

In shielding applications, it is important to know that the absorbed radiation dose for plant operators and the public does not exceed safety limits. This radiation dose is calculated from the reaction rates of neutral particles inside human bodies because of a radiation source over the period of exposure.

In criticality safety applications, it is important that fissionable nuclear material is not placed in a configuration that will allow it to go supercritical accidentally. Hence, it is important to be able to calculate the criticality factor for the material under different

configurations to ensure that processing activities remain within safety limits. Again, reaction rates are needed for the analysis.

Neutral particles are born in one place and cause reactions in other places. Therefore, it is important to understand the transport of neutral particles in the spatial domain. However, the reaction rates are also angle, energy, and time dependent. Hence, neutral particle behavior has to be understood as a function space, angle, energy, and time.

Historically, much research and development effort has been expended on improving transport methods in spatial, angular, and time dimensions. However, since the beginning of the nuclear industry, dealing with the fundamental treatment of the energy domain of the transport problem has not received as much attention. Modern computer codes utilize one of two methodologies: full energy spectrum treatment or traditional multigroup treatment.

1.1 Background

1.1.1 Boltzmann Transport Equation

The distribution of neutrons and gamma rays in nuclear reactors is governed by the Boltzmann transport equation:

$$\frac{1}{v(E)}\frac{\partial}{\partial t}\phi(\vec{r},E,\hat{\Omega},t)+\Omega\cdot\nabla\phi(\vec{r},E,\hat{\Omega},t)+\Sigma_T(\vec{r},E,t)\phi(\vec{r},E,\hat{\Omega},t)= \int_0^\infty dE'\int_{4\pi} d\Omega'\Sigma(\hat{r},E'\rightarrow E,\hat{\Omega}'\rightarrow\hat{\Omega},t)\phi(\vec{r},E',\hat{\Omega}',t) + \chi(E)\int_0^\infty dE'\upsilon\Sigma_f(\vec{r},E',t)\int_{4\pi} d\hat{\Omega}\phi(\vec{r},E',\hat{\Omega},t)+S(\vec{r},E,\hat{\Omega},t) \quad \text{Eq. 1.1}$$

where

$\phi(\vec{r},E,\hat{\Omega},t)$ = Neutron flux per unit area, per MeV, per steradian, per unit time (particles/cm^2/MeV/ster/sec).

v(E) = Neutron speed (cm/sec) as a function of energy.

$\Sigma_t(\vec{r},E,t)$ = Total macroscopic cross section at position $\vec{r}$, energy E, and time t (cm^{-1}).

$\Sigma(\vec{r},E'\rightarrow E,\hat{\Omega}'\rightarrow\hat{\Omega},t)$ = differential macroscopic scattering cross section, describing the transfer of particles with initial coordinates of energy E', and solid angle $\hat{\Omega}'$, final energy E, final solid angle $\hat{\Omega}$, and time.

$\Sigma_f(\vec{r},E,t)$ = Fission macroscopic cross section at location $\vec{r}$ and time t for a particle of energy E.

ν = Average number of neutrons per fission process (particles/fission)

$S(\vec{r},E,\hat{\Omega},t)$ = Independent neutron source (particles/cm^2/MeV/unit solid angle/sec).

The reaction rates can be calculated using:

$$R_x(\vec{r})=\int_0^\infty \Sigma_x(\vec{r},E)\phi(\vec{r},E)dE=\sum_i N_i(\vec{r})\int_0^\infty \sigma_{xi}(\vec{r},E)\phi(\vec{r},E)dE \qquad \text{Eq. 1.2}$$

where

$R_x(\vec{r})$ = Reaction rate for reaction x [#/cm^3 / sec.]

$\Sigma_x(\vec{r},E)$ = Macroscopic cross section for reaction x for isotope i [cm^{-1}]
$\sigma_{xi}(\vec{r},E)$ = Microscopic cross section for reaction x for isotope i [cm^2]
$\phi(\vec{r},E)$ = Neutron scalar flux [# of particles/cm^2/sec.]
$N_i(\vec{r})$ = Atoms of isotope i per unit area [#/cm^3]

The macroscopic cross section corresponds to the probability that a particle will undergo a certain reaction per centimeter of travel.

The atomic cross section is the relative probability that a certain reaction takes place, given that an atom is bombarded with a neutral particle with a given kinetic energy. Therefore, this probability is a function of which atom is being bombarded, the energy of the particle, and the temperature of the atom. Each atom has its own unique cross section that is a function of its nuclear structure. For example boron is an extremely good neutron absorber. On the other hand, the hydrogen is a poor neutron absorber. The cross section also varies significantly as function of energy. For example, U235 is a relatively poor absorber of neutrons at high energy, but is extremely good at absorbing neutrons in the resonance and thermal regions. A good illustration of the variation of the cross section as function of energy is shown in Figure 1.1. In this figure, the total cross section of U235 is shown to vary from between 10 barns and 4x10^4 barns over the energy range. This is a change of more than three orders of magnitude.

The measurement of microscopic cross sections is a field of specialization in itself. It requires elaborate experimental facilities, complex mathematical modeling of the atomic

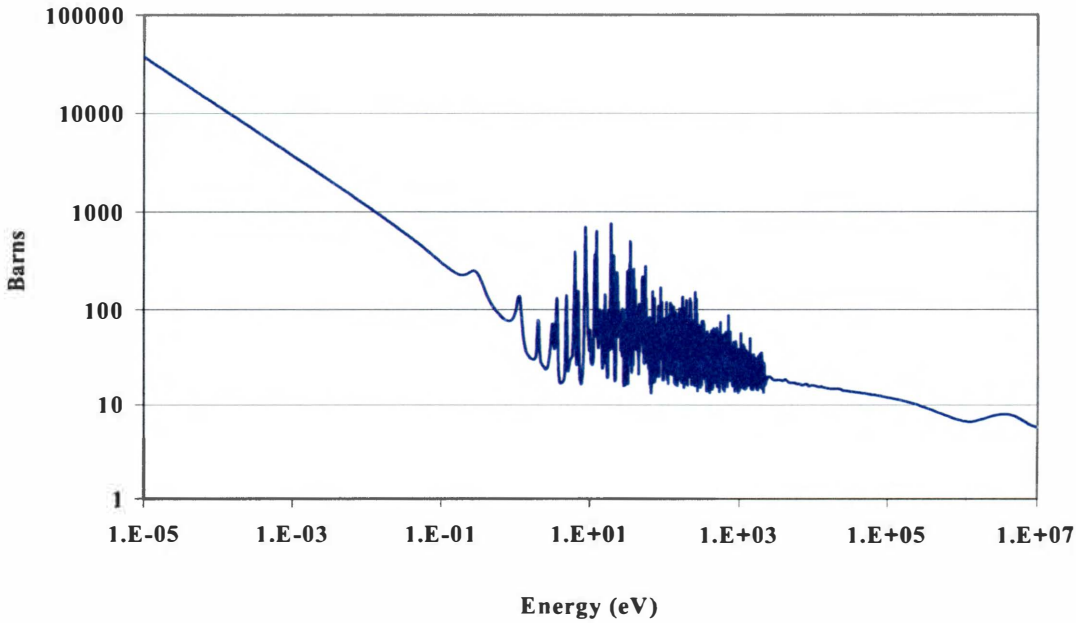


Figure 1.1: Total Cross section for ^{235}U

nuclear structure, and highly skilled personnel for the experimental and theoretical work. Even after measurements are carried out, the evaluation of the cross sections and dissemination of the information to the nuclear industry are complex processes. These processes are designed to assure consistency and accuracy of the data that are used in the nuclear industry. Despite the complexities involved in cross section measurements, these cross sections are readily available for use by the nuclear engineering community. They are always being refined and their accuracy improved.

The Boltzmann transport equation is solved to determine the particle distribution throughout the reactor. It accounts for all the sources and sinks of neutral particles as a function of energy, space, direction, and time. The equation states that the rate of change in the particle flux at a given spatial point, energy, direction, and time must equal the

difference between the production and loss rates at these given points. Therefore, it conserves particles throughout the medium.

Sources of neutral particles are neutrons born due to fission, neutrons born due to fission product decay, gamma rays due to fission, fission product decay, constant sources, and in-flow of neutral particles from outside the region of interest. Neutral particle sinks are neutron or gamma ray losses due to interaction and leakage. Analytical solution of the transport equation is not feasible due to complexities of most realistic geometries, and also due to the cross section dependence on energy and position.

Since an analytical solution for real problems is not feasible, the only approaches left to solve the equation are numerical or stochastic methods. For fission problems, the energy of neutrons varies from 10^{-5} eV to $2x10^{7}$ eV. The variation in cross section magnitudes over this span is drastic, and can be many orders of magnitude. To solve a problem involving resonance nuclides in the energy domain alone using the finite difference method would require over 100,000 energy subdivisions. If the proper resolution of the problem is to be carried out in the spatial dimension also, then the number of subdivisions increases exponentially as more spatial dimensions are added. If 1000 mesh points are needed per spatial dimension, then the number of mesh points for a 1-D problem is 10^{8} points. The number of mesh points increases by a factor of 1000 for each extra spatial dimension.

1.1.2 Multigroup Method

The Boltzmann equation represents the flux as continuous function of space, direction, energy, and time; the magnitude of the cross sections also varies as function of space, energy, direction and time. Hence, when the transport equation is solved using finite difference methods, the intervals have to be sufficiently small such that there is no significant change in these cross sections between adjacent calculational intervals. Due to the extreme sensitivity of cross sections to energy, as shown in Figure 1.1, this approach will introduce an extremely large number of intervals. In addition, large changes in cross sections tend to produce a correspondingly (opposite direction) change in particle fluxes, as shown in Figure 1.2 [1]. To avoid this, the multigroup method was invented as an attempt to incorporate energy dependent details into the cross sections and flux distributions without such a large computational burden. The basic idea of the multigroup method is to replace the reaction rate calculation as a function of energy with a reaction rate balance over a small number (2-500) of energy intervals, referred to as energy groups.

To conserve reaction rates over each energy group, the detailed energy-dependent cross-section data are replaced by a single effective cross section for each group, defined as the integral of the product of the flux and cross section over energy interval, divided by the integral of the flux over the same energy interval. As an example, the effective total cross section over an energy group, is defined as:

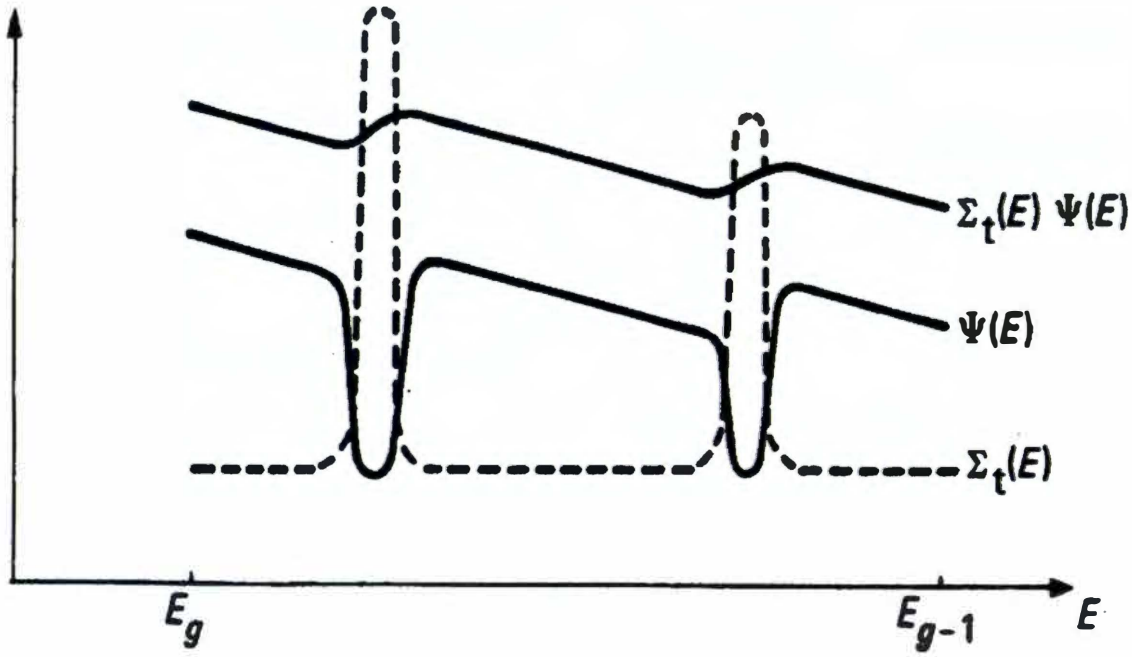


Figure 1.2: Effect of cross section on particle flux and reaction rate distribution.

$$\Sigma_{tg} = \frac{\int_{E_g}^{E_{g-1}} \Sigma_t(E)\Phi(E)dE}{\int_{E_g}^{E_{g-1}} \Phi(E)dE} \qquad \text{Eq. 1.3}$$

Even though this method of calculating effective cross sections formally conserves the reaction rate, use of the formula requires knowledge of the energy dependence of the flux. The multigroup method therefore requires approximate spectrum shapes to calculate the effective cross sections. With this approach, the accuracy of the reaction rates for each energy group depends on the accuracy of these assumed spectrum shapes.

The resulting time-independent multigroup Boltzmann equation for an infinite medium (i.e., with space, direction, and time suppressed) is:

$$\Sigma_{tg}\phi_g = \sum_{g'=1}^{G} \Sigma_s^{g'\to g}\phi_{g'} + \chi_g \sum_{g'=1}^{G} \nu\Sigma_f^{g'}\phi_{g'} + S_g \qquad g = 1, \ldots, G \qquad \text{Eq. 1.4}$$

where:

g, g' = energy group number.

G = total number of energy groups.

$$\phi_g \equiv \int_{E_g}^{E_{g-1}} \phi(E)dE \qquad \text{Eq. 1.5}$$

≡ group flux for energy group g.

$$\Psi_g \equiv \int_{E_g}^{E_{g-1}} \Psi(E)dE \qquad \text{Eq. 1.6}$$

≡ integral of the assumed spectrum over energy group g.

$$\Sigma_{tg}^{g} \equiv \frac{\int_{E_g}^{E_{g-1}} \Sigma_t(E)\Psi(E)dE}{\Psi_g} \qquad \text{Eq. 1.7}$$

≡ total group cross section for energy group g.

$$\Sigma_s^{g'\to g} \equiv \frac{\int_{E_g}^{E_{g-1}} dE \int_{E_{g'}}^{E_{g'-1}} \Sigma_s(E'\to E)\Psi(E')dE'}{\Psi_{g'}} \qquad \text{Eq. 1.8}$$

≡ scattering cross section from energy group g' to energy group g.

$$\chi_g \equiv \int_{E_g}^{E_{g-1}} \chi(E)dE \qquad \text{Eq. 1.9}$$

$\equiv$ the fraction of fission neutrons born in group g.

$$\nu\Sigma_{fg} \equiv \frac{\int_{E_g}^{E_{g-1}} \nu\Sigma_f(E)\Psi(E)dE}{\Psi_g} \qquad \text{Eq. 1.10}$$

$\equiv$ the fission cross section for group g times ν, the average number of neutrons per fission.

This multigroup equation is solved to find the group fluxes that balance the sources and the sinks of neutral particles in the problem. After solving for the group fluxes, the reaction rate for each energy group zone can then be calculated.

1.1.3 Difficulties of Multigroup Method

Since the assumed spectrum, which is an approximation of the true spectrum, is implicitly imbedded in the cross sections, the calculated group scalar fluxes act to shift the assumed spectrum shapes vertically to balance the sources and the sinks. The shift in the spectrum is shown in Figure 1.3 [1]. In this figure, the assumed, calculated, and actual spectrum for a hypothetical three energy group problem are presented. It is noticed that the assumed spectrum is not exactly like the actual spectrum. The calculated multigroup fluxes shift the assumed spectrum vertically to preserve the reaction rates over each group.

This spectrum treatment has a couple of built in disadvantages. The first disadvantage is that by shifting the spectrum vertically in each energy group, the spectrum becomes

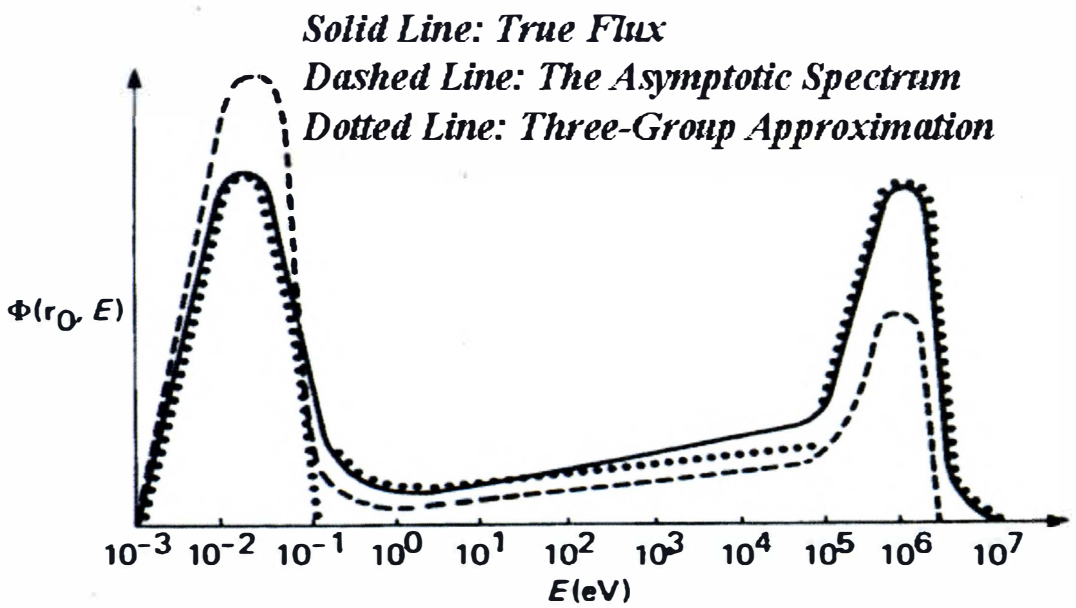


Figure 1.3: Plot of the assumed, calculated and actual spectrum for a three energy groups problem (reprinted with permission from Reference 1).

discontinuous. This is generally a non-physical artifact, since the spectra generally vary smoothly as a function of energy. The second disadvantage is that the within-group shape of the spectrum is fixed to the assumed shape; although the calculated group fluxes can correct group-integrated flux ratio errors, within-group flux shape errors are not corrected. To obtain a more accurate solution, many energy groups are needed to give a better approximation of the true spectrum. On the other hand, the multigroup method is numerically robust and mathematically simple. For these reasons, it has become a time-tested standard method of representing the energy domain despite the unrealistic spectrum that reduces the accuracy of the solution.

1.2 Objectives of Dissertation

The deficiencies that exist in the multigroup method, i.e. the discontinuity and rigidity of the assumed spectrum, present an opportunity for improvement in the solution of the transport equation. The objective of this dissertation is to tackle the principal deficiencies in the calculated spectrum: the discontinuity and the rigidity. These problems can be tackled by giving the spectrum new degrees of freedom to adjust to the physical situation of individual problems.

1.3 Original Contribution

A new generalized multigroup method was developed to deal with both of these deficiencies. In this new method, extra degrees of freedom are incorporated into the assumed spectrum shape by using non-orthogonal basis functions that allow the calculated spectrum to adjust to give an optimal fit to the physical spectrum. A constraint could also be added to the new basis functions to require them to be continuous at the energy boundaries. Hence, proper choice of these basis functions can resolve both of the deficiencies that are present in the conventional multigroup cross sections.

Thus the new generalized multigroup method, with properly chosen basis functions, should give a more accurate solution to the transport equation with fewer energy groups. The increase in the accuracy of the flux calculation will result in more accurate fuel burn up calculations, more reliable k_{eff} for criticality calculations, and more accurate transient analysis for safety analysis.

In addition to introducing the generalized multigroup theory, linear "hat" basis functions were introduced to derive the linear multigroup method (LMG). The hat functions have the advantages of allowing the spectrum to adapt linearly between the energy boundaries, and be continuous at those boundaries. The LMG was then implemented for gamma rays and a comparison was made between the conventional and the linear multigroup method for two infinite medium sample problems.

1.4 Organization of the Dissertation

In this first chapter, the background of cross section generation in the nuclear industry is introduced and the original contribution of this dissertation is presented and discussed. In Chapter 2, a literature review of previous research on energy treatments is carried out. In Chapter 3, the nuclear data format (ENDF/B) is discussed. In Chapter 4, the theory of generalized multigroup cross sections is introduced and the particular implementation of linear multigroup (LMG) theory is described and developed. In Chapter 5, the code NJOY99, which has been modified to generate the new cross sections, is discussed in detail, with particular attention given to how the theory has been implemented in NJOY99. The changes implemented to adapt the code to calculate LMG cross sections are presented. In Chapter 6, two sample problems involving gamma ray spectrum calculation in an infinite medium are presented and solved for two source conditions: prompt fission problem and monoenergetic source. Difficulties due to pair production in gamma ray problems are highlighted, along with proposed modifications to deal with negative fluxes. In Chapter 7, conclusions are outlined. In Chapter 8, future work on the generalized multigroup theory is proposed.

2. Review of the Literature

Effort to improve the accuracy of the transport equation has been of primary concern to researchers in the nuclear field because accurate solutions of neutral particle distribution have significant safety implications. Despite this, there has been surprisingly little effort spent on improving the treatment of the energy dimension of the transport equation: either a full energy treatment is used (limited to Monte Carlo methods) or the traditional multigroup method is used. Four areas or research that involve energy domain improvement are:

1. Improving the approximate spectrum shapes in an infinite medium for various energy regions.

2. Accounting for spatial self-shielding effects.

3. Using ultrafine energy groups to get a "brute force" best spectral fit under given conditions, and

4. Using of the multiband method to give the spectrum one or more degrees of freedom to adjust within an energy group.

2.1 Spatial Shape Improvements

It is well known that improving the approximation of the assumed spectrum will have positive impact [1,2,4,5,10,25,22], since the solution of the transport equation ultimately consists of shifting the spectrum to balance the sources and sinks in the problem.

Therefore much research effort has gone into improving these assumed spectra. Effort has been focused on a number of energy subdomains: the fission region, slowing region, unresolved resonance region, resolved resonance region (wide resonance, intermediate resonance, and narrow resonance), and thermal region [1-5,8,10,12,13,16,17]. The objective of this research is to identify spectrum shapes that best represent specialized energy subdomains to use for cross section collapsing. However, there has been little effort to improve the formalisms of multigroup theory that incorporates these improved spectra into the multigroup data.

2.2 Spatial Homogenization

Another area of research for improving the calculation of effective cross sections is spatial homogenization [1,2,4,29,31,32.33]. When calculations are a carried on large spatial intervals, such as the case for a reactor core, it is important that cross sections are available for homogenous spatial regions. As in the energy domain, there are spatial self-shielding effects that have to be taken into account in collapsing group cross sections over such spatial cells [29,32,33].

2.3 Ultrafine Group Collapsing

The simplest solution to overcome the problem of evaluating the energy spectrum is to subdivide the energy domain into a large number of groups (as many as 100,000 groups), which is referred to as an ultrafine group approximation [1,2]. Detailed spectra calculated on this grid replace assumed shapes in the calculation of the few group cross sections. The approach to this problem assumes the spectrum to be the following:

$$\Psi(E) \approx \frac{1}{\Delta E_n} = \frac{1}{E_{n-1} - E_n} \qquad \text{Eq. 2.1}$$

where n is the energy group number, and ΔE is the width of the energy group n.

This results in the following ulrtafine group cross sections:

$$\nu\Sigma_f^n \equiv \frac{1}{\Delta E_n} \int_{\Delta E_n} \nu\Sigma_f(E)dE$$

≡ Fission cross section for group n.

$$\Sigma_T^n \equiv \frac{1}{\Delta E_n} \int_{\Delta E_n} \Sigma_T(E)dE$$

≡ Total cross section for group n.

$$\Sigma_s^{n'\to n} \equiv \int_{\Delta E_{n'}} \frac{dE}{\Delta E_{n'}} \int_{\Delta E_n} \Sigma_s(E' \to E)dE'$$

≡ Scattering cross section from group n' to group n

$$\chi^n \equiv \int_{\Delta E_n} \chi(E)dE$$

= Fraction of neutrons contributed due to fission group n' to group n.

Using the ultrafine group cross sections, the transport equation in the energy domain takes the following form:

$$\Sigma_T^n \phi_n = \sum_{n'=1}^{N} \Sigma_s^{n'\to n} \phi_{n'} + \chi_n \sum_{n'=1}^{N} \nu\Sigma_f^{n'} \phi_{n'} \qquad n = 1,....,N \qquad \text{Eq. 2.2}$$

After solving this equation, the few group cross sections can be computed using the following equations:

$$\Sigma_T^g \equiv \frac{\sum_{n\subset g} \Sigma_T^n \phi_n}{\sum_{n\subset g} \phi_n} \qquad g = 1, 2, ..., G$$

≡ Few group total cross section for group g.

$$\nu\Sigma_f^g \equiv \frac{\sum_{n\subset g} \nu\Sigma_f^n \phi_n}{\sum_{n\subset g} \phi_n}$$

≡ Few group fission cross section for group g.

$$\chi_g \equiv \sum_{n\subset g} \chi_n$$

≡ Fraction of fission neutrons born in group g.

$$\Sigma_s^{g'\to g} \equiv \frac{\sum_{n\subset g} \sum_{n'\subset g'} \Sigma_s^{n'\to n} \phi_{n'}}{\sum_{n'\subset g'} \phi_{n'}}$$

≡ Few group scattering cross section from group g' to group g.

In this manner, few group cross sections are calculated without the need to approximate spectrum shapes. It is a brute force approach to approximate the spectrum, but still gives accurate few group cross sections. However, it suffers from long computational times. This becomes a serious problem when calculations need to be carried out for nonuniform depletion cases, which results in materials' behavior becoming different in different regions.

2.4 Multiband Methods

In the multigroup method, the energy self-shielding is not considered to be spatially dependent. However, in reality, the self-shielding deep within resonance absorbers is much more pronounced than at its surface. This shortcoming of the multigroup method, even if it conserves reaction rate over the volume, results in erroneous calculations of the reaction rates as a function of space. This introduces error into fuel burn-up calculations, and in calculated heat distributions in fuel pin cells. The obvious method to overcome this problem is to subdivide the energy groups even further, which leads to the ultrafine group structure discussed in the previous section. As mentioned before, this is not feasible on routine problems.

To overcome this problem with the multigroup theory, the multiband theory was developed [2,4]. In the multiband theory, the groups are further subdivided along the magnitude of the cross sections, as shown in Figure 2.1 [4]. With this approach, the resonance cross sections are given a degree freedom to allow them to adjust their amplitude as selfshielding effects change spatially. This will result in better fuel burn-up calculations and better fission heat distribution. The multiband cross sections are evaluated in a similar manner as the multigroup cross sections. However, the cross sections are integrated over the energy bands.

$$\phi_{gb} \equiv \int_{E_g}^{E_{g-1}} dE \int_{\Sigma_{Tgb}}^{\Sigma_{Tgb+1}} d\Sigma_T^*(E) H_{gb}(E) \Psi(E) \qquad \textbf{Eq. 2.3}$$

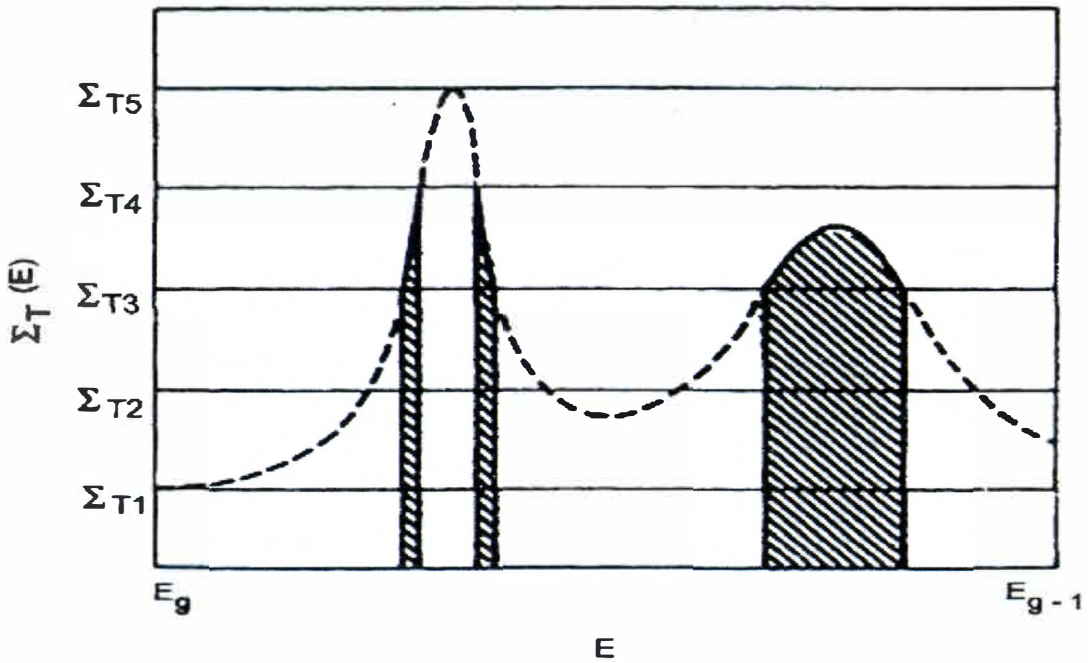


Figure 2.1: Multiband subdivisions within an energy group. (Printed with permission reference 2)

$$\Sigma_T^{gb} \equiv \frac{\int_{E_g}^{E_{g-1}} dE \int_{\Sigma_T^{gb}}^{\Sigma_T^{gb+1}} d\Sigma_T^*(E) H_{gb}(E) \Sigma_T(E) \Psi(E)}{\Psi_{gb}} \qquad \textbf{Eq. 2.4}$$

$$\Sigma_s^{g'b' \to gb} \equiv \int_{E_g}^{E_{g-1}} dE \int_{E_{g'}}^{E_{g'-1}} dE' \int_{\Sigma_{Tb}}^{\Sigma_{Tb+1}} d\Sigma_t^*(E) \times \int_{\Sigma_{Tb'}}^{\Sigma_{Tb'+1}} d\Sigma_T^* H_{gb}(E) H_{g'b'}(E') \times \Sigma_s(E' \to E) \Psi(E') / \Psi_{g'b'} \qquad \textbf{Eq. 2.5}$$

$$H_{gb}(E) = \begin{cases} 1 & \Sigma_{tgb-1} < \Sigma_t(E) < \Sigma_{tgb} \\ 0 & \textit{Otherwise} \end{cases} \qquad \text{Eq. 2.6}$$

with the quantity Σ_T^* normalized such that

$$\sum_{b=1}^{B} \int_{E_{Tb}}^{E_{Tb+1}} d\Sigma_T^*(E) = 1 \qquad \text{Eq. 2.7}$$

Even though the multiband method improves the solution of the transport equation, it is not used extensively in industry. It is much more complex to calculate the cross sections, especially the scattering cross sections. The primary reason that it is not used extensively is that most major codes in the nuclear industry are not designed to generate and utilize multiband cross sections.

3. ENDF Data and Codes Review

In this chapter, the Cross Section Evaluation Working Group evolution and its role in the development of the ENDF cross section libraries is discussed. Furthermore, the data processing code NJOY99 will be discussed briefly [7].

3.1 ENDF Data

In nuclear engineering, wrong solutions can have catastrophic consequences for the people and the environment. If the fission sources in a criticality analysis are underestimated, this might result in a criticality accident. Or if the gamma or neutron absorption in a shielding problem is overestimated, this might result in the exposure of plant personnel to harmful radiation levels. To mitigate these problems, the nuclear industry expends considerable effort in providing analysts with accurate and consistent cross sections.

Obtaining the raw cross section data is a very difficult task, involving experimental and theoretical nuclear physics. Conferences, and workshops are organized that specifically deal with cross section measurement. The experimental data are obtained from accelerators, nuclear reactors, and fixed neutrons and gamma sources. Collecting the data requires complex experiments conducted using highly specialized instrumentations. Usually this effort is carried on at national and international levels. A number of national laboratories in the US, Europe, Russia, and Japan work collaboratively to ensure consistency in the results. Also, theoretical models are developed to obtain data in energy regions where experimental data are not available; the theoretical models are also utilized

to ensure that there is consistency between the experimental results and the nuclear models of the atoms [9].

As the fields of nuclear science and engineering evolved, the demand for nuclear data increased significantly. The size of experimental data collected became staggering. With the proliferation of reactor physics codes, the need for standardization of the storage and retrieval of the data became apparent. It also became important that the calculation of energy group cross sections be mathematically rigorous and repeatable. This urgency manifested itself when discrepancies in data sets from different national labs and different countries made codes' performance comparisons difficult.

These needs resulted in the establishment of the Cross Section Evaluation Working Group (CSEWG) [9]. The group has its own elaborate and peer reviewed procedures for collecting, organizing, and disseminating nuclear data for utilization by nuclear engineers and physicists. In this section, an overview of the evaluation and storage is discussed. A brief history of the CSEWG and its responsibilities in the production and distribution of evaluated cross sections is outlined.

3.1.1 CSEWG

The need for appropriate understanding of the data and the results obtained from reactor physics analysis requires the existence of data libraries that are consistent and mathematically rigorous to give repeatable results. The primary push for standardized libraries was made by Henry Honeck. Honeck credits his push for standard libraries to a

stimulating conversation between him and Allan Henry of Westinghouse and George Joanou of General Atomic at the bar of the Colony Restaurant in Washington D.C. [9]

According to Sol Pearlstein of Brookhaven National Laboratory (BNL), prior to the start of CSEWG in 1966, reactor physics analysis appeared to be in a state of disarray. There were discrepancies in the results between the different analysis systems and the experimental results. There was also a plethora of libraries and neutronics codes for analysis. In 1963, there were already a number of standard data libraries; the best known being the United Kingdom Data Library (UKNDL), the library for fast reactors at Karlsruhe (KDK), and Lawrence Livermore Laboratory Evaluated Nuclear Data Library (ENDL). However, there were enough differences in the libraries' formats and data to make the understanding of the causes of discrepancies in the results impossible.

At the first CSEWG meeting at BNL June 9-10, 1966, the Atomic Energy Commission issued a statement. It said in part: "One of the long range goals of the Reactor Physics Branch of the Division of Reactor Development and Technology is the development of a set of basic nuclear data which can be used to accurately predict the behavior of neutrons in a nuclear reactor. ... More meaningful comparisons can be made if certain select design calculations were made using the reference data. However, even in this case the comparison would not be meaningful if the reference set were not duly tested and shown to reproduce the essential physics." [9] Therefore, the thrust behind the CSEWG is to provide sets of neutronic and gamma ray data that give consistent results that best match the actual particles' behavior in the reactor. The CSEWG became a collaborative effort between the national laboratories, universities, and the nuclear industry. Its primary

objective was to develop the tools needed to provide the nuclear analysts with a standard and consistent set of nuclear data needed for reactor physics and shielding calculations.

The CSEWG has come up with two Evaluated Nuclear Data File (ENDF) formats. The first format is referred to as ENDF/A. In this format, the requirements on the data set are less stringent. The data do not have to cover the whole energy range and do not have to be complete. More than one set of data can be on file for each material. On the other hand, in the second file format, ENDF/B, the data are required to be complete and unambiguous; the data cover a predetermined energy range, and there is only one set of data for each material. During the first meeting, five subcommittees were set up to give guidance for the development of the ENDF/B. The committees are the following: Codes and Formats Subcommittee, Data Testing Subcommittee, Normalization Subcommittee, Resolved Resonance Region Subcommittee, and the Shielding Subcommittee.

In the four decades of its existence, CSEWG has fulfilled its main role and played an important role in the technological advancements in automation, quality assurance, artificial intelligence, network communication, documentation, and uncertainty analysis.

3.1.2 ENDF/B Format Overview

The ENDF/B format has gone through a number of upgrades since its first inception in 1966. The latest version is ENDF/B-VI. The primary objective of the ENDF system is to allow the storage and retrieval of evaluated nuclear data to be used in nuclear applications. The term "evaluated" is essential. It indicates that all of the necessary data in each evaluation are properly *evaluated and complete*. Any data that have not been

obtained through measurements have to be supplied using nuclear models. The ENDF system is a set of rigorous formats and procedures for storing and utilizing the given data. The rigor in the specifications of data storage and processing is to ensure that all of the calculations are repeatable.

The ENDF library is maintained at the National Nuclear Data Center (NNDC) at BNL. The library contains a number of evaluated data sets for various materials. The material is chosen by CSEWG from evaluations that are submitted for reviews. The decision to update data for a given nuclide evaluation is based on the following critera: (1) new significant experimental results become available, (2) integral tests show that the data give erroneous results, or (3) user's requirements indicate a need for a more accurate data and/or better representation of the data for a particular material. After inspection and testing, new or revised data sets are included in new release of ENDF/B library.

The library has a well-defined structure to ensure compatibility with many data processing codes [6]. The structure of the library is based on the following hierarchy. A library (NLIB) specifies the evaluation group (e.g. ENDF/B), a version number (NEVER) specifies the version of the format (e.g. ENDF/B-VI), a sub-library number (NSUB) specifies the particular data type, (such as incident-neutron data, or thermal neutron scattering), a format number (NFOR) specifies the format type in which the data is stored, a material number (MAT) gives the target material in the sub-directory (or the radioactive (parent) nuclide in a decay sub-library), a modification flag indicates the data update version, a file number (MF) identifies a subdivision in the material file that contains certain types of data (such as reaction cross sections, or angular distributions)

and a section number (MT), indicates a particular reaction, such as total cross section, or a particular type of auxiliary data. Typical sub-libraries and files are listed in Table 3.1 and Table 3.2.

The structured data storage in ENDF/B allows the users of many data processing codes to retrieve the data and produce consistent group cross sections for a number of applications. These cross sections can then be stored in various formats to allow for usage in different transport codes.

In summary, the ENDF/B format has made a vital contribution in the quality of reactor physics analysis. The rigorous evaluation and testing procedures of the cross sections has made it possible to have a well documented set of data that is quality assured and robust. Furthermore, it is now possible to compare different reactor physics methods using consistent cross sections. The standardization of data storage and tabulation has made the cross sections easily transportable from one format to another.

Table 3.1: Typical ENDF/B Sublibraries and Names

NSUB	IPART	ITYPE	Sub-library Names
0	0	0	Photo-Nuclear Data
1	0	1	Photo-Induced Fission Product Yields
4	0	4	Radioactive Decay Data
5	0	5	Spontaneous Fission Product Yield
10	1	0	Incident-Neutron Data
11	1	1	Neutron-Induced Fission Products Yields
12	1	2	Thermal Neutron Scattering Data

Table 3.2: Definitions of Files Types (MF)

MF	Description
1	General Information
2	Resonance parameters data
3	Reaction Cross section
4	Angular distributions for emitted particles
5	Energy distributions for emitted particles
6	Energy-angle distributions for emitted particles
7	Thermal neutron scattering law data
8	Radioactivity and fission-product yield data
9	Multiplicities for radioactive nuclide production
10	Cross sections for radioactive nuclide production
12	Multiplicities for photon production

3.2 Data Processing Codes

The standardization of the cross section data using the ENDF files did not result in the standardization of the group cross sections codes. There is a variety of codes that generate these cross sections, reflecting the differing needs of different applications. The diversity of codes also guarantees that cross sections are produced consistently. To ensure that a code does not have an error in calculating a cross section, it can be checked using other codes. A popular code for producing cross sections from ENDF/B data libraries is NJOY, which was developed and is maintained by the Los Alamos National Laboratory.

It produces neutron cross sections, and group-to-group scattering matrices, heat production cross sections, photon production matrices, photon interaction cross sections, and group-to-group matrices, delayed neutron spectra, thermal scattering cross sections and matrices, and cross-section covariances.

Since NJOY99 is the code that adapted for this research, its detailed structure and functionality are discussed in Chapter 5.

4. Theory

The multigroup method described in Chapter 1 can be generalized by extending the methodology for incorporating the assumed spectrum shape into the energy dependent Boltzmann transport equation. The traditional use of histogram basis functions to map energy values to particular energy groups presents limitations to the solution of the transport equation that can be overcome through an improved treatment.

The Generalized Multigroup (GMG) method treats the spectrum in a more flexible manner by removing restrictions on what group membership-defining basis functions are utilized in discretizing the continuous energy transport equation in the energy domain. The use of different basis functions will allow added degrees of freedom in allowing the assumed spectrum to adapt to problem conditions. The greater the number of degrees of freedom the greater the flexibility in fitting the spectrum to the solution. In this chapter, the GMG is developed from first principles. Secondly, the conventional multigroup method is shown to be a special case of the GMG. Thirdly, the hat basis functions are utilized as a test case for solving the transport equation with new basis functions. The hat basis functions have the advantages of allowing the spectrum to adapt linearly between the energy boundaries and retain spectrum continuity at boundaries. When the hat functions are used, the implementation is called the linear multigroup method (LMG). This adaptability allows the assumed spectrum to adjust to better match the true spectrum, hopefully reducing the number of required energy groups to converge to an acceptable solution.

4.1 The Generalized Multigroup Theory

To solve the transport problem in the energy domain, the flux as a function of energy is represented as the product of a guide function built from a basis function set and an assumed spectrum shape:

$$\phi(\vec{r},\hat{\Omega},E)=\sum_{g=1}^{G}\frac{f_g(E)\Psi(E)}{\Psi_g}\phi_g(\vec{r},\hat{\Omega},E) \qquad \text{Eq. 4.1}$$

where the group flux Ψ_g is defined to be

$$\Psi_g=\int_0^{\infty} f_g(E)\Psi(E)dE \qquad \text{Eq. 4.2}$$

The assumed flux with the group flux coefficients is then substituted into the infinite medium Boltzmann transport equation:

$$\begin{aligned}\Sigma_T(E)\phi(E)dE=&\int_0^{\infty}dE'\Sigma_s(E'\rightarrow E)\phi(E')\,dE\\&+\chi(E)\int_0^{\infty}dE'\nu\Sigma(E')\phi(E')dE+S(E)\end{aligned} \qquad \text{Eq. 4.3}$$

to get:

$$\begin{aligned}\Sigma_t(E)\sum_g\frac{\Psi(E)f_g(E)}{\Psi_g}\phi_g=&\int_0^{\infty}dE'\Sigma_s(E'\rightarrow E)\sum_{g=1}^{G}\frac{\Psi(E')f_g(E')}{\Psi_g}\phi_{g'}\\&+\chi(E)\int_0^{\infty}dE'\nu\Sigma_f(E')\sum_{g=1}^{G}\frac{\Psi(E')f_g(E')}{\Psi_g}\phi_{g'}+S(E)\end{aligned} \qquad \text{Eq.4.4}$$

This can be converted to an equation for each group by multiplying Eq. 4.4 by the basis function for group k and integrating over all energy:

$$\int_0^\infty \Sigma_t(E)\Psi(E)\sum_g \frac{f_g(E)f_k(E)}{\Psi_g}\phi_g dE = \int_0^\infty f_k(E)dE\int_0^\infty dE'\Sigma_s(E'\to E)\Psi(E')\sum_{g'}\frac{f_{g'}(E')}{\Psi_{g'}}\phi_{g'} + \int_0^\infty f_k(E)dE\chi(E)\int_0^\infty dE'\nu\Sigma_f(E')\Psi(E')\sum_{g'}\frac{f_{g'}(E')}{\Psi_{g'}}\phi_{g'} + \int_0^\infty f_k(E)S(E)dE \qquad \text{Eq. 4.5}$$

This reduces to:

$$\sum_{g=1}^{G}\Sigma_{tkg}\phi_g = \sum_{g=1}^{G}\Sigma_{skg}\phi_g + \chi_k\sum_{g=1}^{G}\nu\Sigma_{fg}\phi_g + S_k \qquad \text{Eq. 4.6}$$

where

$$\Sigma_{tkg} \equiv \int_0^\infty \frac{\Sigma_t(E)\Psi(E)f_g(E)f_k(E)}{\Psi_g}dE \qquad \text{Eq. 4.7}$$

$$\Sigma_{skg} \equiv \int_0^\infty f_k(E)dE\int_0^\infty \frac{f_g(E')\Sigma_s(E'\to E)\Psi(E')}{\Psi_g}dE' \qquad \text{Eq. 4.8}$$

$$\chi_k(E) \equiv \int_0^\infty f_k(E)\chi(E)dE \qquad \text{Eq. 4.9}$$

$$\nu\Sigma_{fk} \equiv \int_0^\infty \frac{\nu\Sigma_f(E)\Psi(E)f_k(E)dE}{\Psi_k} \qquad \text{Eq. 4.10}$$

$$S_k \equiv \int_0^\infty S(E)f_k(E)dE \qquad \text{Eq. 4.11}$$

The resulting coupled equations can be solved for the multigroup coefficients. The coefficients can then be used to reproduce the original spectrum using Eq. 4.1 and can be used to calculate the reaction rate for reaction x using:

$$R_x = \int_0^\infty \Sigma_x(E)\phi(E)dE = \sum_g \phi_k \Sigma_{xk} \qquad \text{Eq. 4.12}$$

where

$$\Sigma_{xk} = \int_0^\infty \frac{\Sigma_x(E)\Psi(E)f_k(E)dE}{\Psi_k} \qquad \text{Eq. 4.13}$$

4.2 Multigroup Method from the GM Method

In the traditional multigroup method, f_g(E) is a rectangular membership function with magnitude of 1 over the domain of group g and 0 elsewhere. If the traditional membership functions are substituted into the generalized multigroup method, Equations 4.6 – 4.11, then the multigroup equation, Equations 1.4 – 1.9 are obtained. To illustrate this, take the basis functions to be the following:

$$f_g(E) = \begin{cases} 1 & E_g < E < E_{g-1} \\ 0 & Otherwise \end{cases} \qquad \text{Eq. 4.14}$$

If this is substituted into Equation 4.10, for example, the result is:

$$\nu\Sigma_{fk} \equiv \frac{\int_0^\infty \nu\Sigma_f(E)\Psi(E)f_k(E)dE}{\int_0^\infty \Psi(E)f_k(E)dE}$$
$$= \frac{\int_{E_k}^{E_{k-1}} \nu\Sigma_f(E)\Psi(E)f_k(E)dE}{\int_{E_k}^{E_{k-1}} \Psi(E)f_k(E)dE} \qquad \text{Eq. 4.15}$$

which is equivalent to Eq. 1.10 (using Eq. 1.6 for the denominator).

The situation is a little more complicated for the total cross-section, but if Eq. 4.14 is substituted into Eq. 4.7, the result is:

$$\Sigma_{tkg} \equiv \frac{\int_0^\infty \Sigma_t(E)\Psi(E)f_g(E)f_k(E)dE}{\int_0^\infty \Psi(E)f_k(E)dE}$$
$$= \frac{\int_{E_k}^{E_{k-1}} \Sigma_t(E)\Psi(E)f_g(E)f_k(E)dE}{\int_{E_k}^{E_{k-1}} \Psi(E)f_k(E)dE}$$
$$= \begin{cases} \dfrac{\int_{E_k}^{E_{k-1}} \Sigma_t(E)\Psi(E)dE}{\int_{E_k}^{E_{k-1}} \Psi(E)dE} & \text{if } g = k \\ 0 & \text{if } g \neq k \end{cases} \qquad \text{Eq. 4.16}$$

This is equivalent to Eq. 1.7 for the case g = k, with off-diagonal terms not appearing in the traditional multigroup method. This result comes from the fact that the multigroup basis functions are orthogonal to each other, as shown in Figure 4.1.

$$f_k(E)f_g(E) = \begin{cases} 1 & if \quad k = g \\ 0 & if \quad k \neq g \end{cases} \qquad \text{Eq. 4.17}$$

Since the magnitude of the membership functions is nonzero only over the defined energy group, the integration from zero to infinity transforms the integrals of equations 4.7 to 4.11 into discrete integrals over each energy group.

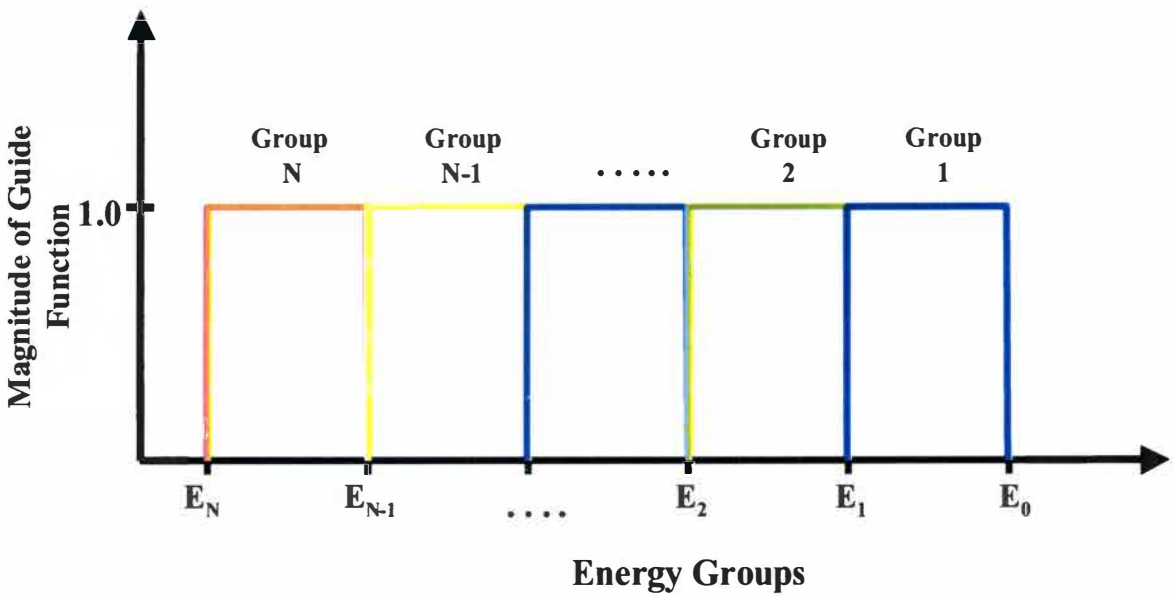


Figure 4.1: Histogram basis functions for conventional multigroup

4.3 Linear Multigroup Method

To demonstrate and test the effectiveness of using nontraditional basis functions, use of linear Lagrangian interpolation functions as the group basis function was investigated. Two points in a plane are as shown in Figure 4.2. The linear function that passes through the points (x_0, y_0) and (x_1, y_1) can be represented by the following linear polynomial:

$$P(x) = \frac{x - x_1}{x_0 - x_1} P(x_0) + \frac{x - x_0}{x_1 - x_0} P(x_1) \qquad \text{Eq. 4.18}$$

The function is constructed using the two basis functions in the domain (x_0, x_1):

$$f_0(x) = \frac{x_1 - x}{x_1 - x_0} \qquad \text{Eq. 4.19}$$

and

$$f_1(x) = \frac{x - x_0}{x_1 - x_0} \qquad \text{Eq. 4.20}$$

with the coefficients of each basis function being the values of the function at its endpoints (shown in Figure 4.3). If the idea is extended to allow a function to be approximated with N intervals, the resulting basis functions become:

$$f_0(x) = \frac{x_1 - x}{x_1 - x_0} \qquad x_0 < x < x_1 \qquad \text{Eq. 4.21}$$

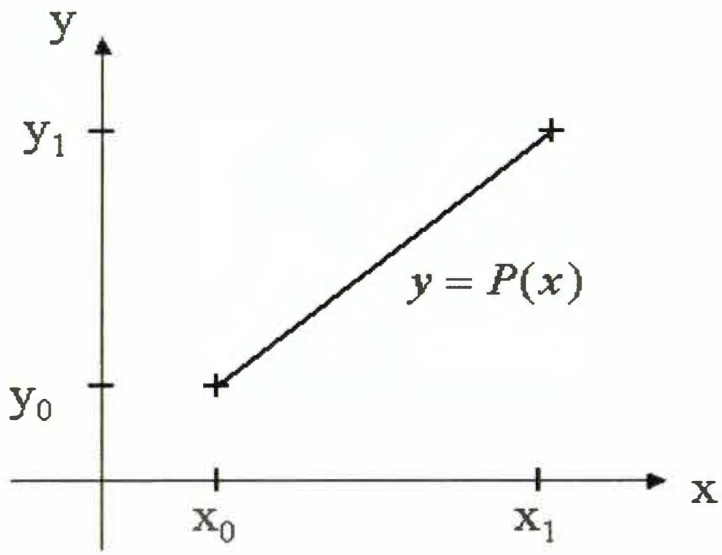


Figure 4.2: linear interpolation between two points

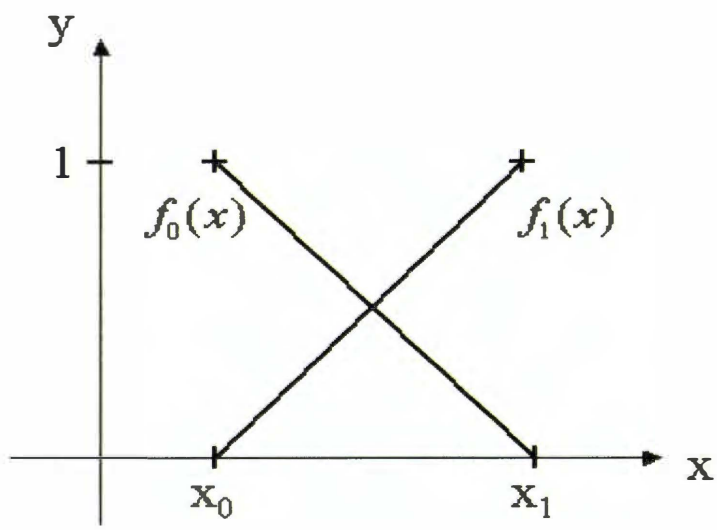


Figure 4.3: Linear basis functions for an interval

$$f_g(x) = \begin{cases} \dfrac{x - x_{g-1}}{x_g - x_{g-1}} & x_{g-1} < x < x_g \\ \dfrac{x_{g+1} - x}{x_{g+1} - x_g} & x_g < x < x_{g+1} \end{cases} \qquad \text{Eq. 4.22}$$

$$f_N(x) = \frac{x - x_{N-1}}{x_N - x_{N-1}} \qquad x_N < x < x_{N-1} \qquad \text{Eq. 4.23}$$

These basis functions are commonly used in finite element modeling because of their ability to adapt linearly and continuously between known values at the nodes. This set of basis functions was chosen for implementation because it overcomes the disadvantages of the traditional multigroup method: it allows (linear) adaptability of the spectrum within each energy group and imposes spectrum continuity at the energy boundaries, Figure 4.4.

Inserting these basis functions into Equations 4.6 to 4.11 gives a set of coupled equations that will be referred to as the linear multigroup method (LMG). The details of the resulting equations are presented in Chapter 5.

It is interesting to note that the resulting equation set has a structural feature not found in the conventional multigroup method: the total cross section has group-to-group dependence, like scattering cross sections. This is caused by the overlapping basis functions.

The overlap in the basis functions will similarly result in non-physical scattering from each group to the next higher group (which will be referred to as "fictitious" upscattering). Because of this overlap, a particle always exists in two different groups

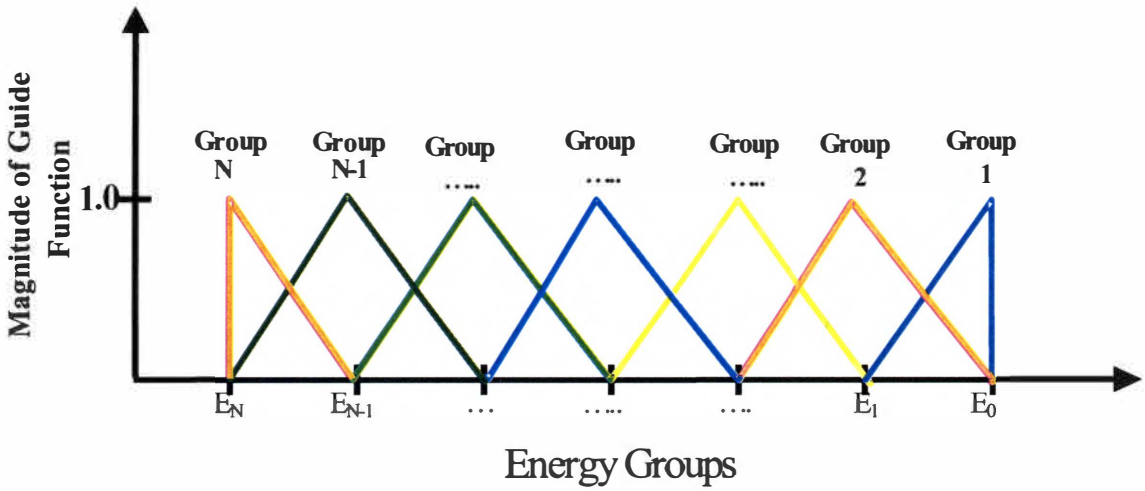


Figure 4.4: Hat basis functions for linear Multigroup Method

simultaneously. Hence, when a particle scatters within group, it will scatter partially into its own group, and partially into the energy group above it in energy. This will result in "mathematical" and not actual up scattering of particles. These two aspects of the hat functions alter the mathematical approach to solving the transport problem, but they do not alter any other physical assumptions about the problem to be solved. For example, when solving a transport using four energy groups above the thermal energy range, the conventional multigroup method requires that the following matrix be solved:

$$\begin{bmatrix} \Sigma_{t1}-\Sigma_{s11} & 0 & 0 & 0 \\ -\Sigma_{s12} & \Sigma_{t22}-\Sigma_{s22} & 0 & 0 \\ -\Sigma_{s13} & \Sigma_{t23}-\Sigma_{s23} & \Sigma_{t33}-\Sigma_{s33} & 0 \\ -\Sigma_{s14} & -\Sigma_{s24} & \Sigma_{t34}-\Sigma_{s34} & \Sigma_{t44}-\Sigma_{s44} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{bmatrix} \qquad \text{Eq. 4.24}$$

With no up scattering terms, the matrix is solved in one sweep solving each row one at a time. However, with the LMG "hat" basis functions, the group-to-group total and fictitious upscattering result in the following matrix:

$$\begin{bmatrix} \Sigma_{t11}-\Sigma_{s11} & \Sigma_{t21}-\Sigma_{s21} & 0 & 0 \\ \Sigma_{t12}-\Sigma_{s12} & \Sigma_{t22}-\Sigma_{s22} & \Sigma_{t32}-\Sigma_{s32} & 0 \\ -\Sigma_{s13} & \Sigma_{t23}-\Sigma_{s23} & \Sigma_{t33}-\Sigma_{s33} & \Sigma_{t34}-\Sigma_{s43} \\ -\Sigma_{s14} & -\Sigma_{s24} & \Sigma_{t34}-\Sigma_{s34} & \Sigma_{t44}-\Sigma_{s44} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{bmatrix} \qquad \text{Eq. 4.25}$$

With the presence of the upscattering terms, the matrix has to be solved iteratively, using standard numerical techniques (usually reserved for thermal groups) in which upscattering is taken into account.

4.4 Negative Fluxes and Hybrid LMG/MG Method

The objective of the LMG is to obtain a more realistic and accurate spectrum by forcing the spectrum to be continuous at the boundaries and adaptive linearly within each energy group. The LMG can be a disadvantage, though, in solving problems for which the calculated spectrum is expected to be discontinuous due to discontinuous sources, pair production (in gamma rays) or where there are energy discontinuities in cross sections. When such a spectrum is forced into the LMG framework, the flux coefficients in the group below the fixed source tend to become negative.

To mitigate this problem, a hybrid form of the Generalized Multigroup Method was also implemented and tested; in this hybrid approach certain pre-defined energy ranges – where flux discontinuities are expected – are represented by conventional MG while the

rest of the domain employs the continuous LMG approximation. This approach helps resolve the problem of obtaining negative fluxes close to discontinuities in the spectrum. The development and testing of this LMG/MG hybrid is discussed in later chapters.

5. Modification of NJOY99 for LMG

The incorporation of the basis functions into the multigroup cross sections has added adaptability to the assumed spectrum of neutral particles. This added flexibility comes at the cost of altering the definition of the group cross sections. To produce these cross sections, it was imperative that an industry code is utilized to generate the newly defined cross sections. This is because the objective is to demonstrate that industry codes, with minimal modification, can be used for generating the new cross sections.

The user of existing cross section codes such as NJOY99 has control over only a few input options that are needed to generate the group cross sections. Examples are the isotopes desired, the energy groups structure, the choice of assumed spectrum, and the Bondarenko factors. This rigidity is desired to minimize the possibility of the user entering erroneous data. It also simplifies the process of generating the cross sections. The disadvantage of such rigidity is that it makes the incorporation of new approaches more difficult.

The calculation of cross sections falls into two categories: cross sections for reactions that do not result in the readmission of new particles, and cross sections for reactions that do. For the first category, the energy dependent cross sections and assumed spectrum are simply flux-averaged over each energy group interval to get the MG cross sections. An example of this is the total microscopic cross section, which is calculated as follows:

$$\sigma_{tg} = \frac{\int_{E_{g-1}}^{E_g} \sigma_t(E)\Psi(E)dE}{\int_{E_{g-1}}^{E_g} \Psi(E)dE} \qquad \text{Eq. 5.1}$$

where σ_{tg} is the total microscopic cross section for group g, $\sigma_t(E)$ is the energy dependent total cross section, and $\Psi(E)$ is the assumed energy dependent spectrum.

For the scattering cross sections and fission source matrices, the calculation is slightly different. The cross sections are integrated over both the initial energy and the final energy groups, flux-weighting only the initial group. For example, the scattering matrix in NJOY99 is calculated as follows:

$$\sigma_{g'g} = \frac{\int_{g'} F(E')\sigma(E')\phi(E')dE'}{\int_{g'} \phi(E')dE'} \qquad \text{Eq. 5.2}$$

where

$$F_{lg}(E) = \int_{g'} dE' \int_{-1}^{+1} d\omega f(E' \rightarrow E, \omega) P_l(\mu(\omega)) \qquad \text{Eq. 5.3}$$

and $f(E' \rightarrow E, \omega)$ is the probability of scattering from E' to E through the center-of-mass cosine ω and P_l is a Legendre polynomial of the cosine of the deflection angle in the laboratory system cosine μ.

5.1 NJOY99 Modules

NJOY99 is a versatile modular code that produces all the multigroup cross sections required for nuclear analysis. It comprises a number of modules: MODER, RECONR, BROADR, UNRESR, HEATER, THERMR, GROUPR, GAMINR, ERRORR, DTFR, CCCR, and MATXSR. It utilizes ENDF data to produce the cross sections and the scattering matrices.

5.1.1 MODER Module

MODER is a simple module for transforming data files from ASCII format to binary format and vice versa. The purpose of this transformation is to speed up the calculation.

5.1.2 RECONR Module

RECONR reformats ENDF data from nonlinear interpolation schemes into linearly interpolable points, and reconstructs point resonance cross sections from resonance parameters, producing a new set of pointwise-ENDF data. The pointwise ENDF (PENDF) data are the values of the cross section data at particular energies. These data points are then used in the numerical integration of the data. These data for a particular isotope use the same energy grid for all reaction types, where the energy grid is chosen so that energy-dependent data can be reproduced within a specified error tolerance. Redundant cross sections that are the sum of other cross sections, such as the total cross section, are reconstructed from the sum of their constituents.

5.1.3 BROADR Module

Once the PENDF file is created, then the cross section broadening of resonance cross sections is handled by the BROADR code. It applies the "kernel broadening" method [3], which is an accurate method that treats the resonance and nonresonance cross sections, including the multi-level effects. The cross sections are broadened to user specified temperatures.

The kernel broadening method produces effective cross sections by taking into consideration the relative velocity of the particle and the target nuclei. The effective cross sections are given by:

$$\rho v\bar{\sigma}(v,T) = \int dv' \rho |v - v'| \sigma (|v - v'|) P(v',T) \qquad \text{Eq. 5.4}$$

where v is the velocity of the particle, v' is the velocity of the target, ρ is the density of the target, σ is the cross section for a stationary nuclei, and P(v',T) is the distribution of target velocities in the laboratory system. In many cases, the velocity distribution of the target nuclei can be described by a Maxwell-Boltzmann function. It is given by

$$P(v',T)dv' = \frac{\alpha^{3/2}}{\pi^{3/2}} \exp(-\alpha v'^2) dv' \qquad \text{Eq. 5.5}$$

where $\alpha = M/(2kT)$, k is the Boltzmann's constant, and M is the target mass.

5.1.4 UNRESR Module

From a resonance point of view, the energy domain is divided into four regions: low energy region, resolved resonance region, unresolved resonance region, and high energy region. It is divided into these regions to properly deal with the issue of self-shielding. In the low energy region, the cross sections are tabulated as smooth functions of energy. For light elements, whose natural resonance widths far exceed their Doppler width, there is a minimal Doppler effect and the smooth functions are sufficiently representative. For heavy elements, this method is sufficient below the resolved resonance region.

In the resolved resonance region, individual resonances are represented using parameters from common shape functions: single-level Breit-Wigner (SLBW), multilevel Breit-Wigner (MLBW), Adler-Adler (A-A), and Reich-Moore (RM) formalisms [1,4,19-21]. These resonances are converted to PENDF form by fitting with the appropriate parameters, and incorporated into the PENDF file. Because the resonances are well defined, the self-shielding effects can be taken into consideration by utilizing the narrow, intermediate, or wide resonance approximation.

Above the resolved energy range is the unresolved range, where individual resonances are present but cannot be experimentally resolved. Since they do not overlap, the self-shielding effect of these resonances still has to be taken into consideration. Since it is difficult to specify the parameters to reproduce these resonances, statistical techniques are employed to calculate the shielded effective cross sections. The UNRESR module in NJOY99 performs this step by finding the equivalent point-wise shielded cross sections to be added to the background cross sections.

More specifically, in the unresolved energy range, it is impossible to define the cross sections as function of energy. It is only possible to calculate effective or average cross sections over energy interval. The average cross section preserves the reaction rate:

$$\sigma_{0x}(E^*) = \frac{\int_{E_1}^{E_2} \sigma_x(E)\phi_0(E)dE}{\int_{E_1}^{E_2} \phi_0(E)dE} \qquad \text{Eq. 5.6}$$

where ϕ_0 is the scalar flux, E* is the effective energy in the range $[E_1,E_2]$. The energy range is chosen to be wide enough to contain a number of resonances, but small relative to the changes in the slowly varying function of E. It is important also to calculate the current weighted total cross section to calculate the effective values for the transport cross sections. This cross section is given by:

$$\sigma_{1x}(E^*) = \frac{\int_{E_1}^{E_2} \sigma_x(E)\phi_1(E)dE}{\int_{E_1}^{E_2} \phi_1(E)dE} \qquad \text{Eq. 5.7}$$

Then the issue becomes which assumed flux spectrum to use. NJOY99 assumes that for a homogenous system the scalar flux can be represented by the narrow resonance approximation of the spectrum. The flux is assumed to have the shape:

$$\phi_l(E) = \frac{C(E)}{[\Sigma_t(E)]^l} \qquad \text{Eq. 5.8}$$

where C(E) is a slowly varying function of energy. In NJOY, C(E) is assumed to be 1/E. However, it could also be a user defined slowly varying function of energy. The power l is set to zero for the scalar flux, and set to one when the flux is proportional to the current. To account for the spectral effects of other elements in the mixture, the constant σ_0 is incorporated into the flux approximation., with σ_0 defined as:

$$\sigma_0 \equiv \sum_{j \neq i} \frac{N_j \sigma_{Pj}}{N_i} \qquad \text{Eq. 5.9}$$

for

$N_i \equiv$ atom density of isotope being processed
$N_j \equiv$ atom density of other isotopes in mixture
$\sigma_{Pj} \equiv$ potential scattering of isotope j

The resultant assumed spectrum is given to be:

$$\phi_l(E) = \frac{C(E)}{[\sigma_0 + \sigma_t(E)]^l} \qquad \text{Eq. 5.10}$$

where C(E) is a slowly varying function of energy. NJOY99 uses 1/E or user a user defined function.

Therefore, the effective shielded point-wise cross section in the unresolved resonance is given by:

$$\bar{\sigma}_{0x}(E^*) = \frac{\int_{E_1}^{E_2} \frac{\sigma_x(E)}{\sigma_0 + \sigma_t(E)} C(E)dE}{\int_{E_1}^{E_2} \frac{1}{\sigma_0 + \sigma_t(E)} (E)dE} \qquad \text{Eq. 5.11}$$

with x being t for total, e for elastic, f for fission, and γ for capture, and

$$\bar{\sigma}_{1t}(E^*) = \frac{\int_{E_1}^{E_2} \frac{\sigma_t(E)}{[\sigma_0 + \sigma_t(E)]^2} C(E)dE}{\int_{E_1}^{E_2} \frac{1}{[\sigma_0 + \sigma_t(E)]^2} C(E)dE} \qquad \text{Eq. 5.12}$$

Once the equivalent shielded cross sections are calculated, then these point-wise cross sections are added to the PENDF file.

5.1.5 THERMR

In the thermal energy range, the kinetic energy of the neutrons is the same order of magnitude as the kinetic energy of the atoms with which they are colliding. Therefore, the neutrons do not always lose energy when they scatter, but they can also gain energy. Since the energy of the neutrons has reached thermal equilibrium with the medium, the kinetic energies of the neutrons are comparable to the kinetic energy of the atoms in the medium. Inelastic cross sections are produced either using the free gas model or the $S(\alpha, \beta)$ functions available for bound scatterers.

5.1.6 GROUPR Module

The last step for obtaining the group cross sections is by running the module GROUPR. GROUPR is a self-contained module that utilizes the energy dependent cross sections and spectral shapes produced by the previous modules to produce all the group cross sections.

The generalized integral that NJOY99 GROUPR uses is:

$$\sigma_g = \frac{\int_g F(E)\sigma(E)\phi(E)dE}{\int_g \phi(E)dE} \qquad \text{Eq. 5.13}$$

where F(E) is called the "feed function", whose definition depends on cross section type. This function is set equal to one when average neutron cross sections are calculated. For photon production, F(E) is the photon yield. As for scattering matrices, F(E) is the *l*-th Legendre component of the normalized probability of scattering into a secondary energy group g from initial energy E. It is defined as:

$$F_{lg}(E) = \int_{g'} dE' \int_{-1}^{+1} d\omega f(E' \rightarrow E, \omega) P_l(\mu(\omega)) \qquad \text{Eq. 5.14}$$

where $f(E' \rightarrow E, \omega)$ is the probability of scattering from E' to E through the center-of-mass cosine ω and P_l is a Legendre polynomial for laboratory cosine μ.

In NJOY, these cross sections are calculated in two subroutines: PANEL and GETFF. The PANEL subroutine is where the integration of the average and scattering matrices

integrals is carried out. GETFF is where the feed function for the scattering matrices is calculated. Therefore, any changes in NJOY99 have to involve these subroutines.

5.2 Modifying PANEL and GETFF to Produce LMG Cross Sections

Modification of NJOY99 to calculate the LMG cross sections requires modifications of:

1. the functions integrated;
2. the domain of the integration; and
3. the number of integrations performed.

An example is the calculation of the LMG total cross section, which is defined to be:

$$\sigma_{tgk} = \frac{\int_0^\infty \Sigma_t(E)\Psi(E)f_g(E)f_k(E)dE}{\Psi_g} \qquad \text{Eq. 5.15}$$

where Ψ_g is defined to be:

$$\Psi_g = \int_0^\infty f_g(E)\Psi(E)dE \qquad \text{Eq. 5.16}$$

To calculate the group flux, the product of the assumed spectrum and basis function for group g has to be integrated over the energy group boundaries. The hat basis functions have finite support base over the energy domain, which extends over two conventional energy groups. Each half of the hat function can be represented by a piece wise linear

function, as shown in Figure 5.1. The hat function can then represented by the following equation:

$$f_g(E)=\begin{cases} m_{rg+1}E+b_{rg+1} & E_{g+1}\le E\le E_g \\ m_{lg}E+b_{lg} & E_g\le E\le E_{g-1} \end{cases} \qquad g=2,3,\cdots,G \qquad \text{Eq. 5.17}$$

where E_g, E_{g-1}, and E_{g+1} are the energy boundaries of the conventional multigroup method, G is the number of multigroup energy groups.

There are two special cases: the first energy group and the last energy group. The first energy group has only the left segment of the hat function. The last group, G+1, has only the right segment of the hat function, as shown in Figure 5.2.

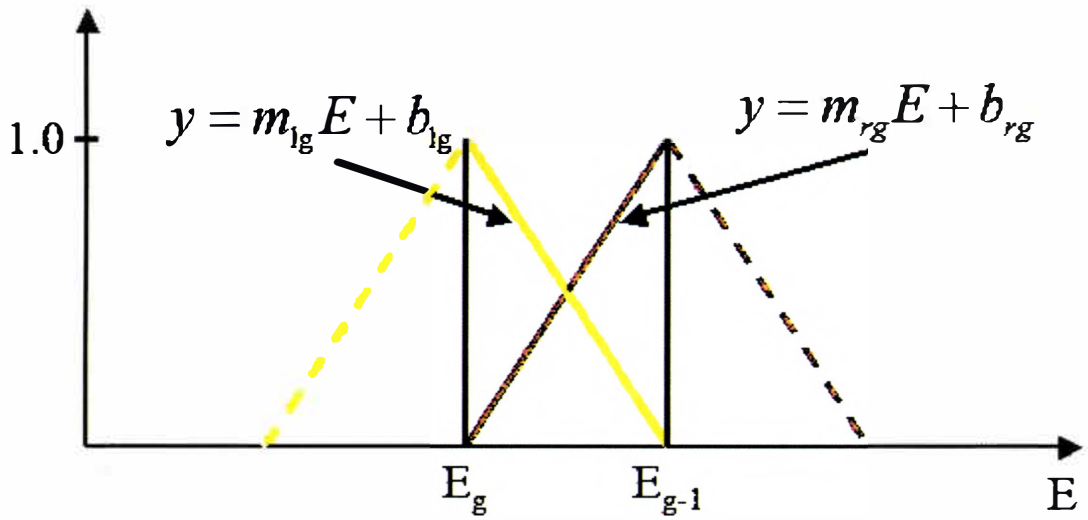


Figure 5.1: Two half segments of adjacent hat basis functions that overlap orignal MG energy group

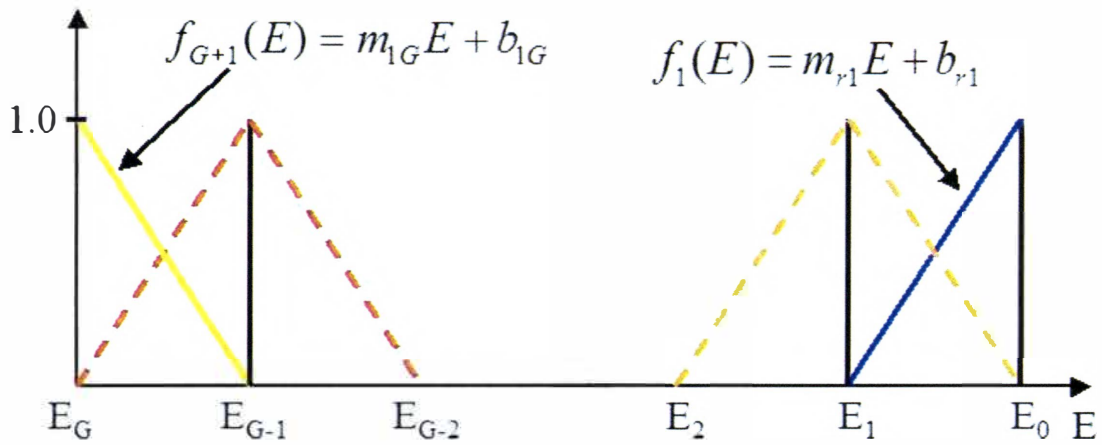


Figure 5.2: Segments of the hat functions that make up the first and last energy groups

These special cases result in the following basis functions:

$$\begin{aligned} f_1(E) &= m_{r1}E + b_{r1} \qquad E_1 \le E \le E_0 \\ f_{G+1}(E) &= m_{lG}E + b_{lG} \qquad E_G \le E \le E_{G-1} \end{aligned} \qquad \text{Eq. 5.18}$$

When these basis functions are incorporated into the integral of the group flux, the results are:

$$\phi_g = \int_{E_g}^{E_{g-1}} (m_{lg-1}E + b_{lg-1})\Psi(E)dE + \int_{E_{g+1}}^{E_g} (m_{rg}E + b_{rg})\Psi(E)dE \qquad \text{Eq. 5.19}$$

where the first integral represents the right segment of the hat function that spans energy group g for the MG energy group and the second integral represents the left segment of the hat function that spans group g+1. There are also the two special cases for the first and last energy groups. The group fluxes can be calculated using their own basis

functions as specified above. As a the result, the first and last group fluxes are the following:

$$\phi_1 = \int_{E_1}^{E_0} (m_{r1}E + b_{r1})\Psi(E)dE \qquad \text{Eq. 5.20}$$

$$\phi_{G+1} = \int_{E_G}^{E_{G-1}} (m_{lG}E + b_{lG})\Psi(E)dE \qquad \text{Eq. 5.21}$$

If the integrals for the group flux were expanded, then the equations would be:

$$\begin{aligned}\phi_g = m_{lg}\int_{E_g}^{E_{g-1}} E\Psi(E)dE + b_{lg}\int_{E_g}^{E_{g-1}} \Psi(E)dE \\ + m_{rg+1}\int_{E_{g+1}}^{E_g} E\Psi(E)dE + b_{rg+1}\int_{E_{g+1}}^{E_g} \Psi(E)dE\end{aligned} \qquad \text{Eq. 5.22}$$

The slopes and the intercepts are constants that are dependent on the energy group structure. Once the energy group structure has been specified, then these constants are determined. It is apparent that the group flux is dependent on two integrals: $\int_{E_{g+1}}^{E_g} \Psi(E)dE$ and $\int_{E_{g+1}}^{E_g} E\Psi(E)dE$. The first integral is exactly the same integral that NJOY99 and other cross section codes generate. However, the second integral is not generated by NJOY99 or any other code. Therefore, by modifying NJOY99 to allow the spectrum to be multiplied by E before carrying out the integral, the second integrand can be generated.

The subroutine PANEL, in NJOY99, was modified by transforming the group flux integral to be:

$$\phi_g = \int_{E_{g+1}}^{E_g} E^P \Psi(E) dE \qquad P = 0,1 \qquad \text{Eq. 5.23}$$

By energy E is raised to the power P. NJOY99 is then run twice, once with P equal to zero and once with P equal one. Since the slopes and the intercepts are pre-calculated, two runs of NJOY99 provide all the integrals needed to "reassemble" the hat functions from their left and right segments.

5.2.1 Average Group Cross sections

For generating average group cross sections, such as the total group cross section, the generalized multigroup method gives the following:

$$\Sigma_{tgk} = \frac{\int_0^\infty \Sigma_t(E)\Psi(E) f_g(E) f_k(E) dE}{\int_0^\infty f_g(E)\Psi(E) dE} \qquad \text{Eq. 5.24}$$

The group fluxes are calculated as mentioned above. Therefore, it is important to modify NJOY99 to calculate the numerator of the total cross section. The calculation of the group total cross section is a bit more difficult because of the overlap in the basis functions. This results in a "group-to-group" total cross section that does not exist in the conventional MG method. Since hat functions overlap two adjacent energy groups, this results in three possibilities when creating the cross section: k equals g, k equals g+1, or k

equals g-1. If k equals g, the basis functions are both from group g. For k equals g+1, the basis function for group g is overlapping with the basis function of the lower energy group g+1. For k equals g-1, the basis function for group g is overlapping with basis function of the higher energy group g-1. Each of the three situations results in a unique integral.

For the first case,

$$\Psi_g \sigma_{tgg} = \int_0^\infty \sigma_t(E)\Psi(E) f_g^2(E) dE \qquad \text{Eq. 5.25}$$

The integral is then decomposed into the two segments that make up the hat function.

$$\Psi_g \sigma_{tgg} = \int_{E_g}^{E_{g-1}} (m_{lg-1}E + b_{lg-1})^2 \sigma_t(E)\Psi(E) dE + \int_{E_{g+1}}^{E_g} (m_{rg}E + b_{rg})^2 \sigma_t(E)\Psi(E) dE \qquad \text{Eq. 5.26}$$

Expanding the integrals results in:

$$\Psi_g \sigma_{tgg} = \int_{E_g}^{E_{g-1}} (m_{lg-1}^2 E^2 + 2(m_{lg-1} + b_{lg-1})E^2 + b_{lg-1}^2)\sigma_t(E)\Psi(E) dE + \int_{E_{g+1}}^{E_g} (m_{rg}^2 E^2 + 2(m_{rg} + b_{rg})E + b_{rg}^2)\sigma_t(E)\Psi(E) dE \qquad \text{Eq. 5.27}$$

By expanding the integral further, the total cross section becomes:

$$\Psi_k \sigma_{tgg} = m_{lg-1}^2 \int_{E_g}^{E_{g-1}} E^2 \sigma_t(E)\Psi(E)dE + 2(m_{lg-1} + b_{lg-1}) \int_{E_g}^{E_{g-1}} E\sigma_t(E)\Psi(E)dE$$
$$+ b_{lg-1}^2 \int_{E_g}^{E_{g-1}} \sigma_t(E)\Psi(E)dE + m_{rg}^2 \int_{E_{g+1}}^{E_g} E^2 \sigma_t(E)\Psi(E)dE \qquad \text{Eq. 5.28}$$
$$+ 2(m_{rg} + b_{rg}) \int_{E_{g+1}}^{E_g} E\sigma_t(E)\Psi(E)dE + b_{rg}^2 \int_{E_{g+1}}^{E_g} \sigma_t(E)\Psi(E)dE$$

Again, the slopes and the intercepts for the basis functions are pre-calculated based on the energy group boundaries. Two of the integrals, $\int_{E_g}^{E_{g-1}} \sigma_t(E)\Psi(E)dE$ and $\int_{E_{g+1}}^{E_g} \sigma_t(E)\Psi(E)dE$ are the standard integrals that NJOY99 produces for group g and g+1. Therefore, a normal run of NJOY99 will produce these cross sections.

However, to produce the remaining integrals, the PANEL subroutine had to be modified to incorporate E, and E^2 into the integrand. To accommodate this, NJOY99 was modified to produce:

$$\sigma_{tg} = \frac{\int_g E^P F(E)\sigma(E)\phi(E)dE}{\int_g \phi(E)dE} \qquad P = 0,1,2 \qquad \text{Eq. 5.29}$$

With this modification, LMG cross section production requires that NJOY99 be run three times, with P equal zero, one, or two. The resulting outputs from the three runs are then reassembled to calculate the new LMG group cross sections. Notice that the denominator is the same as is used in current NJOY99 runs and is inconsistent with the definition of LMG cross sections. The reason is that the denominator is factored out for the three runs.

However, once the new group cross sections are calculated, they are then rescaled using the new group fluxes, which are based on Eq 5.23.

Case 2 and 3 are similar, except for the difference in the group numbers. In these cases, the basis functions for group g overlap with basis functions of group g-1 and group g+1 respectively. As a result the total cross section from group g to g+1 or g to g-1 can be calculated as follows:

$$\Psi_k \sigma_{tgk} = \int_0^{\infty} \sigma_t(E)\Psi(E) f_g(E) f_k(E) dE \qquad \text{Eq. 5.30}$$

Where $k = g-1$ or $k = g+1$. When $k = g-1$, then by incorporating the linear equation for the hat functions, the following is obtained:

$$\Psi_{g-1}\sigma_{tg,g-1} = \Psi_g \sigma_{tg-1,g} = \int_{E_g}^{E_{g-1}} (m_{lg-1}E + b_{lg-1})(m_{rg-1}E + b_{rg-1})\sigma_t(E)\Psi(E)dE \quad \text{Eq. 5.31}$$

By expanding the above equation, the following is obtained:

$$\begin{aligned}\Psi_{g-1}\sigma_{tg,g-1} = \Psi_g \sigma_{tg-1,g} = m_{lg-1}m_{rg-1} &\int_{E_g}^{E_{g-1}} E^2\sigma_t(E)\Psi(E)dE \\ +m_{lg-1}b_{rg-1} &\int_{E_g}^{E_{g-1}} E\sigma_t(E)\Psi(E)dE \qquad \text{Eq. 5.32} \\ +m_{rg-1}b_{lg-1} &\int_{E_g}^{E_{g-1}} E\sigma_t(E)\Psi(E)dE + b_{lg-1}b_{rg-1}\int_{E_g}^{E_{g-1}} \sigma_t(E)\Psi(E)dE\end{aligned}$$

By similarly expanding the equation for $k = g+1$, the following is obtained:

$$\Psi_{g+1}\sigma_{tg,g+1} = \Psi_g\sigma_{tg+1,g} = m_{lg+1}m_{rg+1}\int_{E_{g+1}}^{E_g} E^2\sigma_t(E)\Psi(E)dE$$
$$+ m_{lg+1}b_{rg+1}\int_{E_{g+1}}^{E_g} E\sigma_t(E)\Psi(E)dE$$
$$+ m_{rg+1}b_{lg+1}\int_{E_{g+1}}^{E_g} E\sigma_t(E)\Psi(E)dE \qquad \text{Eq. 5.33}$$
$$+ b_{lg+1}b_{rg+1}\int_{E_{g+1}}^{E_g} \sigma_t(E)\Psi(E)dE$$

The constant coefficient, that is the slopes and intercepts of the basis functions, are pre-calculated. The other integrals are obtained from the three runs of modified NJOY99, as mentioned before.

5.2.2 Scattering Matrices

The calculation of the scattering matrix is much more complex. Since the hat functions overlap two of the conventional energy groups, the scattering from one group to another group in the new method requires the collection of terms from two source groups and two destination groups of the multigroup method, as shown in Figure 5.3.

The generalized multigroup method results in the following term for the scattering matrix:

$$\sigma_{skg} = \frac{\int_0^\infty f_k(E)dE \int_0^\infty f_g(E')\sigma_s(E' \to E)\Psi(E')dE'}{\Psi_g} \qquad \text{Eq. 5.34}$$

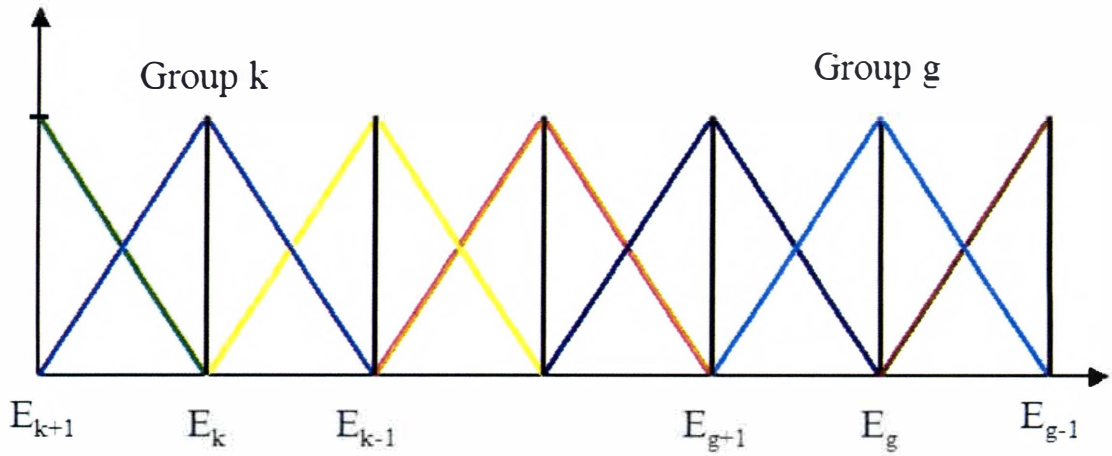


Figure 5.3: Hat basis functions utilized to calculate the scattering cross sections from group g to group k

Prior to normalizing the scattering terms with the group cross sections, the non-normalized scattering cross section has to be calculated. Hence, this cross section is:

$$\Psi_g \sigma_{skg} = \int_0^\infty f_k(E)dE \int_0^\infty f_g(E')\sigma_s(E' \to E)\Psi(E')dE' \qquad \text{Eq. 5.35}$$

where the non-normalized scattering cross section is defined to be:

$$\Psi_{g'} \sigma_{gg'} = \int_g F(E)\sigma(E)\phi(E)dE \qquad \text{Eq. 5.36}$$

and the feed function is defined to be:

$$F_k(E) = \int_k dE' f(E \to E') \qquad \text{Eq. 5.37}$$

Therefore, by incorporating the basis functions the resultant equations are:

$$\Psi_k \sigma_{gk} = \int_g f_g(E) F_k(E) \sigma(E) \phi(E) dE \qquad \text{Eq. 5.38}$$

and

$$F_k(E) = \int_k dE' f_k(E') f(E \rightarrow E') \qquad \text{Eq. 5.39}$$

By substituting the values of $f_g(E)$ and $f_k(E)$ the resultant equations are:

$$\begin{aligned} \Psi_k \sigma_{gk} = & \int_{E_{g-1}}^{E_{g-2}} (m_{lg-1} E + b_{g-1}) F_k(E) \sigma(E) \phi(E) dE \\ & + \int_{E_g}^{E_{g-1}} (m_{rg} E + b_{rg}) F_k(E) \sigma(E) \phi(E) dE \end{aligned} \qquad \text{Eq. 5.40}$$

The feed function in NJOY99 is then transformed into:

$$\begin{aligned} F_k(E) = & \int_{E_{k-1}}^{E_{k-2}} (m_{lk-1} E' + b_{lk-1}) f(E \rightarrow E') dE' \\ & + \int_{E_k}^{E_{k-1}} (m_{rk} E' + b_{rk}) f_k(E') f(E \rightarrow E') dE' \end{aligned} \qquad \text{Eq. 5.41}$$

By substituting the feed function integral into Eq. 1.36, then the scattering cross section becomes:

$$
\begin{aligned}
\Psi_k \sigma_{gk} = & \int_{E_{g-1}}^{E_{g-2}} (m_{lg-1}E + b_{g-1}) \left(\int_{E_{k-1}}^{E_{k-2}} (m_{lk-1}E' + b_{lk-1}) f(E \to E')dE' \right) \sigma(E)\phi(E)dE \\
& + \int_{E_{g-1}}^{E_{g-2}} (m_{lg-1}E + b_{g-1}) \left(\int_{E_k}^{E_{k-1}} (m_{rk}E' + b_{rk}) f_k(E') f(E \to E')dE' \right) \sigma(E)\phi(E)dE \\
& + \int_{E_g}^{E_{g-1}} (m_{rg}E + b_{rg}) \left(\int_{E_{k-1}}^{E_{k-2}} (m_{lk-1}E' + b_{lk-1}) f(E \to E')dE' \right) \sigma(E)\phi(E)dE \\
& + \int_{E_g}^{E_{g-1}} (m_{rg}E + b_{rg}) \left(\int_{E_k}^{E_{k-1}} (m_{rk}E' + b_{rk}) f_k(E') f(E \to E')dE' \right) \sigma(E)\phi(E)dE
\end{aligned}
$$

Eq. 5.42

Expanding the above equation results in:

$$
\begin{aligned}
\Psi_k \sigma_{gk} = & m_{lg-1}m_{lk-1} \int_{E_{g-1}}^{E_{g-2}} E \left(\int_{E_{k-1}}^{E_{k-2}} E' f(E \to E')dE' \right) \sigma(E)\phi(E)dE + m_{lg-1}b_{lk-1} \int_{E_{g-1}}^{E_{g-2}} E \left(\int_{E_{k-1}}^{E_{k-2}} f(E \to E')dE' \right) \sigma(E)\phi(E)dE \\
& + m_{lk-1}b_{g-1} \int_{E_{g-1}}^{E_{g-2}} E \left(\int_{E_{k-1}}^{E_{k-2}} E' f(E \to E')dE' \right) \sigma(E)\phi(E)dE + b_{g-1}b_{lk-1} \int_{E_{g-1}}^{E_{g-2}} \left(\int_{E_{k-1}}^{E_{k-2}} f(E \to E')dE' \right) \sigma(E)\phi(E)dE \\
& + m_{lg-1}m_{rk} \int_{E_{g-1}}^{E_{g-2}} E \left(\int_{E_k}^{E_{k-1}} (E' f(E \to E')dE' \right) \sigma(E)\phi(E)dE + m_{lg-1}b_{rk} \int_{E_{g-1}}^{E_{g-2}} E \left(\int_{E_k}^{E_{k-1}} f(E \to E')dE' \right) \sigma(E)\phi(E)dE \\
& + m_{rk}b_{g-1} \int_{E_{g-1}}^{E_{g-2}} \left(\int_{E_k}^{E_{k-1}} E' f(E \to E')dE' \right) \sigma(E)\phi(E)dE + b_{g-1}b_{rk} \int_{E_{g-1}}^{E_{g-2}} \left(\int_{E_k}^{E_{k-1}} f(E \to E')dE' \right) \sigma(E)\phi(E)dE \\
& + m_{rg}m_{lk-1} \int_{E_g}^{E_{g-1}} E \left(\int_{E_{k-1}}^{E_{k-2}} E' f(E \to E')dE' \right) \sigma(E)\phi(E)dE + m_{rg}b_{lk-1} \int_{E_g}^{E_{g-1}} E \left(\int_{E_{k-1}}^{E_{k-2}} f(E \to E')dE' \right) \sigma(E)\phi(E)dE \\
& + m_{lk-1}b_{rg} \int_{E_g}^{E_{g-1}} \left(\int_{E_{k-1}}^{E_{k-2}} E' f(E \to E')dE' \right) \sigma(E)\phi(E)dE + b_{rg}b_{lk-1} \int_{E_g}^{E_{g-1}} \left(\int_{E_{k-1}}^{E_{k-2}} f(E \to E')dE' \right) \sigma(E)\phi(E)dE \\
& + m_{rg}m_{rk} \int_{E_g}^{E_{g-1}} E \left(\int_{E_k}^{E_{k-1}} E' f(E \to E')dE' \right) \sigma(E)\phi(E)dE + m_{rg}b_{rk} \int_{E_g}^{E_{g-1}} E \left(\int_{E_k}^{E_{k-1}} f_k(E') f(E \to E')dE' \right) \sigma(E)\phi(E)dE \\
& + m_{rk}b_{rg} \int_{E_g}^{E_{g-1}} \left(\int_{E_k}^{E_{k-1}} E' f(E \to E')dE' \right) \sigma(E)\phi(E)dE + b_{rg}b_{rk} \int_{E_g}^{E_{g-1}} \left(\int_{E_k}^{E_{k-1}} f(E \to E')dE' \right) \sigma(E)\phi(E)dE
\end{aligned}
$$

Eq. 5.43

The above complex expression can be assembled by evaluating the following integrals:

$$\int_{E_g}^{E_g}\left(\int_{E_k}^{E_{k-1}} f(E\rightarrow E')dE'\right)\sigma(E)\phi(E)dE,$$

$$\int_{E_g}^{E_{g-1}}\left(\int_{E_k}^{E_{k-1}} E'f(E\rightarrow E')dE'\right)\sigma(E)\phi(E)dE,$$

$$\int_{E_g}^{E_{g-1}} E\left(\int_{E_k}^{E_{k-1}} f(E\rightarrow E')dE'\right)\sigma(E)\phi(E)dE,$$

and

$$\int_{E_g}^{E_{g-1}} E\left(\int_{E_k}^{E_{k-1}} E'f(E\rightarrow E')dE'\right)\sigma(E)\phi(E)dE$$

Since the current NJOY99 cannot produce these integrals, it had to be modified using the following alterations. The resulting modified version of NJOY99 is run four times to produce each of the above integrals.

$$\sigma_{gg'}=\frac{\int_{E_g}^{E_{g-1}} E^P F_{g'}\sigma(E)\phi(E)dE}{\int_{g}\phi(E)dE} \qquad P=0,1 \qquad \text{Eq. 5.44}$$

$$F_{g'}=\int_{E_{g'}}^{E_{g'-1}} E^L f(E\rightarrow E')dE' \qquad L=0,1 \qquad \text{Eq. 5.45}$$

From the results of the four runs, the scattering matrix for the hat cross sections is assembled from these components.

5.2.3 Fission Spectrum

The neutron fission spectrum is not strongly dependent on the energy of the neutron that caused the fission except at very high energy. Most codes use a common fission spectrum that is not dependent on the energy of the fission inducing neutron. In such a case, the fission cross section can be calculated as follows:

$$\Psi_{g'}\nu\sigma_{fg'} = \int_0^{\infty} \nu\sigma_f(E')\Psi(E')f_{g'}(E')dE' \qquad \text{Eq. 5.46}$$

By substituting $f_{g'}(E')$ into the equation, the following is obtained:

$$\begin{aligned}\Psi_{g'}\nu\sigma_{fg'} = &\int_{E_{g'-1}}^{E_{g'-2}} \nu\sigma_f(E')\Psi(E')(m_{lg'-1}E + b_{lg'-1})dE' \\ &+ \int_{E_{g'}}^{E_{g'-1}} \nu\sigma_f(E')\Psi(E')(m_{rg'}E' + b_{rg'})dE'\end{aligned} \qquad \text{Eq. 5.47}$$

By expanding the above equation, the following is obtained:

$$\begin{aligned}\Psi_{g'}\nu\sigma_{fg'} = &m_{lg'-1}\int_{E_{g'-1}}^{E_{g'-2}} \nu\Sigma_f(E')\Psi(E')E'dE' + b_{lg'-1}\int_{E_{g'-1}}^{E_{g'-2}} \nu\Sigma_f(E')\Psi(E')dE' \\ &+m_{rg'}\int_{E_{g'}}^{E_{g'-1}} \nu\Sigma_f(E')\Psi(E')E'dE' + b_{rg}\int_{E_{g'}}^{E_{g'-1}} \nu\Sigma_f(E')\Psi(E')dE'\end{aligned} \qquad \text{Eq. 5.48}$$

Hence, the following terms need to be calculated:

$$\int_{E_g}^{E_{g-1}} \nu\Sigma_f(E')\Psi(E')E'dE'$$

and

$$\int_{E_g}^{E_{g-1}} \nu\Sigma_f(E')\Psi(E')dE'.$$

To compute the fractional contribution of group g of fission neutrons, the following integral has to be evaluated:

$$\chi_k = \int_0^{\infty} f_k(E)\chi(E)dE \qquad \text{Eq. 5.49}$$

By substituting $f_k(E)$ into the equation, the following is obtained:

$$\chi_k = \int_{E_{k-1}}^{E_{k-2}} (m_{lk-1}E + b_{lk-1})\chi(E)dE + \int_{E_k}^{E_{k-1}} (m_{rk}E + b_{rk})\chi(E)dE \qquad \text{Eq. 5.50}$$

To obtain these cross sections, the following terms have to be calculated: $\int_{E_k}^{E_{k-1}} E\chi(E)dE$ and $\int_{E_k}^{E_{k-1}} \chi(E)dE$. Again, NJOY99 was modified to produce these integrals in two separate runs, with the resulting LMG value of χ_k constituted from the results of the two runs.

In this manner, all of the new terms needed for the new cross sections are calculated using NJOY99. Hence, the objective of generating the new cross section using an existing industry code has been met.

5.3 Data Preparation for Gamma Rays

NJOY99 has a GAMNIR module that is similar to GROUPR, but produces gamma ray cross sections. For gamma problems, there are two average cross section processes that are needed from the ENDF-B files. The first process is the total cross section, MT501. The second process is the photoelectric effect, MT522. This process acts as the absorption cross section. For scattering processes there are three processes that are available in the ENDF-B files. They are coherent scattering, incoherent scattering, and pair production, MT502, MT504 and MT516, respectively. Besides specifying the processes for NJOY99 to evaluate, the Legendre order of the scattering matrices and the assumed spectrum shape have to be specified. For gamma absorption problems in an infinite medium, because of lack of directional dependence for gamma ray transport, scattering of Legendre order of zero is what is needed. As for spectrum specifications, there are three options: user specified, constant, or 1/E with roll-off. The last option is the best representative spectrum for a general gamma problem. The 1/E region accounts for gamma build up due to energy loss. The roll-off represents the flux drop due to the photoelectric effect, where most gamma absorption takes place. The spectrum is shown in Figure 5.4.

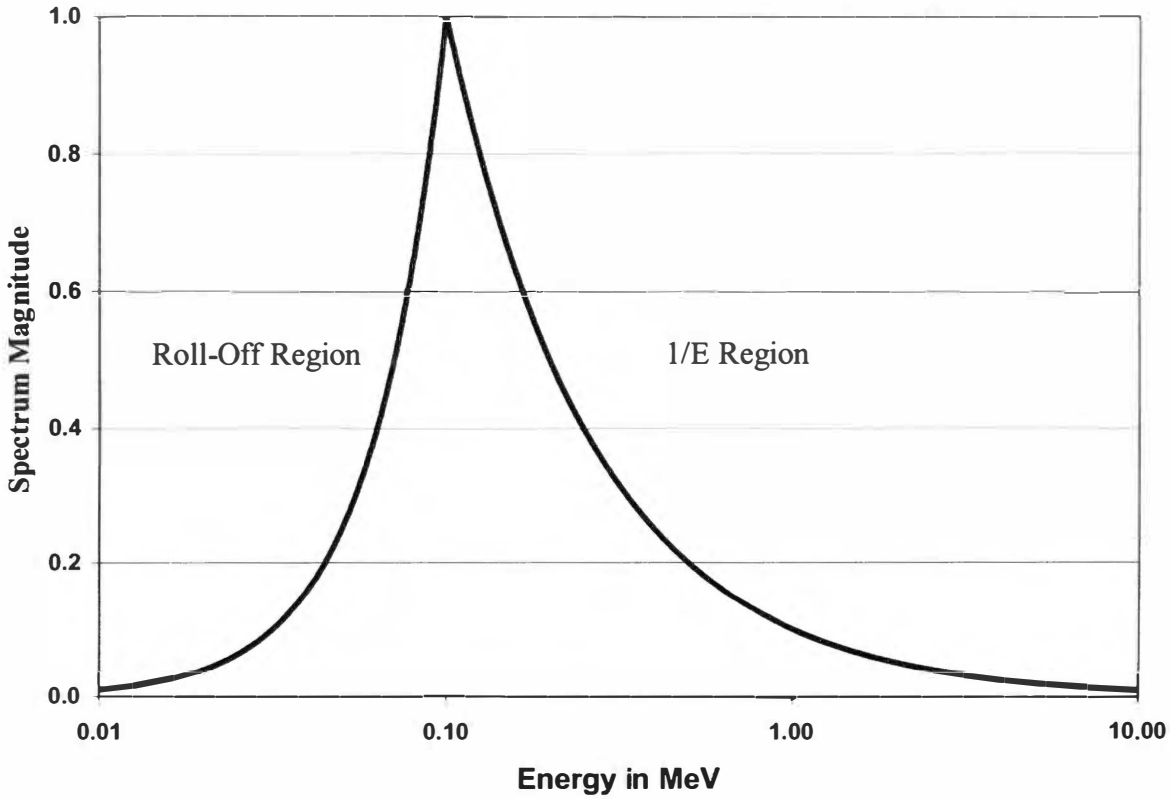


Figure 5.4: Assumed spectrum specified by NJOY99. It consist of 1/E and Roll-Off region to account for photoelectric effect

Once the data requirement is specified, a batch file is executed to run NJOY99 six times to produce the required data. As mentioned above, NJOY99 was modified to produce group cross sections with a hat function as the basis function. However, once the batch file is run six times, the data are stored in a flexible manner to allow the assembling of the data into either multigroup or LMG cross sections. The flexibility in assembling the data becomes important later when the hybrid form of the cross section is used to resolve the problem of negative fluxes.

5.4 Data Assembly

By running NJOY99 six times, all the data needed to construct the new cross sections are produced. NJOY99 produces the cross sections in its own specific format. It has a number of modules that transform the data to other formats that make the data usable in a number of transport codes. For this dissertation, an Oak Ridge National Lab code, SMILER, was utilized to transform the data to AMPX master library format. This data format is used by the SCALE system that is also written and maintained by ORNL. The advantage of using AMPX is that the format of data storage is very well documented. This makes the process of reading the cross sections and reassembling them using the appropriate coefficients much easier.

Once the data from the six runs of NJOY have been produced, the data are transformed into AMPX master library format. A code was written to produce the slopes and the intercepts for all of the hat functions. At this point, all the data needed to calculate the new cross sections are available. The code, READWRK, reads all the average cross sections from the six working libraries produced from the six runs of NJOY. Then, factoring out the group fluxes by which they were normalized in NJOY99, renormalizes the data. A second loop then goes through the data and assembles the LMG group cross sections by reading the appropriate NJOY99 output and multiplying by the right slope or intercept. Next, dividing the cross sections by their group fluxes renormalizes the data. The scattering matrices go through the same process of factoring out the group fluxes and assembling the new cross sections using the appropriate coefficients. Once the new cross sections are calculated, they are then stored back in a new file in AMPX working library

format. This allows the data to be used in the SCALE package for solving 1-D pin-cell problems, shielding or criticality problems.

6. Sample Problems

The primary objective of the Linear Multigroup Method is to reduce the number of energy groups needed to obtain a given accuracy. To demonstrate the superiority of the new method over the conventional method, two problems are benchmarked versus the conventional multigroup method. The criterion for comparing the two methods is the number of energy groups versus error. The LMG is applicable to both neutrons and gamma particles. However, the less detailed treatment of the assumed spectrum in gamma shielding allows us a clearer comparison of how the new method adjusts the assumed spectrum to fit the true one, so gamma ray problems were chosen.

In this chapter two sample problems that deal with infinite media are presented. The first problem is for an oxygen infinite medium with a continuous fission gamma spectrum as a source. The second problem is also an infinite oxygen medium, but with a monoenergetic source. Oxygen is chosen as the medium for gamma spectrum calculation because it is a relatively moderate absorber in comparison to lead. This choice also allows a spectrum to develop as the gamma rays lose energy due to scattering. A number of issues, such as negative fluxes around the pair production energy range and around monoenergetic source energies will be discussed. Strategies to eliminate these negative fluxes will be outlined.

6.1 First Sample Problem: Prompt Fission Gamma Source

6.1.1 Source Description

When atoms undergo fission, they produce a significant number of gamma rays within the first 60 ns after fission. These gamma rays are referred to as prompt fission gamma rays. Other photons are produced later on due to the decays of fission products. These photons are of great importance in reactor shielding and gamma heating. For these reasons, they are studied extensively. Prompt fission gamma photons of ^{235}U due to thermal-neutron-induced fission have been the focus of most studies [34]. The measured empirical fit of the fission prompt gamma spectrum for energies between 0.1 MeV and 10.5 MeV is given by [35] as:

$$N_{P\gamma}(E) = \begin{cases} 6.6 & 0.1 < E < 0.6\ MeV \\ 20.2e^{-1.78E} & 0.6 < E < 1.5\ MeV \\ 7.2e^{-1.09E} & 1.5 < E < 10.5\ MeV \end{cases}$$

where E is in MeV and $N_{P\gamma}$ (E) is the number of photons/MeV/fission. Work to determine the prompt fission photon spectra of other fission nuclides has been minimal. However, studies on some fission nuclides have indicated little variation in their spectrum from the one for ^{235}U. For this reason, the ^{235}U data are used for shielding purposes for other fission isotopes. The gamma spectrum fit is shown in Figure 6.1.

For this benchmark, this spectrum is utilized. The calculation of a group source is a simple integration of the spectrum over each group's energy range. Hence, the gamma

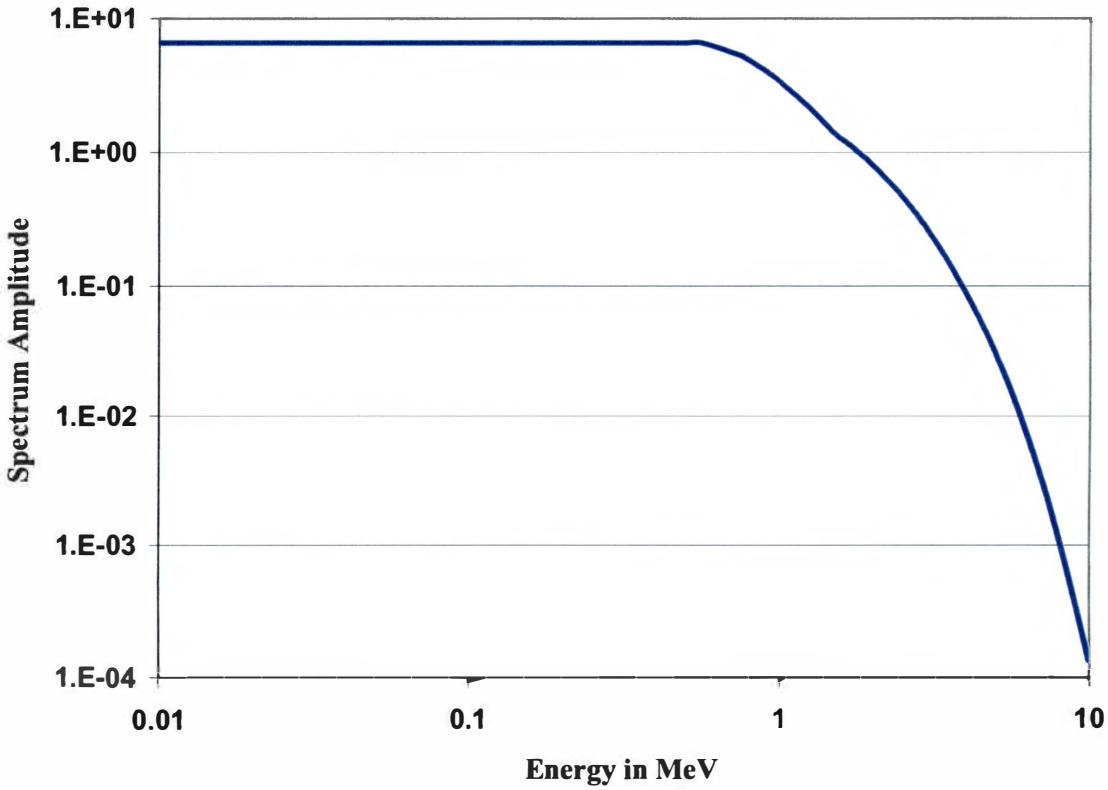


Figure 6.1: Prompt fission gamma photon spectrum for U-235

source for group g is calculated as follows:

$$S_g = \int_{E_g}^{E_{g-1}} N_{p\gamma}(E)dE \qquad \text{Eq. 6.1}$$

When the hat functions are used as the basis functions to solve the spectrum calculation problem, then the definition of a group also changes. Therefore, to calculate the gamma source for group g, the following integral is used:

$$S_g = \int_{E_g}^{E_{g-1}} f_g(E) N_{P\gamma}(E) dE \qquad \text{Eq. 6.2}$$

where, as before,

$$f_g(E) = \begin{cases} \dfrac{E - E_{g+1}}{E_g - E_{g+1}}, & E_{g+1} < E < E_g \\ \dfrac{E_{g-1} - E}{E_{g-1} - E_g}, & E_g < E < E_{g-1} \end{cases} \qquad \text{Eq. 6.3}$$

6.1.2 Comparison of Results

To compare the performance of the two methods to each other, some measure of performance is required. For these benchmarks, a good measure of the performance of these methods is the number of energy groups required for a given error in the absorption of gamma rays over some energy sub domain. A rapid convergence to the desired accuracy would confirm the superiority of one method over the other. For these benchmarks, the energy domain is subdivided into four logarithmically equivalent intervals. The absorption problem is then solved using eight energy groups that overlap the four intervals. The absorption rate in each of the four original groups is then calculated using the calculated group fluxes and the corresponding absorption cross sections. The problem is then solved again using twice the number of energy groups, with the absorption in the original four groups again calculated. This process is repeated until the number of energy groups reaches 256 groups. At that point, the solution is found to have reached its asymptotic limit. Furthermore, NJOY99 can only calculate cross

sections for 399 energy groups, which makes solving the problem with 512 groups impossible.

Calculating the reaction rate over a subdomain of energy requires that absorption coefficients be available for both sides of the hat function for each LMG group. These were saved from the NJOY99 data processing step described in the previous chapter.

6.1.3 Issues of Pair Production and Discrete Energy Sources

Pair production is a process that takes place when a photon is completely absorbed and a positron-electron pair appears. The reaction takes place when the original gamma ray interacts with the strong electric field around the nucleus of an atom. The positron very quickly collides with an electron, converting the rest mass of each into two 511 keV gamma rays. The threshold gamma ray energy for this process is 1.022 MeV. The overall result of this process is equivalent to a high energy scattering event, where a gamma ray with energy greater than 1.022 MeV is absorbed and two 511 keV gamma rays emerge. The resulting gamma ray spectrum will have a spike in its shape at exactly 511 keV.

This discrete jump in magnitude cannot be represented properly with the hat functions. These functions assume a smooth and continuous change in the spectrum as a function of energy, and cannot represent the sudden rise in the spectrum's magnitude due to a discontinuity in the spectrum. In response to this discontinuity, the calculated LMG fluxes of the two adjacent groups go negative for the spectrum to properly follow this sudden rise in flux at the pair production energy. This phenomenon is shown in Figure

6.2. This figure represents the spectrum due to a discrete energy source at 10 MeV. It is apparent that fluxes are negative on both sides of the pair production group.

From Figure 6.2, it is also observed that the spectrum becomes negative adjacent to an energy discontinuity in the source. The same reason for the cause of the negative fluxes adjacent to pair production energy groups is also valid for the discrete energy sources. The hat functions cannot represent the discontinuity accurately. Therefore, there is a fundamental discrepancy between the assumed spectrum behavior that motivated the introduction of the hat functions and the real spectrum.

The mitigation of this problem for discrete energy sources will be discussed in a later section. As for the pair production problem, representing the energy group that contains 511 keV with a conventional multigroup basis function can ameliorate it. This allows the spectrum to represent the higher 511 keV flux without forcing the spectra of other groups to become negative. Therefore, a discontinuous spectrum for that energy group gives a more accurate representation than the hat functions.

6.1.4 Results for Fission Spectrum Gamma Source

To compare the efficiency of the conventional and LMG method, the absorption rate in four original energy groups is calculated using conventional multigroup methods, hat functions, and hat functions with a conventional multigroup for the group that contains 511 keV.

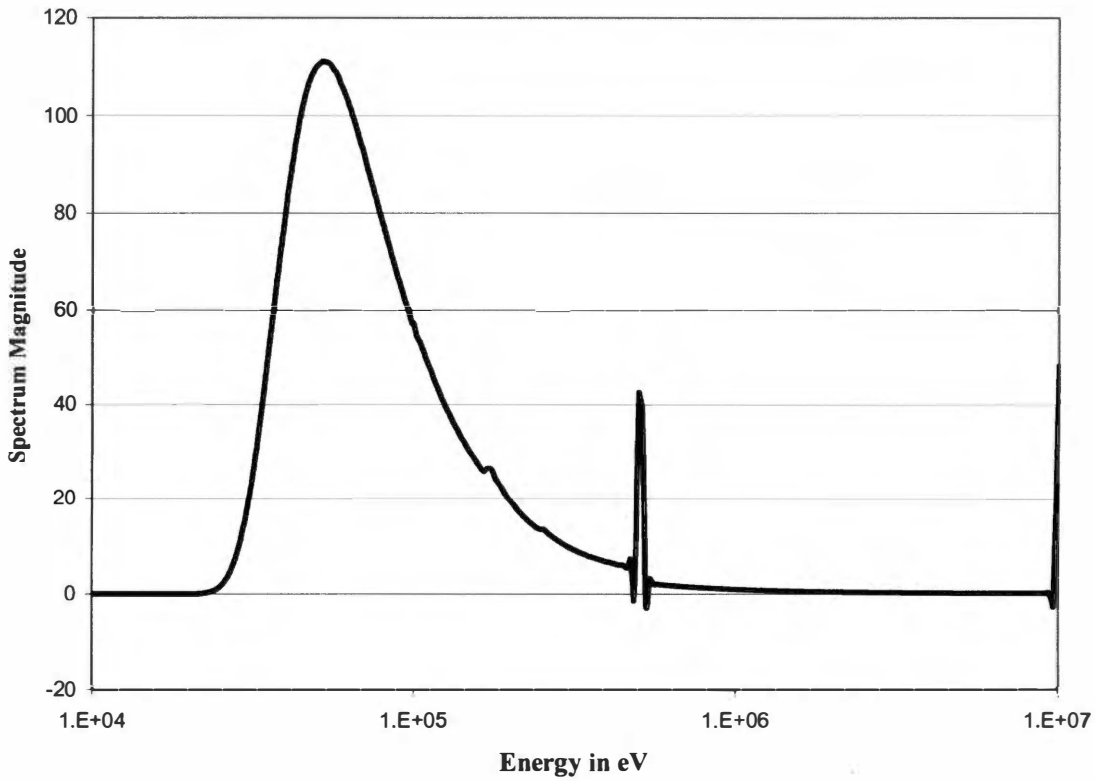


Figure 6.2: Spectrum becomes negative when hat functions are used due to discontinuities caused by pair production gammas and discrete energy sources.

The results for absorption in the third and fourth original energy groups are tabulated in Table 6.1 and Table 6.2 respectively and plotted in Figure 6.3 and Figure 6.4 respectively. This hybrid approach in generating the cross sections gives the best representation of the spectrum. For this reason, the results from the 256 energy groups will be used as the reference solution to which all methods are compared to it. It is apparent from the data that this hybrid approach has the fastest convergence rate and most stable solution. These two energy ranges were selected because most of the gamma absorption takes place in them. This is primarily due to the strength of the photoelectric

Table 6.1: Absorption rate and the % error in the third energy group as function of number of energy groups

Number of Energy Groups	Multigroup Method	Error in %	Hat Basis Functions	Error in %	Hat Functions with MG Pair Production	Error in %
8	3.74740E-01	5.42E+01	3.92359E-01	6.14E+01	3.92808E-01	6.16E+01
16	2.95600E-01	2.16E+01	2.64450E-01	8.80E+00	2.64476E-01	8.81E+00
32	2.51780E-01	3.59E+00	2.43329E-01	1.10E-01	2.43410E-01	1.43E-01
64	2.45420E-01	9.70E-01	2.43030E-01	1.29E-02	2.43129E-01	2.76E-02
128	2.43580E-01	2.13E-01	2.42983E-01	3.22E-02	2.43087E-01	1.05E-02
256	2.43090E-01	1.17E-02	2.42954E-01	4.43E-02	2.43062E-01	-

Table 6.2: Absorption rate and % error in the fourth energy group as function of number of energy groups

Number of Energy Groups	Multigroup method	Error in %	Hat Basis functions	Error in %	Hat functions with MG Pair Production	Error in %
8	6.30650E-01	1.71E+01	6.10782E-01	1.97E+01	6.11396E-01	1.97E+01
16	7.08500E-01	6.89E+00	7.39640E-01	2.80E+00	7.39655E-01	2.80E+00
32	7.51990E-01	1.18E+00	7.60387E-01	7.34E-02	7.60710E-01	3.08E-02
64	7.58270E-01	3.51E-01	7.60634E-01	4.08E-02	7.61019E-01	9.72E-03
128	7.60090E-01	1.12E-01	7.60572E-01	4.90E-02	7.60993E-01	6.37E-03
256	7.60580E-01	4.79E-02	7.60505E-01	5.77E-02	7.60945E-01	-

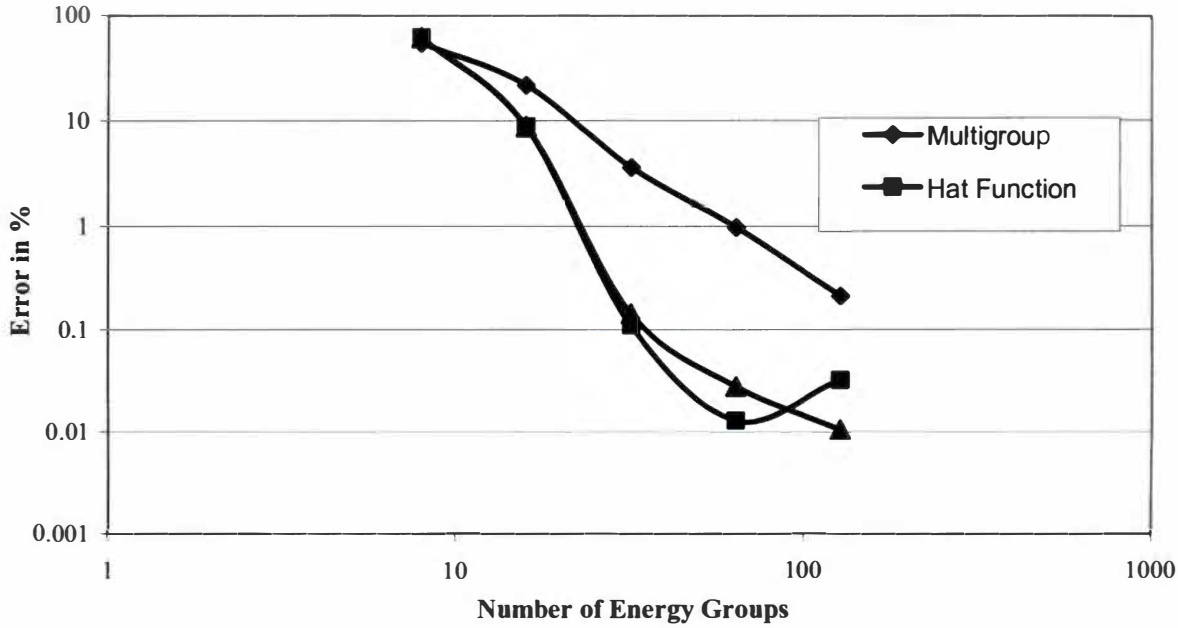


Figure 6.3: The error in absorption rate in the third energy group as function of number of energy groups.

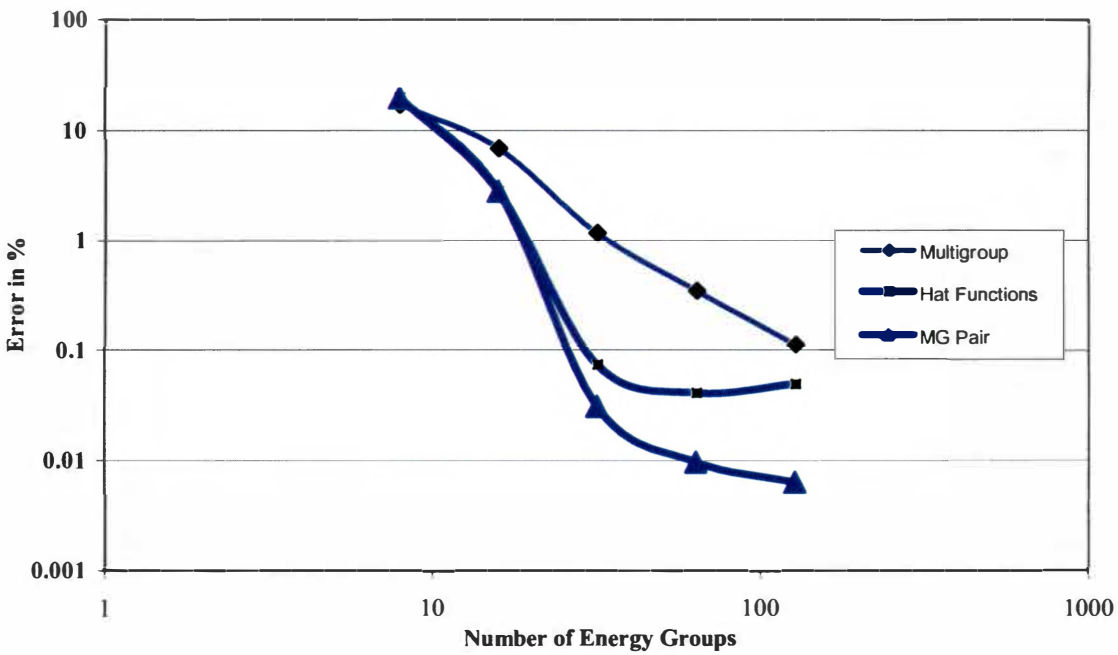


Figure 6.4: Error in absorption rate in the fourth energy group as function of number of energy groups

absorption in the low energy range. The absorption in each energy range is tabulated as a function of number of energy groups. The error for each energy range is also tabulated.

6.1.5 Negative Fluxes Due to Spectrum Mismatch

In a previous section, it is noted that the fluxes can become negative due to discontinuities caused by pair production or discontinuous energy sources. A third reason is due to severe mismatch between the assumed spectrum and the actual one. This issue manifests itself in the energy range below the pair production energy. Below that energy, the gamma spectrum rises rapidly. This rise in the spectrum is not represented well in the assumed spectrum. Since a linear adjustment of the spectrum is carried within these energy groups, the best linear fit of the calculated LMG spectrum causes the spectrum in some energy groups to become negative, as shown in Figure 6.5.

To overcome this problem, a minor adjustment in the energy group structure is carried out. By bringing the energy boundaries closer to the peak of the spectrum, the linear fit of the spectrum does not need to become negative. This is due to the fact that the linear fit of the spectrum is over a smaller interval. Interestingly, by adjusting the energy boundaries from their equal logarithmic intervals, the magnitude of the error, when few groups are used, is reduced. This is because of the improved representation of the spectrum by using more appropriate energy boundaries. The original and adjusted boundaries are listed in Table 6.3.

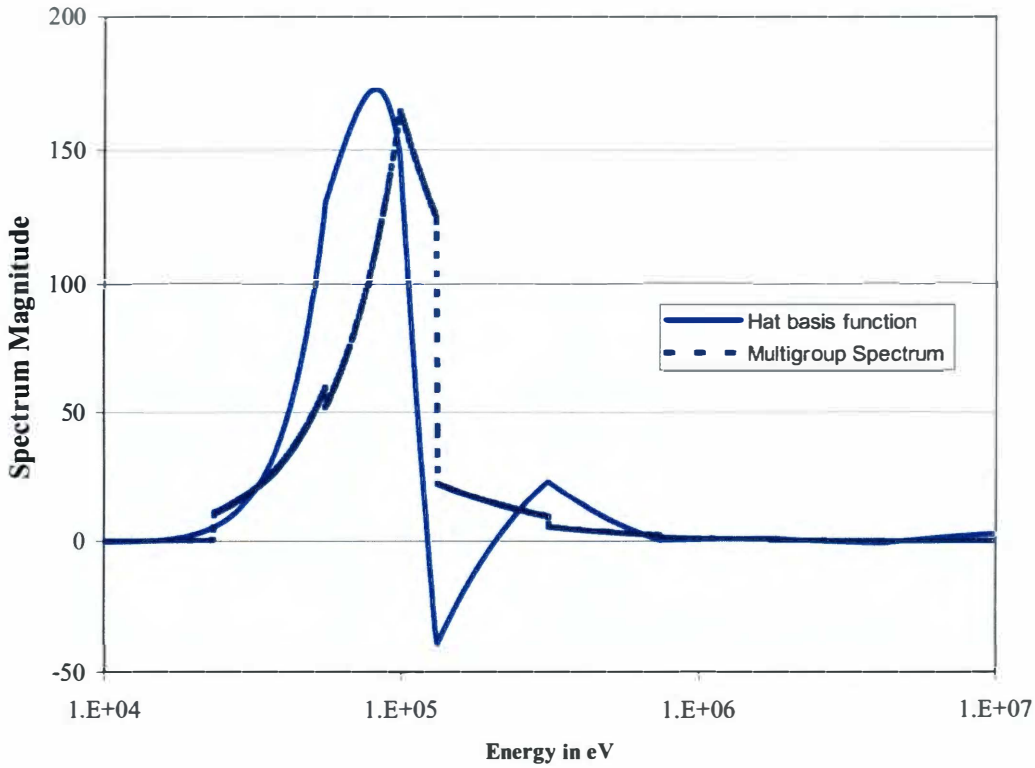


Figure 6.5: Negative spectrum due to mismatch between assumed and actual spectrum

Table 6.3: The energy boundaries for eight groups for equal logarithmic interval and adjusted intervals

Group Number	Equal Logarithmic Intervals	Non-equal Logarithmic Intervals
0	10000000	10000000
1	4216965	4216965
2	1778279	1000000
3	749894	650000
4	316227	480000
5	133352	95000
6	56234	50000
7	23713	27000
8	10000	10000

6.1.6 Results with Adjusted Energy Boundaries

After adjusting the energy group boundaries, which resulted in four different intervals than the original four, the absorption in the new energy groups was calculated. The results for absorption in the third and fourth energy ranges are listed in Table 6.4 and Table 6.5, respectively. The errors in the absorption calculation, for both of these groups, as a function of the number of energy groups, are plotted in Figure 6.6 and Figure 6.7. Figure 6.8 shows how the spectrum is no longer negative in the energy range below the pair production energy. Figure 6.9 shows the calculated spectrum for 256 energy groups using the LMG/MG method.

6.1.7 Summary of Results for Fission Gamma Source

Solving the gamma problem using hat basis functions has demonstrated the superiority of the linear multigroup method. By allowing the spectrum to adapt linearly and be continuous at the energy boundaries, the solution has converged at a faster rate than using the conventional method. For the original fourth energy group, the error in the prediction of gamma absorption is 6.89% for 16 energy groups. However, using the linear multigroup method for the same number of groups, the error is 2.80%. This is a reduction of more than a factor of two in the error. When 32 groups are used, the multigroup method error is reduced to 1.18%, while the linear multigroup method is reduced to 3.02E-2%. This is a reduction in error by a factor of 39, which is a significant reduction in error. When 128 energy groups are used, the error for the multigroup method is 1.12E-1% and it is 6.37E-3% for the linear multigroup method. Even when a large number of

Table 6.4: Absorption in third original energy group for adjusted energy boundaries

Number of Energy Groups	Multigroup Method	Error in %	Hat Basis Functions	Error in %	Hat Functions with MG Pair Production	Error in %
8	4.77605E-01	3.41E+01	4.37025E-01	2.27E+01	4.37319E-01	2.27E+01
16	3.95683E-01	1.11E+01	3.64918E-01	2.42E+00	3.64918E-01	2.42E+00
32	3.64438E-01	2.29E+00	3.56713E-01	1.22E-01	3.56712E-01	1.22E-01
64	3.58357E-01	5.83E-01	3.56353E-01	2.09E-02	3.56356E-01	2.16E-02
128	3.56666E-01	1.09E-01	3.56306E-01	7.63E-03	3.56305E-01	7.35E-03
256	3.56248E-01	8.67E-03	3.56278E-01	8.42E-05	3.56279E-01	-

Table 6.5: Absorption in the fourth original energy group for adjusted energy boundaries

Number of Energy Groups	Multigroup Method	Error in %	Hat Basis Functions	Error in %	Hat Functions with MG Pair Production	Error in %
8	5.3033780E-01	1.82E+01	5.67563E-01	1.24E+01	5.67573E-01	1.24E+01
16	6.0928500E-01	5.98E+00	6.39596E-01	1.30E+00	6.39593E-01	1.30E+00
32	6.3975730E-01	1.28E+00	6.47722E-01	4.62E-02	6.47721E-01	4.64E-02
64	6.4566140E-01	3.64E-01	6.48039E-01	2.72E-03	6.48043E-01	3.35E-03
128	6.4730790E-01	1.10E-01	6.48047E-01	3.94E-03	6.48045E-01	3.66E-03
256	6.4771520E-01	4.73E-02	6.48021E-01	9.26E-05	6.48022E-01	-

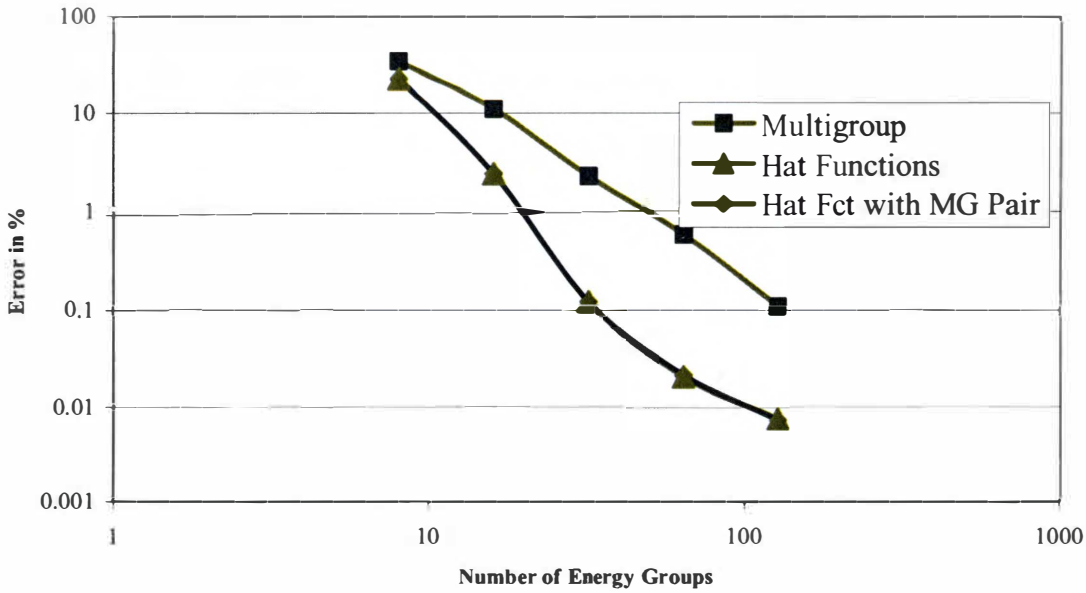


Figure 6.6: Absorption calculation error in third energy group for adjusted boundaries

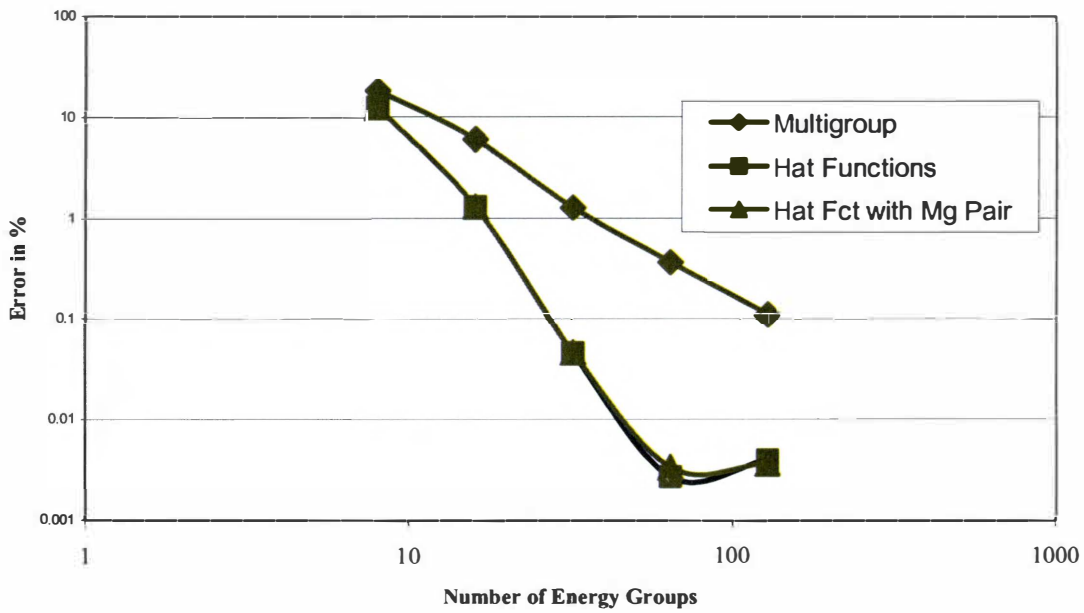


Figure 6.7: Absorption calculation error in fourth energy by group for adjusted boundaries

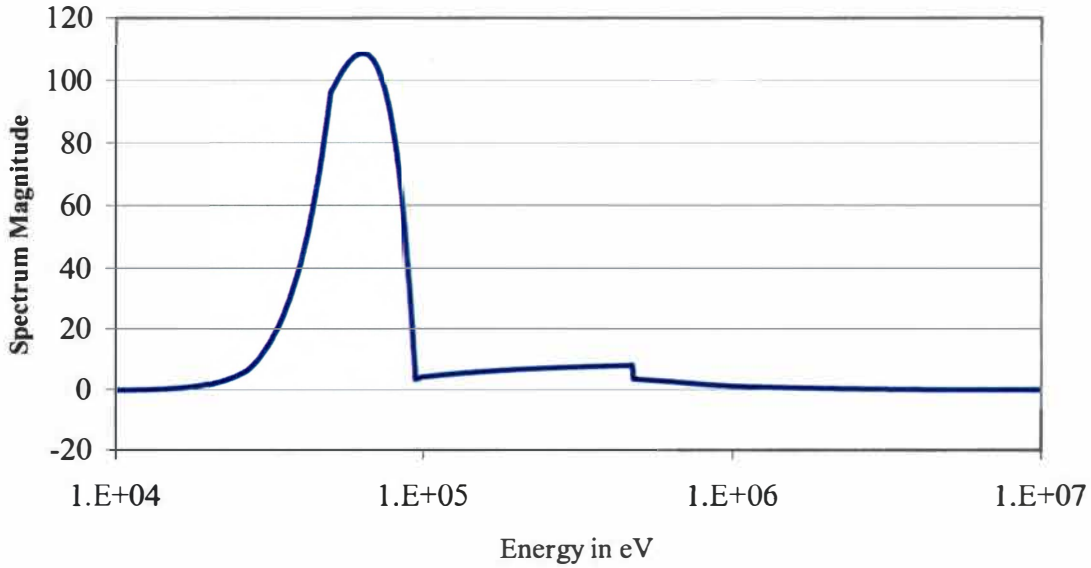


Figure 6.8: Elimination of negative of fluxes due to mismatch between assumed and actual spectrum by adjusting group energy boundaries. The spectrum is for eight energy groups solution using hybrid LMG/MG method

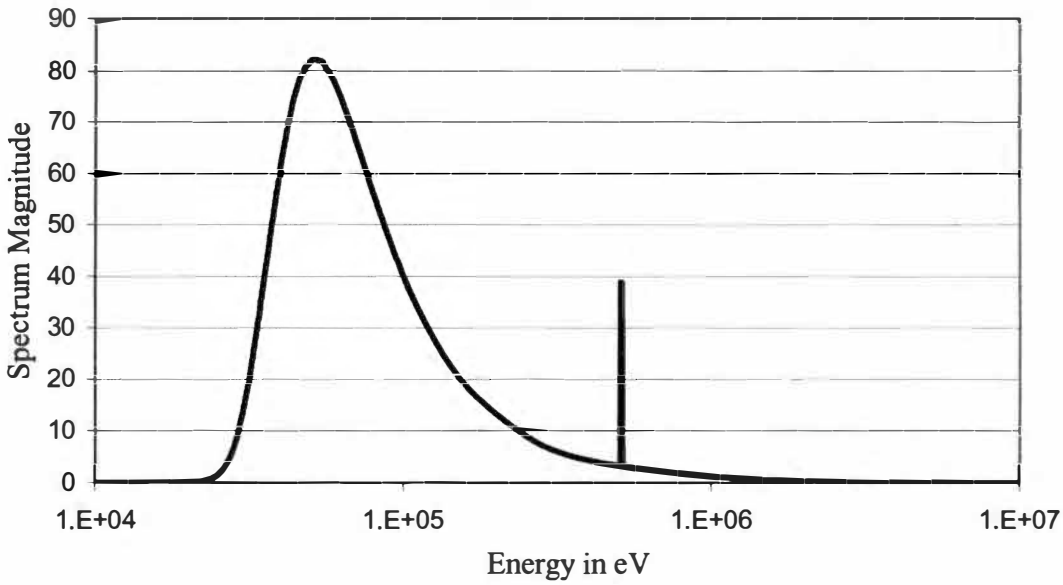


Figure 6.9: Calculated spectrum using 256 energy groups and hat/MG basis functions

energy groups are utilized, the linear multigroup method is still 17 times more efficient than the conventional multigroup method.

When fewer energy groups are used, the spectrum of the solution calculated by the linear multigroup method can go negative because of the mismatch between the assumed spectrum and the actual spectrum. In such cases, the solution can force the calculated spectrum to go negative for the best fit. Modifying the energy group boundaries to be slightly different than equal logarithmic segments solves this problem. As shown in Figure 6.8, using the modified energy group boundaries, the spectrum no longer goes negative. Furthermore, the predicted error in gamma absorption for both methods is reduced because of the better match of the spectrum in these intervals. For the multigroup method, the error is reduced from 6.89% to 5.98% for sixteen energy groups. For the linear multigroup method, the error is reduced from 2.80% to 1.30%. Overall, using the new energy boundaries, the error in absorption prediction for the multigroup method is 5.98% for 16 energy groups, compared to 1.3% for the linear multigroup method. This is a reduction by a factor of 4.5 times. When 128 groups are used, the error is reduced from 1.10E-1 for the MG to 3.66E-3 for LMG. This is a reduction by a factor of 30. Using this approach, the possibility of obtaining negative fluxes has been reduced.

Beyond demonstrating the superiority of the linear group method in converging to the true solution, this problem also demonstrates its ability to adapt the spectrum shape. This is shown in Figure 6.10. In this figure, the spectrum from the multigroup method shows the discontinuity of the spectrum and its rigidity. The spectrum shape within each group has not changed from the original assumed spectrum. Only the magnitude of the spectrum

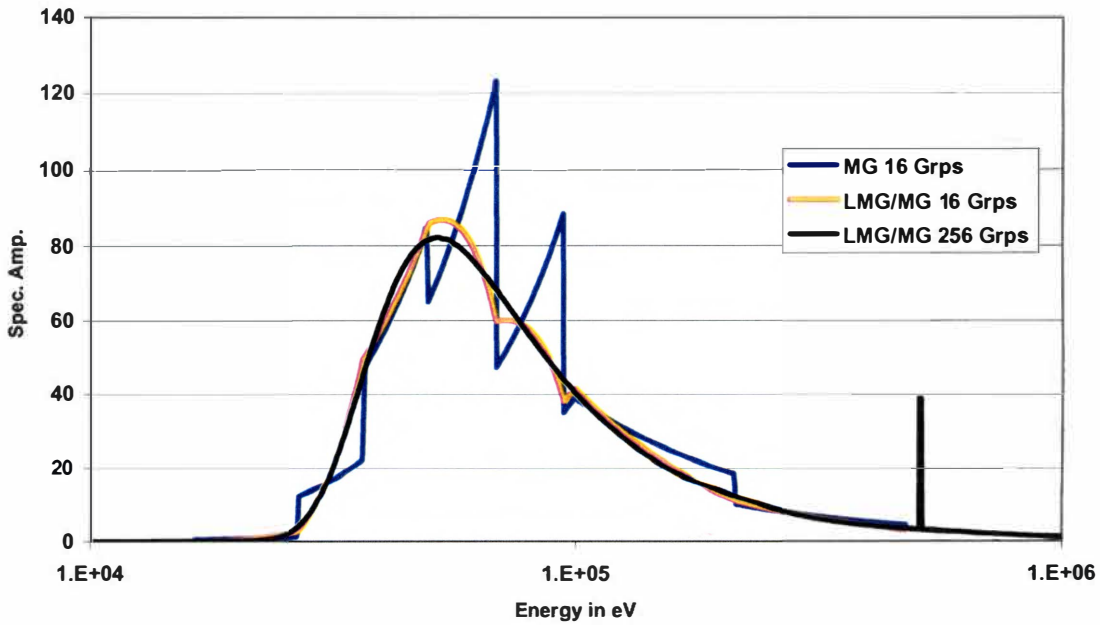


Figure 6.10: Comparison of spectrum shapes for MG and LMG/MG methods

within an energy group has changed. On the other hand, the linear multigroup spectrum is continuous at the energy boundaries, and the spectrum shape has changed to match better the true solution. This spectrum is a more realistic representation of the true spectrum and significantly different than the assumed spectrum.

6.2 Second Sample Problem: Monoenergetic Gamma Ray Source

6.2.1 Source Description

In many situations in gamma shielding, the sources tend to have discrete energies. This is because many radioactive isotopes, when they decay, emit a number of gamma rays with discrete energies.

To test the generalized multigroup method using discrete sources, an oxygen infinite medium is utilized. A discrete source with an energy of 10 MeV is used. Since the source is a delta function, the integral of the source over the original energy group that contains that energy is equal to one. So, in this case, the MG source in the first energy group is $S_1 = 1$, i.e.

$$S_1 = \int_{E_1}^{E_0=10\ \mathrm{MeV}+\varepsilon} \delta(E-10MeV)dE = 1 \qquad \text{Eq. 6.4}$$

The sources in the remaining groups are $S_g = 0$ for $g = 2,\cdots,G$. Therefore, the discrete source has been smeared over the whole original energy group of the first group. The energy is again subdivided into four equal logarithmic intervals. The absorption as function of the number of energy groups is calculated. The convergence of both methods to the true solution is observed.

For the LMG source, since the monoenergetic source is at the top energy of the first group, again results in a unit source,

$$S_1 = \int_{E_1}^{E_0=10\mathrm{MeV}+\varepsilon} f_0(E)\delta(E-10MeV)dE = 1 \qquad \text{Eq. 6.5}$$

6.2.2 Comparison of Results

The results for calculating the absorption in the fourth original energy group due to the discrete energy source are listed in Table 6.6. The results from 384-group MG calculations were used as the reference results. These results seem inadequate for use as a

Table 6.6: Absorption rate in the fourth energy group

Number of Energy Groups	Multigroup Method	Error in %	Hat Basis Functions	Error in %
8	7.27146E-01	2.57E+01	7.16911E-01	2.68E+01
16	8.63270E-01	1.18E+01	9.06851E-01	7.40E+00
32	9.43502E-01	3.65E+00	9.56592E-01	2.32E+00
64	9.64895E-01	1.47E+00	9.69797E-01	9.69E-01
128	9.73992E-01	5.40E-01	9.76266E-01	3.08E-01
256	9.78025E-01	1.28E-01	9.80051E-01	7.85E-02

reference because the solution has not converged to the true solution within the desired accuracy. But, since NJOY99 limits the number of groups to 399, the 384-group solution will be used as the reference.

It is observed from the data that the solution using the hat functions converges faster to the reference solution, as shown in Figure 6.11. However, due to the discrete source in the first energy group, and the pair production at 511 keV, the solution becomes oscillatory. This is because the hat basis functions assume a smooth transition in the spectrum shape. However, the discrete source and the pair production represent sudden jumps in the spectrum. So, as the number of energy groups increases, the spectrum becomes more oscillatory because, as energy groups narrow, the sudden jump in the spectrum becomes more drastic. To compensate for the increase in the sudden jump, the fluxes in the adjacent groups become more negative. As a result, the fluxes in the following groups start to compensate for these changes by becoming, alternately, more negative and more positive. This becomes more apparent as shown in Figure 6.12 and Figure 6.13. Figure 6.12 contains the plot of the solution for 64 groups using hat functions. At the first energy group, the adjacent flux becomes negative to compensate

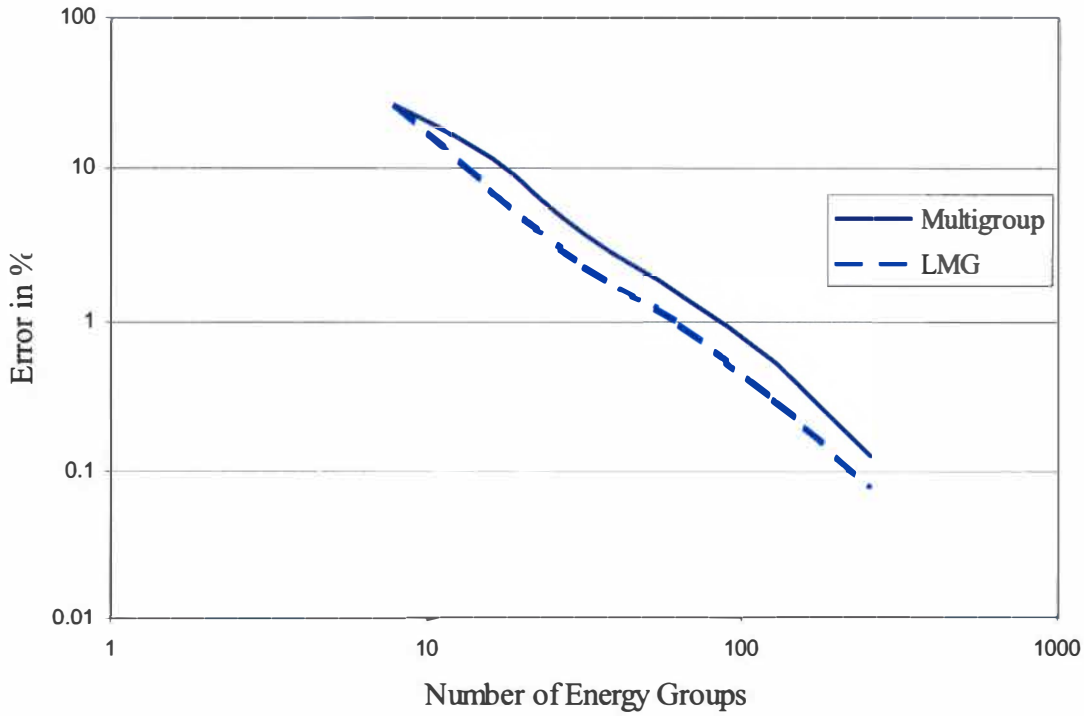


Figure 6.11: Error in absorption rate in fourth energy group as function of number of energy groups for MG method and LMG method

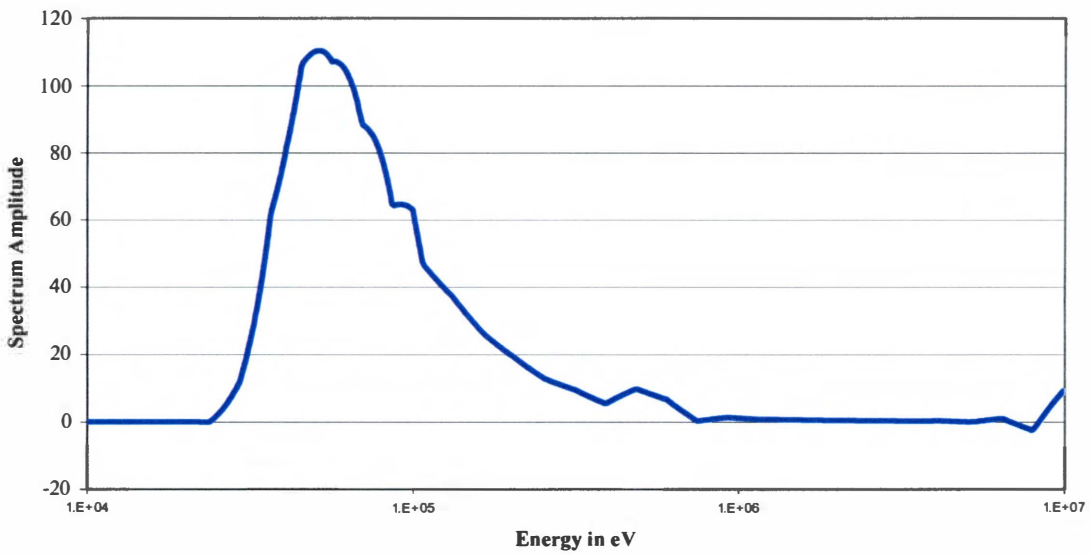


Figure 6.12: Spectrum for a discrete source using 64 groups and hat basis functions

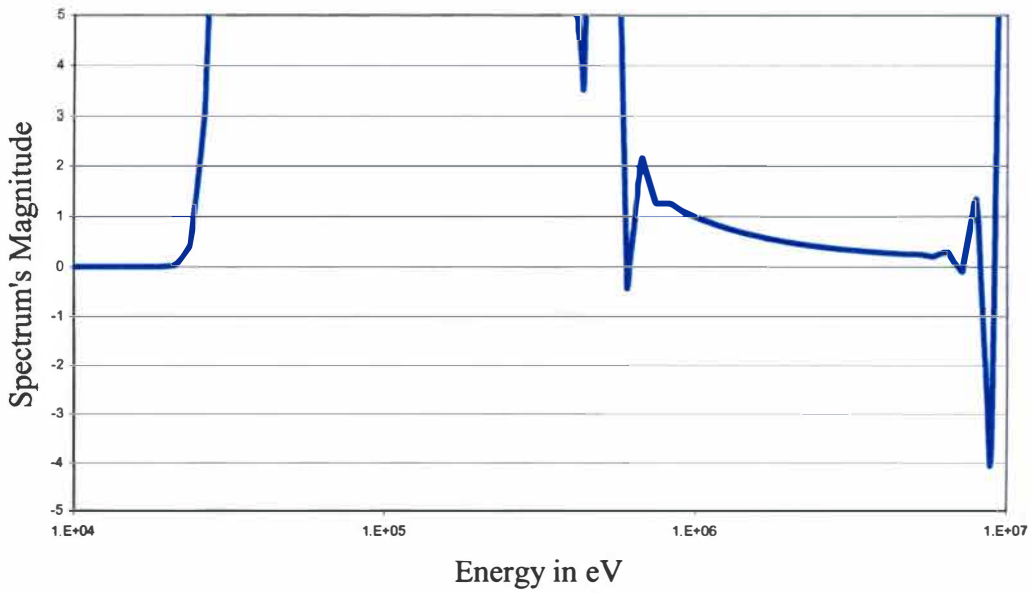


Figure 6.13: Amplification of Figure 6.12 to highlight the oscillatory behavior of the spectrum due to discrrete source and pair production energy range.

for the sudden jump in the spectrum due to the presence of the source. The calculated spectrum oscillates positive and negative for a few groups, although this is not the true behavior of spectrum. In the limit, by continuously increasing the number of energy groups, the calculated spectrum will become increasing oscillatory, and the spectrum will start to diverge from the true spectrum.

6.2.3 Use of MG Method for Discrete Source Group

To overcome the problem of discontinuities at the source and at the pair production energy, both of these groups were treated using the multigroup method in a hybrid LMG/MG calculation to allow for discontinuities in the spectrum. It would have been expected that the solution would converge faster using this approach. However, the convergence rate using this approach is slower than the conventional multigroup method.

This is because, for discrete sources, the multigroup method smears the source over the energy group. As a result, the cross sections do not represent the true response to the discrete sources. By increasing the number of energy groups, the groups eventually become narrow enough to properly represent the response to the discrete sources. Table 6.7 and Table 6.8 represent the absorption rate for the original third and fourth energy groups respectively. Figure 6.14 represents the error in absorption prediction in the original fourth group as a function of the number of energy groups.

6.2.4 Ultra Fine MG Source and Pair Production Groups

To overcome the problem of the slow rate of convergence for the discrete source, the pair production group and the source group were made ultra fine groups. The pair production group is fixed to be between 5.1E5 eV and 5.115 eV. The source group is also fixed to be between 9.998 MeV and 10.0 MeV. By making the source term ultra fine, the group cross sections are much better representative of the medium's response to the gamma rays than a conventional multigroup structure. This is because, for a discrete energy sources, the sources are "averaged" over the energy group in which they fall. However, since the cross sections represent the probabilities of events, such as scattering or absorption, taking place for a given energy, the averaging of sources over wide energy intervals does not give a true representation of such events taking place.

The probability of an event occurring at a discrete energy is usually different than the average group cross section. As the number of energy groups is increased, the cross

Table 6.7: Absorption rate in the third original group using hat basis function and MG source and Pair prodution groups

Number of Energy Groups	Multigroup Method	Error in %	Hat Functions with MG source and Pair production	Error in %
8	0.459600	3.82E+01	0.388018	1.67E+01
16	0.382205	1.50E+01	0.317931	4.37E+00
32	0.335596	9.44E-01	0.314126	5.51E+00
64	0.331842	1.85E-01	0.323760	2.62E+00
128	0.331680	2.34E-01	0.328541	1.18E+00
256	0.332185	8.23E-02	0.330903	4.68E-01
384	0.332458	-	-	-

Table 6.8: Absorption rate in the fourth original group using hat functions with MG source and pair prodution groups

Number of Energy Groups	Multigroup Method	Error in %	Hat Functions with MG source and Pair production	Error in %
8	0.7271458	2.57E+01	0.570172	4.18E+01
16	0.8632696	1.18E+01	0.835903	1.46E+01
32	0.9435016	3.65E+00	0.923706	5.68E+00
64	0.9648954	1.47E+00	0.953623	2.62E+00
128	0.9739918	5.40E-01	0.967859	1.17E+00
256	0.9780247	1.28E-01	0.974847	4.53E-01
384	0.9792822	-	-	-

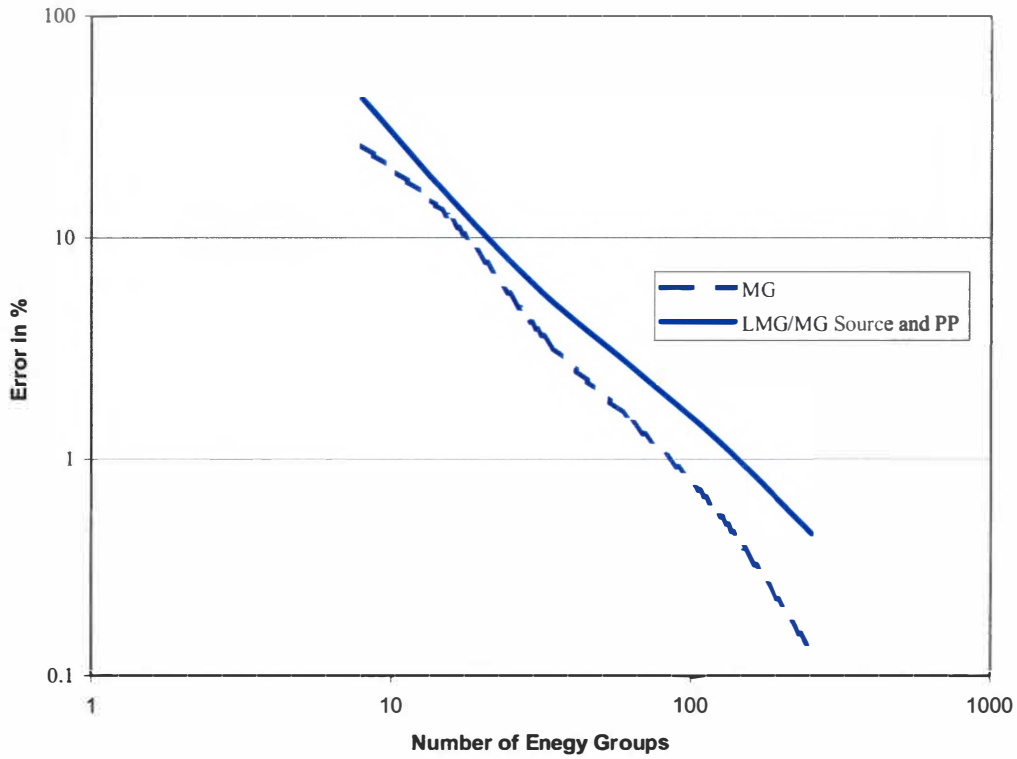


Figure 6.14: Errors for MG and LMG methods in absorption prediction of discrete source for conventional energy structure.

sections become better representatives of particle behavior in the system. As shown in Table 6.7 and Table 6.8, the convergence rate is slow. Even between 256 and 384 groups, the absorption rate is still changing in the third decimal place, which means the solution has not converged sufficiently to the true solution. The reason for this slow convergence is that even for 384 groups, the width of the first group, where the fixed source is located, is 2.0E5 eV. This energy interval is still wide given that the source gamma ray's energy is exactly 10 MeV. Therefore, a narrower energy group is needed to give better cross sections for the source group. When the gamma absorption problem is solved again with

the ultra fine groups for the pair production and source groups, the solution converge significantly more rapidly for both methods. The convergence rate improvement for the LMG method is even more rapid than for the MG method. Table 6.9 and Table 6.10 contain the absorption rates for the third and fourth original energy groups respectively. When only eight groups were utilized, the prediction error is reduced, for the MG method, from 38.2% to 5.48% in the fourth original group. For the hybrid LMG/MG method, the error is reduced from 16.7% to 12.0%. When 16 groups were utilized, the error for the multigroup method in the fourth original group is 2.85%. But for the hybrid LMG/MG method, the error is reduced to 0.11%. This is a reduction by a factor of 100 by just doubling the number of groups. Furthermore, when 32 groups are utilized, for the MG method, the error is 4.35E-1%. For the LMG/MG method, the error is 2.9E-2%. For all practical purpose, the solution has converged to the true solution for the LMG/MG within 32 groups. To reach this error magnitude, the MG method required 128 groups. The error for predicting the absorption as function of energy groups is plotted in Figure 6.15 and Figure 6.16 for the original third and fourth energy group, respectively.

6.2.5 Summary of Discrete Source Results

Solving discrete energy problem presents more challenges beyond the pair production issue faced with the continuous spectrum source problem. The discontinuity of the source forces the flux to go negative in the adjacent energy group when hat functions are utilized. This is the case for the same reasons that were mentioned for the pair production. Therefore, to solve this problem, the group that contains the discrete gamma source has to be treated using the conventional multigroup method. This allows the

Table 6.9: Absorption rate in third original energy group as function of number of groups for ultrafine source and pair production groups

Number of Energy Groups	Multigroup Method	Error in %	Hat Functions with MG source and Pair production	Error in %
8	0.19569	3.68E+01	0.283412	9.82E+01
16	0.17346	2.13E+01	0.141802	8.54E-01
32	0.14753	3.15E+00	0.142782	1.69E-01
64	0.14445	9.97E-01	0.143042	1.26E-02
128	0.14335	2.28E-01	0.143032	5.59E-03
256	0.14310	5.31E-02	0.143024	-
384	0.14305	1.82E-02	-	-

Table 6.10: Absorption rate in the fourth energy group as function of number of groups for ultrafine source and pair production groups

Number of Energy Groups	Multigroup Method	Error in %	LMG/MG source and Pair production	Error in %
8	1.10810	5.48E+00	1.03127	1.20E+01
16	1.13890	2.85E+00	1.17360	1.11E-01
32	1.16720	4.35E-01	1.17264	2.90E-02
64	1.17070	1.36E-01	1.17242	1.02E-02
128	1.17180	4.27E-02	1.17238	6.82E-03
256	1.17210	1.71E-02	1.17230	"---"
384	1.17220	8.53E-03		

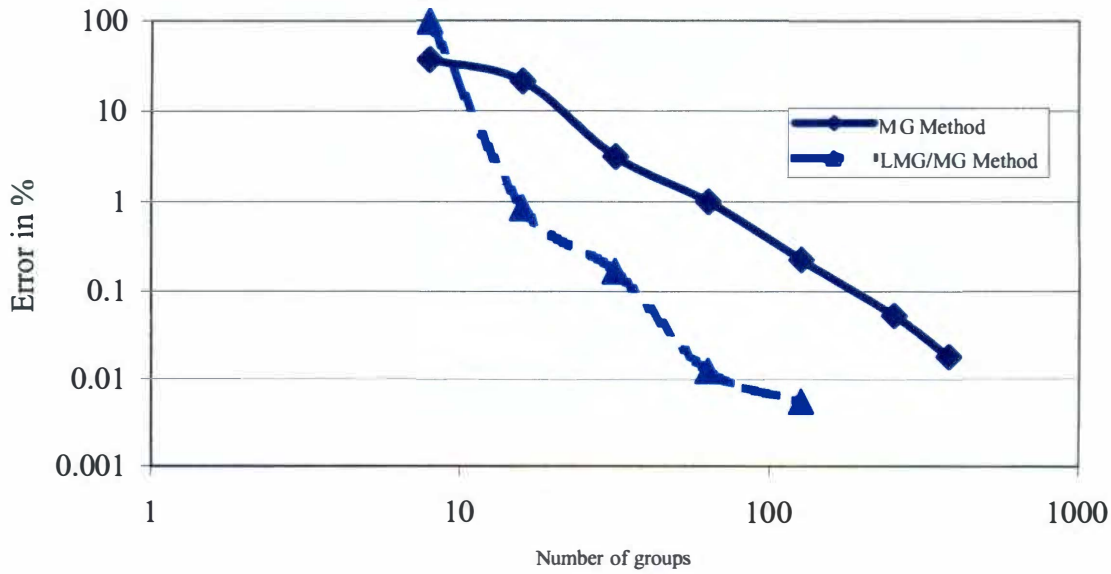


Figure 6.15 Error in absorption rate in third original group as function of number of energy groups for ultrafine source and pair production groups

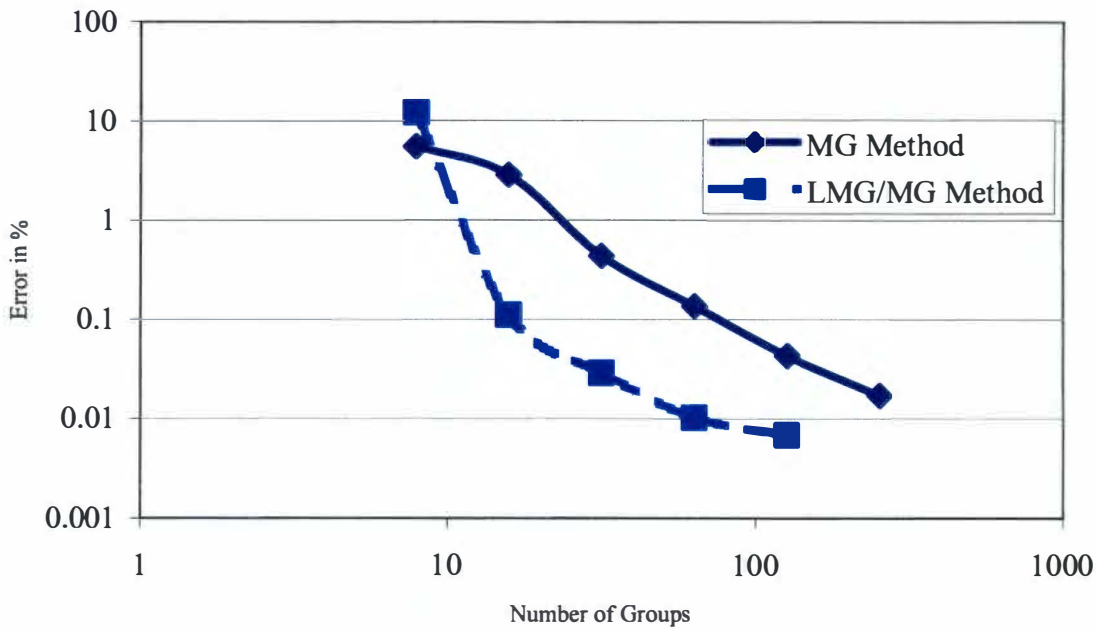


Figure 6.16: Error of absorption rate in the original fourth energy group as function of number of energy groups

spectrum in the group adjacent to the source group to adjust to the solution independently from discrete source. This approach solves the issue of negative fluxes. However, the rate of convergence of the solution is incredibly slow. The solution is still converging in the third decimal place even when 256 groups were utilized.

The slow convergence of the solution for both the MG and LMG methods is primarily due to the improper modeling of the cross sections for the source group. Since the cross sections are averaged over the energy interval of the group, this average is not representative of cross sections of the particles at that energy. Therefore, this misrepresentation of the cross sections results in the wrong reaction rates for the source group. This error is propagated all the way to the last energy group. However, by making the source group an ultra fine group, the cross sections produce more realistic reaction rates for the source group. This improvement in the reaction rate propagate to the lowest group. For this reason, narrowing the source group improved drastically the results for the MG method. The utilization of the LMG/MG method improved the solution even more. Using only 16 groups, the solution converged to 0.11% error, which is an acceptable error level for most shielding applications. This is a much more accurate solution than the MG method, which had a 2.85% error for 16 groups.

From this result, the LMG/MG method is demonstrated to be superior to the conventional method. However, the utilization of the ultra fine groups is problematic since it means that the analyst has to process new cross sections every time he has to deal with one or more different discrete sources.

7. Conclusions

In this dissertation, a new approach for treating the energy dimension of the Boltzmann transport equation has been proposed, developed, and tested. The generalized multigroup method (GMG) is a departure from the conventional method of producing cross sections for solving the neutral particle transport problem. In the conventional method, the assumed spectrum for the calculation of the cross section is fixed in shape within each energy group. This rigidity causes the calculated spectrum to be discontinuous at the energy boundaries and fixed in shape between them. These disadvantages require that the number of energy groups needed to solve a transport problem accurately be large. The proposed generalized multigroup method deals with these two problems by allowing the assumed spectrum to be adaptable between the energy boundaries. Furthermore, if certain basis functions are utilized, such as the hat basis functions, the spectrum is also constrained to be continuous on the energy boundaries. Therefore, since the new method allows the spectrum to adapt and be more representative of the true spectrum, fewer energy groups are needed to solve a neutral particle transport problem accurately.

The method was developed in a generalized form, allowing for the use of any basis functions defining the group membership functions. For demonstration purposes the method is implemented and tested using hat functions as the basis functions. These have the advantage of allowing the spectrum to adapt linearly between the energy boundaries, while maintaining spectrum continuity at the boundaries.

The generalized multigroup method has introduced new features to the transport solution such as group-to-group total cross sections, and upscattering due to the overlap in the basis functions. These two features do not exist in the multigroup method. However, even with these new features, reaction rates can be calculated by using the appropriate segments to reconstruct the conventional multigroup cross sections.

To demonstrate the performance of the new method, it has been tested on two gamma ray problems. These problems involved discrete and continuous fission gamma sources in an infinite oxygen medium. The conventional method and the new method are compared with regard to the number of energy groups needed to converge to a given error. For the continuous fission gamma source, it was discovered that the new method is superior to the conventional method. However, the pair production energy group presented a problem by causing the flux to go negative in the adjacent groups because of the discontinuity of the spectrum at that energy. Treating the pair production group using the conventional multigroup method solved the problem. This is because the conventional method allowed for discontinuity, which is a realistic modeling of the spectrum's behavior. Hence, by exploiting the advantages of the LMG and MG methods, a hybrid form of these two approaches was developed. This hybrid approach gave the best solution for the gamma absorption in an infinite medium, without the problems of negative fluxes and oscillatory spectrum that could result from using LMG alone.

The monoenergetic source problem presented further challenges because of the smearing of the source over a finite energy interval. The resulting cross sections do not give the correct reaction rates for the source group because they represent the average cross

section and not the actual cross section at the discrete energy. As the number of energy groups increases, and the groups' widths decrease, the solution eventually converges, but at a very slow rate. By creating ultra fine groups for the source and pair production groups, the convergence rate of the solution is increased for both methods. However, the convergence rate for the LMG/MG method was faster because it exploited the advantages of both approaches to model the spectrum more appropriately.

In conclusion, the implementation of the LMG has demonstrated the superiority of the generalized multigroup method over the conventional multigroup method. By using proper basis function, the GMG will give a better approximation of actual spectrum by adapting the assumed spectrum to the true one. Hence, fewer energy groups are needed to converge to the true solution. Furthermore, by exploiting the generality of the GMG, a hybrid set of basis functions could be utilized to exploit their advantages and improve the modeling of the spectrum. This was demonstrated by the use of a hybrid of LMG and MG methods to deal with discontinuities that are due to pair production and discrete sources.

8. Recommendations for Future Work

The two sample problems have illustrated the efficacy of the linear multigroup method in solving gamma ray transport problems in an infinite medium with continuous fission spectrum gamma rays. As for discrete sources, the conventional approach of using finite energy intervals does not model the problem appropriately; the use of the ultrafine energy group for the source group has improved the solution drastically. The use of a hybrid form of the LMG/MG has improved the results even more.

Several improvements and extensions to the methodology should be useful:

1. The GMG method could be extended further by developing Dirac delta based energy groups for the discrete energy sources, which are infinitesimally narrow. Hence, no particles can scatter into them, but can scatter out of them. This would give a better representation of monoenergetic sources, which should result in a more accurate solution than the conventional multigroup method as discussed in Chapter 6.

2. The GMG should be tested in spatially dependent problems. This extension would require research into spatial solution algorithms that are consistent with the new features introduced in the cross-section structure: fictitious up-scattering and group-to-group total cross sections. This should further demonstrate the advantages of an adaptable spectrum as the spectrum changes in the spatial domain.

3. The advantages of the linear multigroup method could be demonstrated and exploited in neutron problems, such as criticality or fuel pin cell calculations. The neutron problems have three characteristics that make them even better candidates for GMG than gamma problems. The first advantage is that the energy span of neutron problems is from 10^{-5} eV to $2x10^{7}$ eV, as compared to a gamma ray problem span from 10^{4} eV to 10^{7} eV. The larger energy domain of neutrons usually requires many neutron groups, which gives more potential for improvement of efficiency. Secondly, the resonance treatment of the neutron spectrum allows for a better spectrum to start with. Therefore, the spectrum is not forced to go negative to give the best solution to the problem due to the drastic mismatch between the assumed and actual spectrum. The last advantage is that the "pair-production" phenomenon that exists in gamma ray problems does not exist in neutron problems. Neutrons do not scatter to discrete energies.

To implement the new method on neutron problems, the following steps have to be taken:

1. Modify NJOY99 or any cross section codes to produce the moments needed to implement the linear multigroup method

2. Generate the cross sections with the required moments as listed in Chapter 5.

3. Assemble new cross sections based on the linear basis functions.

4. Use the new cross sections for solving pin cell and criticality problems.

Beyond the technical details of producing cross section for solving neutron problems, it is imperative that modules are developed to create these cross sections. The modules should be developed in a manner that minimizes the technical details needed for the user to generate these cross sections. Modules with input cards requiring basic information on the problem, such as energy boundaries, temperatures, and dilution coefficients, will make the generation and usage of the new cross section sufficiently simple. The wide distribution of cross section codes, such as NJOY99 and AMPX, will allow the nuclear engineering community to utilize the benefits of the new method in the shortest time possible.

4. The last recommendation is that parametric studies be carried out to optimize the group energy boundaries. Optimization of the energy boundaries could have significant improvements to the solution as shown the prompt gamma source problem.

References

References

1. A. F. Henry, Nuclear Reactor Analysis, MIT Press, Cambridge, MA (1975)
2. D. E. Cullen, "Nuclear Cross Section Preparation." in Y. Ronen, Ed. CRC Handbook of Nuclear Reactor Calculations I, CRC Press, Boca Ratom, FL (1986)
3. D. E. Cullen, "Program SIGMA1 (Version 77-1): Doppler Broaden Evaluated Cross Sections in the Evaluated Nuclear Data File/Version B (ENDF/B) Format." Lawrence Livermore National Laboratory report UCRL-50400, Vol. 17, Part B (1977).
4. W. M. Stacey, Nuclear Reactor Physics, John Wiley and Sons, (2001).
5. I. I. Bondarenko et. al., Group Constants for Nuclear Reactor Calculations, Consultants Bureau, New York (1964)
6. V. McLane, ENDF-102 Data Formats and Proceedures for the Evaluated Nuclear Data File ENDF-6, Cross Section Evaluation Working Group, National Nuclear Data Center, BNL-NCS-44945-01/04 Rev, April 2001
7. R. E. MacFarlane, "NJOY99 – Code System for Producing Pointwise, and Multigroup Neutron and Photon Cross Sections from ENDF/B Data",
8. R. E. MacFarlane, "The NJOY Nuclear Data Processing System", Version 91, LA-12740-M, October 1994.
9. S. Pearlstein, " Cross Section Evaluation Working Group History: A CSEWG Retrospective", 35th Anniversary Cross Section Evaluation Working Group", November 5, 2001, National Nuclear Data Center, Brookhaven National Laboratory, BNL-52675
10. W. M. Stacey, "Continuous Slowing Down Theory Applied to Fast-React Assemblies.", Nuclear Science and Engineering: 41, 381-393 (1970)
11. G. de Saussure and R. B. Perez, "Multilevel Effects in the Unresolved Resonances Region of the Cross Sections of Fissile Nuclides", Nuclear Science and Engineering: 52, 382-395 (1973)
12. G. W. Dixon and R. Sher, "Measurements of Thermal Neutron Spectra in Hetrogeneous Lattices by Foil Activation", Nuclear Science and Engineering: 41, 357-366 (1970)
13. Richard N. Hwang, "Efficient Methods for the Treatment of Resonance Cross Sections", Nuclear Science and Engineering: 52, 157-175 (1973)
14. M. W. Dyos and G. C. Pomraning, "Effective Thermal Neutron Cross Sections for Materials with Grain Structure", Nuclear Science and Engineering: 25, 8-11 (1966)
15. Alain Hebert, "Advance in the Development of a Subgroup Method for the Self-Shielding of Resonant Isotopes in Arbitrary Geometries", Nuclear Science and Engineering: 126, 245-263 (1997)
16. H. Roushdy, "A Corrected Formulat for Calculating the Resonance Practical Width as Function of Temperature", Ann. Nucl. Energy, Vol. 24, pp. 267-273, 1997
17. H. Roushdy, "Derivation of Modified Approximate Analytical Formalew for Calculating the Direct and Adjoint Flux Weighted Resonance Integrals Using a Variable Neutron Spectrum, Annals of Nuclear Energy, 27 (2000) 133-141
18. Tadashi Ushio, and Toshikadzu Takeda, "Application of Multiband Method to Pin Cell Calculations", Journal of Nuclear Science and Technology 39[11] pp. 1167-1174
19. G. B. Breit and E. P. Wigner, Phys. Rev., 49, p. 519 (1936)
20. C. W. Reich and M. S. Moore, Phys. Rev. 111, p. 929 (1958)

21. D. B. Adler and F. T. Adler, ANL-6792
22. L. W. Nordheim, "The Theory of Resonance Absorption," Proceedings of Symposia in Applied Mathematics, Vol. XI, p. 58, Garrett Brikhoff and Eugene P. Wigner, Eds., Am. Math. Soc. (1961)
23. C. R. Weisbin et. al., MINX: Multigroup Interpretation of Nuclear Cross Sections from ENDF/B, LA-6486-MS-(ENDF-237), Los Alamos National Laboratory, Los Alamos, NM (1976)
24. H. Henryson et. al., "MC^2-II: A Code to Calculate Fast Neutron Spectra and Multigroup Cross Sections", ANL-8144, Argone National Laboratory, Argone, IL (1976).
25. J. J. Duderstadt and L. J. Hamilton, Nuclear Reactor Analysis, Wiley, New York (1976)
26. J. H. Ferziger and P. F. Zweifel, The Theory of Slowing Down in Nuclear Reactors, MIT Press, Cambridge, MA 1966.
27. R. B. Nicholson and E. A. Fischer, "The Doppler Effect in Fast Reactors." Advances in Nuclear Science and Technology, Academic Press (1968).
28. L. Dresner, Resonance Absorption in Nuclear Reactors, Pergamon Press, Elmsford, NY (1960).
29. W. Rothernstein and M. Segev, "Unit Cell Calculations," in Y. Ronen, ed. CRC Handbook of Nuclear Reactor Calculations I, CRC Press, Boca Raton, FL (1986).
30. G. I. Bell and S. Glasstone, Nuclear Reactor Theory, Wiley (Van Nostrand Reinhold), New York (1970)
31. H. Zhang, Rizwan-Uddin, and J. J. Dorning, "Systematic Homogenization and Self-Consistent Flux and Pin Power Reconstruction for Nodal Diffusion Methods, Part I: Diffusion Theory Based Theory," Nucl. Sci. Eng., 121, 226 (1995).
32. H. Zhang, Rizwan-uddin, and J.J. Dorning, "A Multiple-Scales Systematic Theory for the Simultaneous Homogenization of Lattice Cell Fuel Assemblies," J. Transport Thoery Stat. Phys., 26, 765 (1997).
33. K. S. Smith, "Assembly Homogenization Techniques for Light Water Analysis", Prog. Nucl. Eng., 14, 303 (1986).

Appendices

Appendix A

Linear Functions, Sources and Energy

Boundaries Code

Appendix A contains the source code for the program GUIDES. This codes reads an input deck containg group energy structure, number of energy groups, types of cross sections (MG or LMG), and energy boundaries (if required). The code then proceeds calculate the energy boundaries, the coefficients for the hat basis functions, and group sources. It writes the energy boundaries and coefficients of the hat functions onto data files to be utilized by other codes.

```
      character*80 line
      real*8 ebound(1100),CL0(1100),CL1(1100),CR0(1100),CR1(1100)
       real*8 domain,SC_L(1100),SC_R(1100),SC_MG(1100),SC_Bi(1100)
       real*8 EG(1100),DelE,xsdbd(1100),SC(1100),SCPAIR(1100)
       REAL*8 SMGPA(1100)
       integer nbounds,count,counter,option,ppeg,model
       open(1,file='c:\codes\shielding\energy.inp',form='formatted',
       +status='old')
       read(1,'(a80)')line
       READ(1,'(24x,I10)')option
       read(1,'(24x,I10)')N
       print*,'Number of Energy Groups  ',N
       READ(1,'(24x,I10)')model
       read(1,'(26x,e15.2)')EL
       print*,'Low Energy',EL
       read(1,'(25x,e15.2)')EH
       print*,'High Energy  ',EH
       IF(option.eq.2)then
       do j=1,N+1
       READ(1,'(3x,e15.6)')ebound(j)
       end do
       ebound(N)=9.998E6
       GOTO 77
       ELSEIF(option.eq.1)then
c     define the boundary of the energy groups
       DelE=(log10(EH)-log10(EL))/(N)
       print *,'Delta E',DelE
       do j=1,N+1
       EG(j)=log10(EL)+(j-1)*DelE
       enddo
       do j=1,N+1
       ebound(j)=10**EG(j)
       print*,ebound(j)
       enddo
       GOTO 77
       ELSEIF(option.eq.3)then
       DelE=(log10(5.105E5)-log10(EL))/(N/2)
       do j=1,N/2
       EG(j)=LOG10(EL)+(j-1)*DelE
       end do
       EG(N/2+1)=LOG10(5.105E5)
      DelE=(log10(EH)-log10(5.115E5))/(N/2.-1.)
       EG(N/2+2)=LOG10(5.115E5)
       do j=1,N/2-1
      EG(j+N/2+2)=log10(5.115E5)+(j)*DelE
      enddo
     do j=1,N+1
      ebound(j)=10**EG(j)
      print*,ebound(j)
      enddo
c      make the first group very narrow
      ebound(N)=0.9999E7
      ELSEIF(option.eq.6)then
      DelE=(log10(5.105E5)-log10(2.5E4))/(N/2.-1.)
      EG(1)=LOG10(EL)
      do j=1,N/2-1
      EG(j+1)=LOG10(2.5e4)+(j-1)*DelE
      end do
```

```
        EG(N/2+1)=LOG10(5.105E5)
        DelE=(log10(EH)-log10(5.115E5))/(N/2.-1.)
        EG(N/2+2)=LOG10(5.115E5)
        do j=1,N/2-1
        EG(j+N/2+2)=log10(5.115E5)+(j)*DelE
        enddo
        do j=1,N+1
        ebound(j)=10**EG(j)
        enddo
c       make the first group very narrow
c        ebound(N)=0.9995E7
        ELSEIF(option.eq.4)then
        DelE=(log10(5.10E5)-log10(EL))/(N/2)
        do j=1,N/2
        EG(j)=LOG10(EL)+(j-1)*DelE
        end do

        EG(N/2+1)=LOG10(5.10E5)
        EG(N/2+2)=LOG10(5.11005e5)
        EG(N/2+3)=LOG10(5.12E5)

        DelE=(log10(EH)-log10(5.11005e5))/(N/2-2)

        do j=1,N/2-2
        EG(j+N/2+3)=log10(5.11005E5)+(j)*DelE
        enddo

        do j=1,N+1
        ebound(j)=10**EG(j)
        print*,'Hello 2',ebound(j)
        enddo

        ELSEIF(option.eq.5)then
        DelE1=(LOG10(5.11e5)-LOG10(EL))/(N/2)
        DelE2=(LOG10(9.9999e6)-LOG10(5.11005e5))/(N/2)

        do j=1,N/2+1
        EG(j)=LOG10(EL)+(j-1)*delE1
        end do

        do j=1,(N/2)+1
        EG(j+N/2+1)=LOG10(5.11005e5)+(j-1)*DelE2
        end do

        EG(N+3)=LOG10(EH)

        do j=1,N+3
        ebound(j)=10**EG(j)
        end do

        endif
      close(1)

c       ebound(1)=1e-5
        IF(option.eq.5)then
        N=N+2
        endif
  77    continue
        do j=1,N+1
        print*,ebound(j)
        enddo
        open(2,file='c:\codes\shielding\engbd.inp',form='formatted',
        +status='old')
        write(2,'(a30)')'Energy Boundaries for NJOY'
        write(2,'(I10)')N
        do j=1,N+1
        write(2,'(1pE15.6)')ebound(j)
        enddo
c     Invert the order of the energy in a similar manner used
```

```
c     by XSDRN-PM as opposed to NJOY
      OPEN(7,file='c:\codes\shielding\engbd2.inp',form='formatted',
       +status='unknown')
        WRITE(7,'(I10)')N
       do j=1,N+1
       xsdbd(j)=ebound(N+2-j)
       WRITE(7,'(1pE17.6)')xsdbd(j)
       enddo
c     find pair production energy group
      ppe=511003.4
      do j=1,N
      IF(ppe.gt.xsdbd(j+1).and.ppe.lt.xsdbd(j))then
      ppeg=j
      endif
      end do
c     Do the loops that finds the coefficients for the linear functions
      do i=1,N
       den=xsdbd(i)-xsdbd(i+1)
        CL0(i)=xsdbd(i)/den
        CL1(i)=-1/den
        CR0(i)=-xsdbd(i+1)/den
        CR1(i)=1/den
        enddo
      open(3,file='c:\codes\shielding\coeff.inp',form='formatted',
       +status='old')
       write(3,'(a30)')'Coefficients for the Guide Functions'
        WRITE(3,'(I10)')N
       do j=1,N
       write(3,'(I5,4(1x,1pE15.8))')j,CL0(j),CL1(j),CR0(j),CR1(j)
       enddo
        WRITE(3,'(a30)')'Pair Production Energy Group'
        WRITE(3,'(I5)')ppeg
        WRITE(3,'(a30)')'Fraction of Pair Production'
        WRITE(3,'(1x,1pE15.8)')ppf
        WRITE(2,'(a50)')'Source terms.
        s1=CR1(1)/2*1.95e7+CR0(1)
        s2=CL1(1)/2*1.95e7+CL0(1)
        WRITE(2,'(E15.7)')s1
        WRITE(2,'(E15.7)')s2
        sc(1:1100)=0.0
       summa=0.0
       summg=0.0
      open(73,file='c:\codes\shielding\sor_spec.inp',form='formatted',
     +status='unknown',position='append')
       do j=1,N
       tot_left=0.0
       tot_rig=0.0
       tot_MG=0.0
       kounter=0
       do i=1,1000000
       kounter=kounter+1
       Call RANDOM_NUMBER(ERand)
       ER=(ERand*(xsdbd(j)-xsdbd(j+1))+xsdbd(j+1))
       IF(ER.lt.0.6e6)then
       prod=6.6
       ELSEIF(ER.ge.0.6e6.and.ER.lt.1.5e6)then
       prod=20.2*EXP(-1.78e-6*ER)
       ELSEIF(ER.ge.1.5e6)then
       prod=7.2*EXP(-1.09E-6*ER)
       endif
       if (kounter.eq.10000) then
       WRITE(73,'(2E17.7)')ER,prod
       Kounter=0
       end if
       tot_left=tot_left+prod*(CL0(j)+ER*CL1(j))/1.0e6
       tot_rig=tot_rig+prod*(CR0(j)+ER*CR1(j))/1.e6
       tot_mg=tot_mg+prod/1.e6
       end do
       SC_L(j)=tot_left*(xsdbd(j)-xsdbd(j+1))/1.e6
```

```
      SC_R(j)=tot_rig*(xsdbd(j)-xsdbd(j+1))/1.e6
      SC_MG(j)=tot_mg*(xsdbd(j)-xsdbd(j+1))/1.e6
      summa=summa+SC_L(j)+SC_R(j)
      summg=summg+SC_MG(j)
      end do
      CLOSE(73)
      do j=1,N
      end do
      SC(1)=SC_R(1)
      do i=2,N
      SC(i)=SC_L(i-1)+SC_R(i)
      end do
      SC(N+1)=SC_L(N)
      SCPAIR(1)=SC_R(1)
     do j=2,N
     if(j.lt.ppeg)then
     SCPAIR(j)=SC_R(j)+SC_L(j-1)
     elseif(j.eq.ppeg)then
     SCPAIR(j)=SC_L(j-1)
     SCPAIR(j+1)=SC_R(j)
     SCPAIR(j+2)=SC_L(j)
     SCPAIR(j+3)=SC_R(j+1)
     elseif(j.gt.ppeg+1)then
     SCPAIR(j+2)=SC_R(j)+SC_L(j-1)
     endif
     enddo
     SCPAIR(N+3)=SC_L(N)
c     reconstruct the source term for mutligroup pair.
     SMGPA(1)=SC_R(1)
     do j=2,N-1
     if(j.lt.ppeg)then
     SMGPA(j)=SC_R(j)+SC_L(j-1)
     elseif(j.eq.ppeg)then
     SMGPA(j)=SC_L(j-1)
     SMGPA(j+1)=SC_R(j)+SC_L(j)
     SMGPA(j+2)=SC_R(j+1)
     elseif(j.ge.ppeg+1)then
     SMGPA(j+2)=SC_R(j+1)+SC_L(j)
     endif
     enddo
     SMGPA(N+2)=SC_L(N)
     tem=0.0
     do j=1,N+2
     tem=smgpa(j)+tem
     end do
c     compare multigroup and hat sources
     tsum=0.0
     do i=1,N
     tsum=sc_L(i)+SC_R(i)-SC_MG(i)
     SC_MG(i)=SC_L(i)+SC_R(i)
     SC_Bi(2*i-1)=SC_R(i)
     SC_Bi(2*i)=SC_L(i)
     end do
     open(9,file='c:\codes\shielding\cp_source.inp',form='formatted',
    +status='unknown')
     WRITE(9,'(a40)')'Source Term for Hat Functions'
     WRITE(9,'(I15)')N
     do i=1,N+1
     WRITE(9,'(E18.7)')SC(i)/summa
     end do
     CLOSE(9)
     open(12,file='c:\codes\shielding\bi_source.inp',form='formatted',
    +status='unknown')
     WRITE(12,'(a40)')'Source Term for Hat Functions'
     WRITE(12,'(I15)')2*N
     do i=1,2*N
     WRITE(12,'(E18.7)')SC_Bi(i)/summa
     end do
     CLOSE(12)
```

```
 open(10,file='c:\codes\shielding\MG_source.inp',form='formatted',
+status='unknown')
 WRITE(10,'(a40)')'Source Term for Multigroup Functions'
 WRITE(10,'(I15)')N
 do i=1,N
 WRITE(10,'(E18.7)')SC_MG(i)/summa
 end do
 CLOSE(10)
 open(11,file='c:\codes\shielding\pa_source.inp',form='formatted',
+status='unknown')
 WRITE(11,'(a40)')'Source Term for pair Hat Functions'
 WRITE(11,'(I15)')N
 do i=1,N+3
 WRITE(11,'(E18.7)')SCPAIR(i)/summa
 end do
 CLOSE(11)
 open(13,file='c:\codes\shielding\mgpa1.inp',form='formatted',
+status='unknown')
 WRITE(13,'(a40)')'Source Term for mutligroup pair production'
 WRITE(13,'(I15)')N
 do i=1,N+2
 WRITE(13,'(E18.7)')SMGPA(i)/tem
 end do
 CLOSE(13)
 close(2)
   stop
   end
```

Appendix B

Cross Sections Code for Hat

Basis Functions

Appendix B contains the source code for the READWRK code. This code reads the six working libraries resulting the six NJOY99 runs. It also reads the coefficients for the hat basis functions. It then proceeds to calculate the group cross sections for the LMG method. Next, it reads the group sources from an input file generated by GUIDES code and solves the gamma spectrum problem in an infinite medium.

```
      program main
      integer IDTAPE,NNUC,IGM,IFTG,MSN,ID1(10),ID3(3),ID5(7),ID7(5)
      integer ID9(1),M,count
      integer MT(20),MAGIC(20,280)
      integer NZ(10),z,MT2D(40),L(100),NT(100)
      real EB(300),UB(300),ID2(2),ID4(2),ID6(1)
      real SIGMA(6,20,280),sigmar(280),sigmal(280),sigmac(280)
      real scatfin(10,280,280),sigmaf(280),sigmat(280)
      real SCATMAT(6,10,280,280),MTS(10),orgscat(10,280,280)
      real temscat1(280),temscat2(280),mgabs(280)
      real flx_l(280),flx_r(280),flux(280),tflux(280)
      real flux1(280),flux2(280),sumflx(280),absor(280)
      integer Strngmat(20,280),Strnglen(20)
      integer BOUNDS(20,280,2),Sum(280),tsum(20),ppeg
      real d(30000),CL0(280),CL1(280),CR0(280),CR1(280)
      real sigabst(280),sigabsl(280),sigabsr(280),sigabsc(280)
      real sigpairl(280),sigpairr(280),sigpairf(280)
      real sigabsf(280),TEMPMT(2000),ppf,SC(280),E(2418),SP(2418)
      real dd(10,30000),totscat(280),MAT(280,280),element
      character*4 TITLE(18),ID8(1)
      character*400 tset
      character*30 coftitle
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     Read the working library after being processed by         c
c     Nitwal.                                                   c
c     Read the guide functions coefficients from file           c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      open(22,file='dumout.out',form='formatted',status='old')
      open(20,file='coeff.inp',form='formatted',status='old')
        read(20,'(a30)')coftitle
        read(20,'(I10)')N
        do j=1,N
       read(20,'(I5,4(1x,E15.8))')KN,CL0(j),CL1(j),CR0(j),CR1(j)
       enddo
        read(20,'(a30)')coftitle
        read(20,'(I5)')ppeg
        read(20,'(a30)')coftitle
        read(20,'(1x,E15.8)')ppf
        close(20)
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                               c
c     open all three files for the diagonal terms that do not   c
c     have scattering terms.                                    c
c                                                               c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      SCATMAT(1:6,1:10,1:280,1:280)=0.0
      orgscat(1:10,1:280,1:280)=0.0
      do jj=1,6
      if(jj.eq.1)then
      open(1,file='flat_mg.wrk',form='unformatted',status='old')
      elseif(jj.eq.2)then
      open(2,file='lin_mg.wrk',form='unformatted',status='old')
      elseif(jj.eq.3)then
      open(3,file='qua_mg.wrk',form='unformatted',status='old')
      elseif(jj.eq.4)then
      open(4,file='ee_mg.wrk',form='unformatted',status='old')
      elseif(jj.eq.5)then
      open(5,file='ef_mg.wrk',form='unformatted',status='old')
      elseif(jj.eq.6)then
```

```
      open(6,file='e_flx.wrk',form='unformatted',status='old')
      endif
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     read first header that has general information about      c
c     the data.                                                 c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      read(jj)IDTAPE,NNUC,IGM,IFTG,MSN,IPM,I1,I2,I3,I4,
     +tset
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c  read energy groups boundaries if it is a neutron problem or  c
c  gamma problem.                                               c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
       read(jj)(EB(I),I=1,IPM+1),(UB(I),I=1,IPM+1)
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     Read the fifty word title card.                           c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      read(jj)TITLE,ID1,ID2,ID3,ID4,ID5,ID6,ID7,ID8,ID9
      read(jj)TITLE,ID1,ID2,ID3,ID4,ID5,ID6,ID7,ID8,ID9
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     Read temperature independent cross sections for gamma rays.  c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      if(ID5(6).ne.0)then
        read(jj)(MTS(I),(SIGMA(jj,I,J),J=1,IPM),I=1,ID5(6))
      end if
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     retrieve the  scattering matrix terms and unscramble them.    c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      do j=1,ID1(6)+1
      read(jj)L(j),(d(i),i=1,L(j))
      mw=iset(d(1))
      call unscramble(d,L(j),scatmat,IPM,6,10,j,jj)
      enddo
      enddo
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c         Data up to this point has been read. The next     c
c         step is "unscale" the data.                       c
c       Look for the MT1599                                 c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
        if(ID5(6).ne.0)then
           do j=1,ID5(6)
             if(MTS(j).eq.1599.0)then
             IS=j
             endif
           enddo
        endif
        do j=1,ID5(6)
        if(MTS(j).eq.516.0)then
        ISP=j
        endif
        enddo
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     Start Scaling the scalar processes.                      c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
        if(ID5(6).ne.0)then
           do jz=1,6
           do j=1,ID5(6)
             if(j.ne.IS)then
               do i=1,IPM
                SIGMA(jz,j,i)=SIGMA(jz,j,i)*SIGMA(jz,IS,i)/100000.0
               enddo
             endif
          enddo
          enddo
        endif
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c       Now Scaling the Scattering Matrix.                       c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      do jj=1,5
        do k=1,ID1(6)+1
```

```
            do i=1,IPM
              do j=1,IPM
         scatmat(jj,k,j,i)=scatmat(jj,k,j,i)*SIGMA(1,IS,i)/100000.0
              enddo
            enddo
          enddo
        enddo
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c      Now, manipulate the matrices to produce the hat cross       c
c      sections.                                                     c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
        if(ID5(6).ne.0)then
        do j=1,ID5(6)
        if(MTS(j).eq.501.0)then
        IS1=j
        endif
        enddo
        do j=1,ID5(6)
        if(MTS(j).eq.527.0)then
        IS2=j
        endif
        enddo
        endif
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c        Do all the guide functions.                                 c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
         do j=1,280
         sigmaf(j)=0.0
         enddo
       do i=1,ID1(6)+1
         do   j=1,IPM+1
           do   k=1,IPM+1
              scatfin(i,j,k)=0.0
           enddo
         enddo
       enddo
       do j=1,IPM
       sigmar(j)=CR0(j)**2*SIGMA(1,IS1,j)+2*CR0(j)*CR1(j)*
      +SIGMA(2,IS1,j)+CR1(j)**2*SIGMA(3,IS1,j)
       sigmal(j)=CL0(j)**2*SIGMA(1,IS1,j)+2*CL0(j)*
      +CL1(j)*SIGMA(2,IS1,j)+CL1(j)**2*SIGMA(3,IS1,j)
       scatfin(1,j,j+1)=-(CR0(j)*CL0(j)*SIGMA(1,IS1,j)+(CR1(j)*CL0(j)+
      +CR0(j)*CL1(j))*SIGMA(2,IS1,j)+CR1(j)*CL1(j)*SIGMA(3,IS1,j))
       scatfin(1,j+1,j)=scatfin(1,j,j+1)
       sigmac(j)=-2*scatfin(1,j+1,j)
       sigmat(j)=sigmar(j)+sigmal(j)+sigmac(j)
       enddo
       do j=1,IPM
       sigabsr(j)=CR0(j)*SIGMA(1,IS2,j)+CR1(j)*SIGMA(2,IS2,j)
       sigabsl(j)=CL0(j)*SIGMA(1,IS2,j)+CL1(j)*SIGMA(2,IS2,j)
       enddo
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c            Reconstruct the absoption cross sections               c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
       sigmaf(1)=sigmar(1)
       do j=2,IPM
       sigmaf(j)=sigmar(j)+sigmal(j-1)
       enddo
       sigmaf(IPM+1)=sigmal(IPM)
       sigabsf(1)=sigabsr(1)
       do j=2,IPM
       sigabsf(j)=sigabsr(j)+sigabsl(j-1)
       enddo
       sigabsf(IPM+1)=sigabsl(IPM)
       do i=1,IPM
       flx_l(i)=(CL0(i)*sigma(1,IS,i)+CL1(i)*sigma(6,IS,i))/100000.0
       flx_r(i)=(CR0(i)*sigma(1,IS,i)+CR1(i)*sigma(6,IS,i))/100000.0
       enddo
       flux(1)=flx_r(1)
```

```
      do j=2,IPM
      flux(j)=flx_l(j-1)+flx_r(j)
      enddo
      flux(IPM+1)=flx_l(IPM)
      do j=1,IPM+1
      sigmaf(j)=sigmaf(j)/flux(j)
      sigabsf(j)=sigabsf(j)/flux(j)
      enddo
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     scale absorption cross sections for calculation       c
c     for multigroup absorption calculation.                c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      do j=1,IPM
      sigabsr(j)=sigabsr(j)/flux(j)
      sigabsl(j)=sigabsl(j)/flux(j+1)
      enddo
      open(25,file='absorption.inp',form='formatted',status='unknown')
      write(25,'(I10)')IPM
      do j=1,IPM
      write(25,'(I10,2E15.8)')j,sigabs1(j),sigabsr(j)
      enddo
      close(25)
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c        Calculate the new scattering matrix.                  c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c       first set is right-right guide, then set is right left guide
c       third set is left right guide. then set is left left guide
      do i=1,ID1(6)+1
        do k=1,IPM
          do j=1,IPM
      tempo1=CR0(k)*CR0(j)*scatmat(1,i,j,k)+
     +  CR1(k)*CR0(j)*scatmat(2,i,j,k)+CR1(k)*CR1(j)*scatmat(4,i,j,k)+
     +  CR0(k)*CR1(j)*scatmat(5,i,j,k)
        scatfin(i,j,k)=scatfin(i,j,k)+tempo1
        tempo2=CR0(k)*CL0(j)*scatmat(1,i,j,k)+
     +  CR1(k)*CL0(j)*scatmat(2,i,j,k)+CR1(k)*CL1(j)*scatmat(4,i,j,k)+
     +  CR0(k)*CL1(j)*scatmat(5,i,j,k)
           scatfin(i,j+1,k)=scatfin(i,j+1,k)+tempo2
        tempo3=CL0(k)*CR0(j)*scatmat(1,i,j,k)+
     +  CL1(k)*CR0(j)*scatmat(2,i,j,k)+CL1(k)*CR1(j)*scatmat(4,i,j,k)+
     +  CL0(k)*CR1(j)*scatmat(5,i,j,k)
          scatfin(i,j,k+1)=scatfin(i,j,k+1)+tempo3
           tempo4=CL0(k)*CL0(j)*
     +  scatmat(1,i,j,k)+CL1(k)*CL0(j)*scatmat(2,i,j,k)+
     +  CL1(k)*CL1(j)*scatmat(4,i,j,k)+CL0(k)*CL1(j)*scatmat(5,i,j,k)
      scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo4
      enddo
      enddo
      enddo
      do j=1,IPM
      scatfin(1,ppeg,j)=(CR0(j)*sigma(1,ISP,j)+
     +CR1(j)*sigma(2,ISP,j))+scatfin(1,ppeg,j)
      scatfin(1,ppeg+1,j)=(CR0(j)*sigma(1,ISP,j)
     ++CR1(j)*sigma(2,ISP,j))+scatfin(1,ppeg+1,j)
      scatfin(1,ppeg,j+1)=(CL0(j)*sigma(1,ISP,j)+
     +CL1(j)*sigma(2,ISP,j))+scatfin(1,ppeg,j+1)
      scatfin(1,ppeg+1,j+1)=(CL0(j)*sigma(1,ISP,j)+
     +CL1(j)*sigma(2,ISP,j))+scatfin(1,ppeg+1,j+1)
      end do
      do j=1,IPM
      sigpairr(j)=CR0(j)*sigma(1,ISP,j)+CR1(j)*sigma(2,ISP,j)
      sigpairl(j)=CL0(j)*sigma(1,ISP,j)+CL1(j)*sigma(2,ISP,j)
      end do
      sigpairf(1)=sigpairr(1)
      do j=2,IPM
      sigpairf(j)=sigpairr(j)+sigpairl(j-1)
      enddo
      sigpairf(IPM+1)=sigpairl(j)
      do j=1,IPM+1
```

```
      sigpairf(j)=sigpairf(j)/flux(j)
      enddo
      do j=1,IPM
      tempo1=ppf*2.0*(CR0(j)*sigma(1,ISP,j)+CR1(j)*sigma(2,ISP,j))
      tempo2=(1.-ppf)*2.0*(CR0(j)*sigma(1,ISP,j)+CR1(j)*sigma(2,ISP,j))
      tempo3=ppf*2.0*(CL0(j)*sigma(1,ISP,j)+CL1(j)*sigma(2,ISP,j))
      tempo4=(1.-ppf)*2.0*(CL0(j)*sigma(1,ISP,j)+CL1(j)*sigma(2,ISP,j))
      enddo
      do i=1,ID1(6)+1
       do j=1,IPM+1
          do k=1,IPM+1
          scatfin(i,k,j)=scatfin(i,k,j)/flux(j)
        enddo
       enddo
      enddo
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c        Solve for an infinite Medium.                              c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
        SC(1:280)=0.0
        flux1(1:280)=0.0
        flux2(1:280)=0.0
        alpha=0.05
c     This is for reading fixed source term.
      open(25,file='c:\codes\shielding\engbd.inp',form='formatted',
     +status='old')
        read(25,'(a30)')
        read(25,'(I10)')N
        close(25)
         SC(1)=1.0
c      open(30,file='c:\codes\shielding\cp_source.inp',form='formatted',
c     +status='old')
c      read(30,'(a30)')
c      read(30,'(I15)')N
c      do j=1,N+1
c      read(30,'(E18.7)')SC(j)
c      enddo
      element=3.29674E-2
       do j=1,IGM
       sigmaf(j)=sigmaf(j)*element
       enddo
       do j=1,IGM
       do i=1,IGM
       scatfin(1,i,j)=-element*scatfin(1,i,j)
       enddo
       enddo
       do j=1,IGM
       scatfin(1,j,j)=scatfin(1,j,j)+sigmaf(j)
       enddo
       count=1
       count2=1
  41   continue
       do i=1,IGM
       ssum=0.0
       do j=Bounds(1,i,1),Bounds(1,i,2)
       ssum=ssum+scatfin(1,i,j)*flux1(j)
       enddo
       flux2(i)=flux1(i)+alpha*(sc(i)-ssum)
       enddo
       err=abs(flux1(1)-flux2(1))
       meter=count/10000
       if(meter.eq.count2)then
       print*,count2,flux1(1),flux2(1),err
       count2=count2+1
       endif
       do j=1,IGM
       flux1(j)=flux2(j)
       enddo
       count=count+1
      if (count.gt.1e4) then
```

```
       alpha=0.2
       end if
       if(count.le.1E5)then
       goto 41
       endif
      do j=1,N+1
      mgabs(j)=(flux2(j)*sigabsr(j)+flux2(j+1)*sigabsl(j))*element
      enddo
       k=N/4
       print*,N,K
       tabs=0.0
       do j=1,4
       do i=1,k
       absor(j)=absor(j)+mgabs((j-1)*k+i)
       enddo
       enddo
       tabs=0.0
       do j=1,4
       tabs=tabs+absor(j)
       enddo
       do j=1,4
       print*,j,absor(j)
       enddo
       print*,tabs
      open(45,file='c:\codes\shielding\chapabs.out',form='formatted',
     +status='unknown',position='append')
      write(45,'(2x,I5,5(4x,E16.7))')N,(absor(j),j=1,4),tabs
      close(45)
      open(49,file='c:\codes\shielding\chapflx.out',form='formatted',
     +status='unknown',position='append')
      write(49,'(I10)')N
      do j=1,N+1
      write(49,'(2x,I2,4x,E16.7)')j,flux2(j)
      enddo
      close(49)
      close(22)
c  Reproduce the spectrum for hat functions
      open(32,file='c:\codes\shielding\engbd.inp',form='formatted',
     +status='old')
      read(32,'(a30)')
      read(32,'(I10)')N
      do j=1,N+1
      read(32,'(E15.6)')EB(j)
      enddo
      close(32)
      open(34,file='c:\codes\shielding\chapsp.out',form='formatted',
     +status='unknown',position='append')
      write(34,'(I15)')N
      open(35,file='c:\codes\shielding\New_spectrum.dat',
     +form='formatted',status='old')
      do j=1,2418
      read(35,'(E13.7,8x,E13.7)')E(j),SP(j)
      enddo
       do j=1,2418
       do i=1,N
       if(E(j).lt.EB(i+1).and.E(j).ge.EB(i))then
       sp(j)=sp(j)*(CR0(N-i+1)+CR1(N-i+1)*E(j))*flux2(N-i+1)/
     +flux(N-i+1)+
     +sp(j)*(CL0(N-i+1)+E(j)*CL1(N-i+1))*flux2(N-i+2)/flux(N-i+2)
       write(34,'(E17.6,i10)')sp(j)
       endif
       enddo
       enddo
  30  continue
        stop
        end
```

Appendix C

Cross Sections Code for Hat Basis Functions and MG Pair Production

Appendix C contains the source code for the MGPAIR code. This code is similar to the READWRK code. However, it treats the pair production group as a conventional MG. Hence, the spectrum is not adaptable within the pair production energy group. This code reads the six working libraries data for the six NJOY99 runs and the coefficients for the hat basis functions. Next, It proceeds to calculate the group cross section for the LMG/MG method, and solvesthen reads the group sources from an input file generated by GUIDES code, and solves the gamma spectrum problem in an infinite medium.

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c        Do all the guide functions.                                      c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
        do j=1,280
        sigmaf(j)=0.0
        enddo

       do i=1,ID1(6)+1
         do   j=1,IPM+1
           do   k=1,IPM+1
              scatfin(i,j,k)=0.0
           enddo
         enddo
       enddo

       do j=1,IPM
       sigmar(j)=CR0(j)**2*SIGMA(1,IS1,j)+2*CR0(j)*CR1(j)*
      +SIGMA(2,IS1,j)+CR1(j)**2*SIGMA(3,IS1,j)
       sigmal(j)=CL0(j)**2*SIGMA(1,IS1,j)+2*CL0(j)*
      +CL1(j)*SIGMA(2,IS1,j)+CL1(j)**2*SIGMA(3,IS1,j)
       enddo

       do j=1,IPM
       if(j.lt.ppeg)then
       scatfin(1,j,j+1)=-(CR0(j)*CL0(j)*SIGMA(1,IS1,j)+(CR1(j)*CL0(j)+
      +CR0(j)*CL1(j))*SIGMA(2,IS1,j)+CR1(j)*CL1(j)*SIGMA(3,IS1,j))
       scatfin(1,j+1,j)=scatfin(1,j,j+1)
       elseif(j.eq.ppeg)then
       scatfin(1,j+1,j+1)=-(CR0(j)*CL0(j)*SIGMA(1,IS1,j)+(CR1(j)*CL0(j)+
      +CR0(j)*CL1(j))*SIGMA(2,IS1,j)+CR1(j)*CL1(j)*SIGMA(3,IS1,j))
       scatfin(1,j+1,j+1)=2*scatfin(1,j+1,j+1)
       elseif(j.gt.ppeg)then
       scatfin(1,j+1,j+2)=-(CR0(j)*CL0(j)*SIGMA(1,IS1,j)+(CR1(j)*CL0(j)+
      +CR0(j)*CL1(j))*SIGMA(2,IS1,j)+CR1(j)*CL1(j)*SIGMA(3,IS1,j))
       scatfin(1,j+2,j+1)=scatfin(1,j+1,j+2)
       endif
       enddo

       do j=1,IPM
       sigabsr(j)=CR0(j)*SIGMA(1,IS2,j)+CR1(j)*SIGMA(2,IS2,j)
       sigabsl(j)=CL0(j)*SIGMA(1,IS2,j)+CL1(j)*SIGMA(2,IS2,j)
       enddo

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c            Reconstruct the absoption cross sections                     c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc


c   reconstruct the new hat cross sections
      sigmaf(1)=sigmar(1)
      do j=2,IPM
      if(j.lt.ppeg)then
      sigmaf(j)=sigmar(j)+sigmal(j-1)
      elseif(j.eq.ppeg)then
```

```
      sigmaf(j)=sigmal(j-1)
      sigmaf(j+1)=sigmar(j)+sigmal(j)
      sigmaf(j+2)=sigmar(j+1)
      elseif(j.gt.ppeg+1)then
      sigmaf(j+1)=sigmar(j)+sigmal(j-1)
      endif
      enddo
      sigmaf(IPM+2)=sigmal(IPM)
      sigabsf(1)=sigabsr(1)
      do j=2,IPM
      if(j.lt.ppeg)then
      sigabsf(j)=sigabsr(j)+sigabsl(j-1)
      elseif(j.eq.ppeg)then
      sigabsf(j)=sigabsl(j-1)
      sigabsf(j+1)=sigabsr(j)+sigabsl(j)
      sigabsf(j+2)=sigabsr(j+1)
      elseif(j.gt.ppeg+1)then
      sigabsf(j+1)=sigabsr(j)+sigabsl(j-1)
      endif
      enddo
      sigabsf(IPM+2)=sigabsl(IPM)
c    Reconstruct the new flux
      do i=1,IPM
      flx_l(i)=(CL0(i)*sigma(1,IS,i)+CL1(i)*sigma(6,IS,i))/100000.0
      flx_r(i)=(CR0(i)*sigma(1,IS,i)+CR1(i)*sigma(6,IS,i))/100000.0
      enddo

      flux(1)=flx_r(1)
      do j=2,IPM
      if(j.lt.ppeg)then
      flux(j)=flx_r(j)+flx_l(j-1)
      elseif(j.eq.ppeg)then
      flux(j)=flx_l(j-1)
      flux(j+1)=flx_r(j)+flx_l(j)
      flux(j+2)=flx_r(j+1)
      elseif(j.gt.ppeg+1)then
      flux(j+1)=flx_r(j)+flx_l(j-1)
      endif
      enddo
      flux(IPM+2)=flx_l(IPM)
      do j=1,IPM+2
      sigmaf(j)=sigmaf(j)/flux(j)
      enddo
      do j=1,IPM+2
      sigabsf(j)=sigabsf(j)/flux(j)
      enddo

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     scale absorption cross sections for calculation       c
c     for multigroup absorption calculation.                c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      do j=1,IPM
      if(j.lt.ppeg)then
      sigabsr(j)=sigabsr(j)/flux(j)
      sigabsl(j)=sigabsl(j)/flux(j+1)
      elseif(j.eq.ppeg)then
      sigabsr(j)=sigabsr(j)/flux(j+1)
      sigabsl(j)=sigabsl(j)/flux(j+1)
      elseif(j.gt.ppeg)then
      sigabsr(j)=sigabsr(j)/flux(j+1)
      sigabsl(j)=sigabsl(j)/flux(j+2)
      endif
      enddo

      open(25,file='absorption.inp',form='formatted',status='unknown')
      write(25,'(I10)')IPM
      do j=1,IPM
      write(25,'(I10,2E15.8)')j,sigabsl(j),sigabsr(j)
```

```
      enddo
      close(25)

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c        Calculate the new scattering matrix.                          c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

c       first set is right-right guide
c       second set is right left guide
c       third set is left right guide
c       fourth set is left left guide

  25  continue
      do i=1,ID1(6)+1
        do k=1,IPM
          do j=1,IPM
      if(k.lt.ppeg)then
        if(j.lt.ppeg)then
      tempo1=CR0(k)*CR0(j)*scatmat(1,i,j,k)+
     +  CR1(k)*CR0(j)*scatmat(2,i,j,k)+CR1(k)*CR1(j)*scatmat(4,i,j,k)+
     +  CR0(k)*CR1(j)*scatmat(5,i,j,k)
        scatfin(i,j,k)=scatfin(i,j,k)+tempo1
        tempo2=CR0(k)*CL0(j)*scatmat(1,i,j,k)+
     +  CR1(k)*CL0(j)*scatmat(2,i,j,k)+CR1(k)*CL1(j)*scatmat(4,i,j,k)+
     +  CR0(k)*CL1(j)*scatmat(5,i,j,k)
          scatfin(i,j+1,k)=scatfin(i,j+1,k)+tempo2
        tempo3=CL0(k)*CR0(j)*scatmat(1,i,j,k)+
     +  CL1(k)*CR0(j)*scatmat(2,i,j,k)+CL1(k)*CR1(j)*scatmat(4,i,j,k)+
     +  CL0(k)*CR1(j)*scatmat(5,i,j,k)
          scatfin(i,j,k+1)=scatfin(i,j,k+1)+tempo3
           tempo4=CL0(k)*CL0(j)*
     +  scatmat(1,i,j,k)+CL1(k)*CL0(j)*scatmat(2,i,j,k)+
     +  CL1(k)*CL1(j)*scatmat(4,i,j,k)+CL0(k)*CL1(j)*scatmat(5,i,j,k)
        scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo4
        elseif(j.eq.ppeg)then
         tempo1=CR0(k)*CR0(j)*scatmat(1,i,j,k)+
     +  CR1(k)*CR0(j)*scatmat(2,i,j,k)+CR1(k)*CR1(j)*scatmat(4,i,j,k)+
     +  CR0(k)*CR1(j)*scatmat(5,i,j,k)
        scatfin(i,j+1,k)=scatfin(i,j+1,k)+tempo1
        tempo2=CR0(k)*CL0(j)*scatmat(1,i,j,k)+
     +  CR1(k)*CL0(j)*scatmat(2,i,j,k)+CR1(k)*CL1(j)*scatmat(4,i,j,k)+
     +  CR0(k)*CL1(j)*scatmat(5,i,j,k)
          scatfin(i,j+1,k)=scatfin(i,j+1,k)+tempo2
        tempo3=CL0(k)*CR0(j)*scatmat(1,i,j,k)+
     +  CL1(k)*CR0(j)*scatmat(2,i,j,k)+CL1(k)*CR1(j)*scatmat(4,i,j,k)+
     +  CL0(k)*CR1(j)*scatmat(5,i,j,k)

          scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo3
           tempo4=CL0(k)*CL0(j)*
     +  scatmat(1,i,j,k)+CL1(k)*CL0(j)*scatmat(2,i,j,k)+
     +  CL1(k)*CL1(j)*scatmat(4,i,j,k)+CL0(k)*CL1(j)*scatmat(5,i,j,k)
       scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo4
          elseif(j.gt.ppeg)then
        tempo1=CR0(k)*CR0(j)*scatmat(1,i,j,k)+
     +  CR1(k)*CR0(j)*scatmat(2,i,j,k)+CR1(k)*CR1(j)*scatmat(4,i,j,k)+
     +  CR0(k)*CR1(j)*scatmat(5,i,j,k)
        scatfin(i,j+1,k)=scatfin(i,j+1,k)+tempo1
        tempo2=CR0(k)*CL0(j)*scatmat(1,i,j,k)+
     +  CR1(k)*CL0(j)*scatmat(2,i,j,k)+CR1(k)*CL1(j)*scatmat(4,i,j,k)+
     +  CR0(k)*CL1(j)*scatmat(5,i,j,k)
           scatfin(i,j+2,k)=scatfin(i,j+2,k)+tempo2
        tempo3=CL0(k)*CR0(j)*scatmat(1,i,j,k)+
     +  CL1(k)*CR0(j)*scatmat(2,i,j,k)+CL1(k)*CR1(j)*scatmat(4,i,j,k)+
     +  CL0(k)*CR1(j)*scatmat(5,i,j,k)
          scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo3
           tempo4=CL0(k)*CL0(j)*
     +  scatmat(1,i,j,k)+CL1(k)*CL0(j)*scatmat(2,i,j,k)+
     +  CL1(k)*CL1(j)*scatmat(4,i,j,k)+CL0(k)*CL1(j)*scatmat(5,i,j,k)
       scatfin(i,j+2,k+1)=scatfin(i,j+2,k+1)+tempo4
```

```
  endif
 elseif(k.eq.ppeg)then
 if(j.lt.ppeg)then
    tempo1=CR0(k)*CR0(j)*scatmat(1,i,j,k)+
+   CR1(k)*CR0(j)*scatmat(2,i,j,k)+CR1(k)*CR1(j)*scatmat(4,i,j,k)+
+   CR0(k)*CR1(j)*scatmat(5,i,j,k)
    scatfin(i,j,k+1)=scatfin(i,j,k+1)+tempo1
    tempo2=CR0(k)*CL0(j)*scatmat(1,i,j,k)+
+   CR1(k)*CL0(j)*scatmat(2,i,j,k)+CR1(k)*CL1(j)*scatmat(4,i,j,k)+
+   CR0(k)*CL1(j)*scatmat(5,i,j,k)
       scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo2
    tempo3=CL0(k)*CR0(j)*scatmat(1,i,j,k)+
+   CL1(k)*CR0(j)*scatmat(2,i,j,k)+CL1(k)*CR1(j)*scatmat(4,i,j,k)+
+   CL0(k)*CR1(j)*scatmat(5,i,j,k)
      scatfin(i,j,k+1)=scatfin(i,j,k+1)+tempo3
      tempo4=CL0(k)*CL0(j)*
+   scatmat(1,i,j,k)+CL1(k)*CL0(j)*scatmat(2,i,j,k)+
+   CL1(k)*CL1(j)*scatmat(4,i,j,k)+CL0(k)*CL1(j)*scatmat(5,i,j,k)
 scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo4
 elseif(j.eq.ppeg)then
    tempo1=CR0(k)*CR0(j)*scatmat(1,i,j,k)+
+   CR1(k)*CR0(j)*scatmat(2,i,j,k)+CR1(k)*CR1(j)*scatmat(4,i,j,k)+
+   CR0(k)*CR1(j)*scatmat(5,i,j,k)
    scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo1
    tempo2=CR0(k)*CL0(j)*scatmat(1,i,j,k)+
+   CR1(k)*CL0(j)*scatmat(2,i,j,k)+CR1(k)*CL1(j)*scatmat(4,i,j,k)+
+   CR0(k)*CL1(j)*scatmat(5,i,j,k)
     scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo2
    tempo3=CL0(k)*CR0(j)*scatmat(1,i,j,k)+
+   CL1(k)*CR0(j)*scatmat(2,i,j,k)+CL1(k)*CR1(j)*scatmat(4,i,j,k)+
+   CL0(k)*CR1(j)*scatmat(5,i,j,k)
      scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo3
       tempo4=CL0(k)*CL0(j)*
+   scatmat(1,i,j,k)+CL1(k)*CL0(j)*scatmat(2,i,j,k)+
+   CL1(k)*CL1(j)*scatmat(4,i,j,k)+CL0(k)*CL1(j)*scatmat(5,i,j,k)
 scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo4
 elseif(j.gt.ppeg)then
    tempo1=CR0(k)*CR0(j)*scatmat(1,i,j,k)+
+   CR1(k)*CR0(j)*scatmat(2,i,j,k)+CR1(k)*CR1(j)*scatmat(4,i,j,k)+
+   CR0(k)*CR1(j)*scatmat(5,i,j,k)
    scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo1
    tempo2=CR0(k)*CL0(j)*scatmat(1,i,j,k)+
+   CR1(k)*CL0(j)*scatmat(2,i,j,k)+CR1(k)*CL1(j)*scatmat(4,i,j,k)+
+   CR0(k)*CL1(j)*scatmat(5,i,j,k)
       scatfin(i,j+2,k+1)=scatfin(i,j+2,k+1)+tempo2
    tempo3=CL0(k)*CR0(j)*scatmat(1,i,j,k)+
+   CL1(k)*CR0(j)*scatmat(2,i,j,k)+CL1(k)*CR1(j)*scatmat(4,i,j,k)+
+   CL0(k)*CR1(j)*scatmat(5,i,j,k)
      scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo3
       tempo4=CL0(k)*CL0(j)*
+   scatmat(1,i,j,k)+CL1(k)*CL0(j)*scatmat(2,i,j,k)+
+   CL1(k)*CL1(j)*scatmat(4,i,j,k)+CL0(k)*CL1(j)*scatmat(5,i,j,k)
 scatfin(i,j+2,k+1)=scatfin(i,j+2,k+1)+tempo4
 endif
 elseif(k.gt.ppeg)then
 if(j.lt.ppeg)then
    tempo1=CR0(k)*CR0(j)*scatmat(1,i,j,k)+
+   CR1(k)*CR0(j)*scatmat(2,i,j,k)+CR1(k)*CR1(j)*scatmat(4,i,j,k)+
+   CR0(k)*CR1(j)*scatmat(5,i,j,k)
    scatfin(i,j,k+1)=scatfin(i,j,k+1)+tempo1
    tempo2=CR0(k)*CL0(j)*scatmat(1,i,j,k)+
+   CR1(k)*CL0(j)*scatmat(2,i,j,k)+CR1(k)*CL1(j)*scatmat(4,i,j,k)+
+   CR0(k)*CL1(j)*scatmat(5,i,j,k)
      scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo2
    tempo3=CL0(k)*CR0(j)*scatmat(1,i,j,k)+
+   CL1(k)*CR0(j)*scatmat(2,i,j,k)+CL1(k)*CR1(j)*scatmat(4,i,j,k)+
+   CL0(k)*CR1(j)*scatmat(5,i,j,k)
    scatfin(i,j,k+2)=scatfin(i,j,k+2)+tempo3
       tempo4=CL0(k)*CL0(j)*
```

```
     +  scatmat(1,i,j,k)+CL1(k)*CL0(j)*scatmat(2,i,j,k)+
     +  CL1(k)*CL1(j)*scatmat(4,i,j,k)+CL0(k)*CL1(j)*scatmat(5,i,j,k)
      scatfin(i,j+1,k+2)=scatfin(i,j+1,k+2)+tempo4
        elseif(j.eq.ppeg)then
        tempo1=CR0(k)*CR0(j)*scatmat(1,i,j,k)+
     +  CR1(k)*CR0(j)*scatmat(2,i,j,k)+CR1(k)*CR1(j)*scatmat(4,i,j,k)+
     +  CR0(k)*CR1(j)*scatmat(5,i,j,k)
        scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo1
        tempo2=CR0(k)*CL0(j)*scatmat(1,i,j,k)+
     +  CR1(k)*CL0(j)*scatmat(2,i,j,k)+CR1(k)*CL1(j)*scatmat(4,i,j,k)+
     +  CR0(k)*CL1(j)*scatmat(5,i,j,k)
           scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo2
        tempo3=CL0(k)*CR0(j)*scatmat(1,i,j,k)+
     +  CL1(k)*CR0(j)*scatmat(2,i,j,k)+CL1(k)*CR1(j)*scatmat(4,i,j,k)+
     +  CL0(k)*CR1(j)*scatmat(5,i,j,k)
          scatfin(i,j+1,k+2)=scatfin(i,j+1,k+1)+tempo3
           tempo4=CL0(k)*CL0(j)*
     +  scatmat(1,i,j,k)+CL1(k)*CL0(j)*scatmat(2,i,j,k)+
     +  CL1(k)*CL1(j)*scatmat(4,i,j,k)+CL0(k)*CL1(j)*scatmat(5,i,j,k)
      scatfin(i,j+1,k+2)=scatfin(i,j+1,k+2)+tempo4
      elseif(j.gt.ppeg)then
          tempo1=CR0(k)*CR0(j)*scatmat(1,i,j,k)+
     +  CR1(k)*CR0(j)*scatmat(2,i,j,k)+CR1(k)*CR1(j)*scatmat(4,i,j,k)+
     +  CR0(k)*CR1(j)*scatmat(5,i,j,k)
        scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo1
        tempo2=CR0(k)*CL0(j)*scatmat(1,i,j,k)+
     +  CR1(k)*CL0(j)*scatmat(2,i,j,k)+CR1(k)*CL1(j)*scatmat(4,i,j,k)+
     +  CR0(k)*CL1(j)*scatmat(5,i,j,k)
           scatfin(i,j+2,k+1)=scatfin(i,j+2,k+1)+tempo2

        tempo3=CL0(k)*CR0(j)*scatmat(1,i,j,k)+
     +  CL1(k)*CR0(j)*scatmat(2,i,j,k)+CL1(k)*CR1(j)*scatmat(4,i,j,k)+
     +  CL0(k)*CR1(j)*scatmat(5,i,j,k)
          scatfin(i,j+1,k+2)=scatfin(i,j+1,k+2)+tempo3
           tempo4=CL0(k)*CL0(j)*
     +  scatmat(1,i,j,k)+CL1(k)*CL0(j)*scatmat(2,i,j,k)+
     +  CL1(k)*CL1(j)*scatmat(4,i,j,k)+CL0(k)*CL1(j)*scatmat(5,i,j,k)
      scatfin(i,j+2,k+2)=scatfin(i,j+2,k+2)+tempo4
       endif
       endif
      enddo
      enddo
      enddo
      do j=1,IPM
      if(j.lt.ppeg)then
      scatfin(1,ppeg+1,j)=(CR0(j)*sigma(1,ISP,j)+
     +CR1(j)*sigma(2,ISP,j))+scatfin(1,ppeg+1,j)
      scatfin(1,ppeg+1,j)=(CR0(j)*sigma(1,ISP,j)
     ++CR1(j)*sigma(2,ISP,j))+scatfin(1,ppeg+1,j)
      scatfin(1,ppeg+1,j+1)=(CL0(j)*sigma(1,ISP,j)+
     +CL1(j)*sigma(2,ISP,j))+scatfin(1,ppeg+1,j+1)
      scatfin(1,ppeg+1,j+1)=(CL0(j)*sigma(1,ISP,j)+
     +CL1(j)*sigma(2,ISP,j))+scatfin(1,ppeg+1,j+1)
      elseif(j.eq.ppeg)then
c      scatfin(1,ppeg+1,j+1)=sigma(1,ISP,j)
      scatfin(1,ppeg+1,j+1)=(CR0(j)*sigma(1,ISP,j)+
     +CR1(j)*sigma(2,ISP,j))+scatfin(1,ppeg+1,j+1)
      scatfin(1,ppeg+1,j+1)=(CR0(j)*sigma(1,ISP,j)
     ++CR1(j)*sigma(2,ISP,j))+scatfin(1,ppeg+1,j+1)
      scatfin(1,ppeg+1,j+1)=(CL0(j)*sigma(1,ISP,j)+
     +CL1(j)*sigma(2,ISP,j))+scatfin(1,ppeg+1,j+1)
      scatfin(1,ppeg+1,j+1)=(CL0(j)*sigma(1,ISP,j)+
     +CL1(j)*sigma(2,ISP,j))+scatfin(1,ppeg+1,j+1)
      endif
      end do
      do j=1,IPM
      tempo1=(CR0(j)*sigma(1,ISP,j)+CR1(j)*sigma(2,ISP,j))
      tempo2=(CR0(j)*sigma(1,ISP,j)+CR1(j)*sigma(2,ISP,j))
      tempo3=(CL0(j)*sigma(1,ISP,j)+CL1(j)*sigma(2,ISP,j))
```

```
      tempo4=(CL0(j)*sigma(1,ISP,j)+CL1(j)*sigma(2,ISP,j))
      tempair1=tempair1+tempo1+tempo3
      enddo
      do i=1,ID1(6)+1
       do j=1,IPM+2
          do k=1,IPM+2
          scatfin(i,k,j)=scatfin(i,k,j)/flux(j)
        enddo
       enddo
      enddo
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c        Solve for an infinite Medium.                               c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
        SC(1:280)=0.0
        flux1(1:280)=0.0
        flux2(1:280)=0.0
        alpha=0.08
c     This is for reading fixed source term.
      open(30,file='c:\codes\shielding\MGpa1.inp',form='formatted',
     +status='old')
      read(30,'(a30)')
      read(30,'(I15)')N
       do j=1,N+2
       read(30,'(E18.7)')SC(j)
       write(22,'(E20.7)')SC(j)
       enddo
      element=3.29674E-2
       do j=1,IGM
       sigmaf(j)=sigmaf(j)*element
       enddo
       do j=1,IGM
       do i=1,IGM
       scatfin(1,i,j)=-element*scatfin(1,i,j)
       enddo
       enddo
       do j=1,IGM
       scatfin(1,j,j)=scatfin(1,j,j)+sigmaf(j)
       enddo
       count=1
       count2=1
  41   continue
       do i=1,IGM
       ssum=0.0
       do j=Bounds(1,i,1),Bounds(1,i,2)
       ssum=ssum+scatfin(1,i,j)*flux1(j)
       enddo
       flux2(i)=flux1(i)+alpha*(sc(i)-ssum)
       enddo
       err=abs(flux1(1)-flux2(1))
       meter=count/10000
       if(meter.eq.count2)then
       count2=count2+1
       endif
       do j=1,IGM
       flux1(j)=flux2(j)
       enddo
       count=count+1
       if (count.gt.1e4)alpha=0.2
       if(count.le.1E6)then
       goto 41
       endif
      open(40,file='c:\codes\shielding\MGpaflux.out',form='formatted',
     +status='unknown',position='append')
      do i=1,IGM
      write(40,'(2x,I5,4x,F15.7)')i,flux2(i)
     enddo
      do j=1,N
      if(j.lt.ppeg)then
      mgabs(j)=(flux2(j)*sigabsr(j)+flux2(j+1)*sigabsl(j))*element
```

```
      elseif(j.eq.ppeg)then
      mgabs(j)=(flux2(j+1)*sigabsr(j)+flux2(j+1)*sigabsl(j))*element
      elseif(j.gt.ppeg)then
      mgabs(j)=(flux2(j+1)*sigabsr(j)+flux2(j+2)*sigabsl(j))*element
      endif
      enddo
       k=N/4
       print*,N,K
       tabs=0.0
       do j=1,4
       do i=1,k
       absor(j)=absor(j)+mgabs((j-1)*k+i)
       enddo
       enddo
       tabs=0.0
       do j=1,4
       tabs=tabs+absor(j)
       enddo
       do j=1,4
      print*,j,absor(j)
       enddo
       print*,tabs
      open(45,file='c:\codes\shielding\MGpaabs.out',form='formatted',
     +status='unknown',position='append')
      write(45,'(2x,I5,5(4x,E15.7))')N,(absor(j),j=1,4),tabs
      close(45)
c  Reproduce the spectrum for hat functions
      open(32,file='c:\codes\shielding\engbd2.inp',form='formatted',
     +status='old')
      read(32,'(I10)')N
      do j=1,N+1
      read(32,'(E17.6)')EB(j)
      enddo
      close(32)
      open(34,file='c:\codes\shielding\mgpasp.out',form='formatted',
     +status='unknown',position='append')
      write(34,'(I10)')N
      open(35,file='c:\codes\shielding\New_spectrum.dat',
     +form='formatted',status='old')
      do j=1,2418
      read(35,'(2E18.7)')E(j),SP(j)
      enddo
       do j=1,2418
       do i=1,N
       if(E(j).gt.EB(i+1).and.E(j).le.EB(i))then
       if(i.lt.ppeg)then
       sp(j)=sp(j)*((CR0(i)+CR1(i)*E(j))*flux1(i)/flux(i)+
     +(CL0(i)+E(j)*CL1(i))*flux1(i+1)/flux(i+1))
       write(34,'(E17.6)')sp(j)
       elseif(i.eq.ppeg)then
       sp(j)=sp(j)*((CR0(N/2)+CR1(N/2)*E(j))*flux1(i+1)/flux(i+1)+
     +(CL0(i)+E(j)*CL1(i))*flux1(i+1)/flux(i+1))
       write(34,'(E17.6)')sp(j)
       elseif(i.gt.ppeg)then
       sp(j)=sp(j)*((CR0(i)+CR1(i)*E(j))*flux1(i+1)/flux(i+1)+
     +(CL0(i)+E(j)*CL1(i))*flux1(i+2)/flux(i+2))
       write(34,'(E17.6)')sp(j)
       endif
       endif
       enddo
       enddo
       close(22)
  30  continue
        stop
         end
```

Appendix D

Cross Sections Code for Hat Functions with MG Pair Production and MG Source Group

Appendix D contains the source code for the MGSOURCE code. This code is written for discrete energy source in the first energy group. It treats the first energy group and the pair production group as conventional MG. This code reads the six working libraries data for the six NJOY99 runs, and the coefficients for the hat basis functions. It then proceeds in calculating the group cross section for the LMG/MG method. Next, it reads the group sources from an input file generated by GUIDES code and solves the gamma spectrum problem in an infinite medium.

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c        Do all the guide functions.                                    c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      do j=1,IPM
      sigmar(j)=CR0(j)**2*SIGMA(1,IS1,j)+2*CR0(j)*CR1(j)*
     +SIGMA(2,IS1,j)+CR1(j)**2*SIGMA(3,IS1,j)
      sigmal(j)=CL0(j)**2*SIGMA(1,IS1,j)+2*CL0(j)*
     +CL1(j)*SIGMA(2,IS1,j)+CL1(j)**2*SIGMA(3,IS1,j)
      enddo
      do j=1,IPM
      if(j.lt.ppeg.and.j.eq.1)then
      scatfin(1,j,j)=-(CR0(j)*CL0(j)*SIGMA(1,IS1,j)+(CR1(j)*CL0(j)+
     +CR0(j)*CL1(j))*SIGMA(2,IS1,j)+CR1(j)*CL1(j)*SIGMA(3,IS1,j))
      scatfin(1,j,j)=2*scatfin(1,j,j)
      elseif(j.lt.ppeg.and.j.ne.1)then
      scatfin(1,j,j+1)=-(CR0(j)*CL0(j)*SIGMA(1,IS1,j)+(CR1(j)*CL0(j)+
     +CR0(j)*CL1(j))*SIGMA(2,IS1,j)+CR1(j)*CL1(j)*SIGMA(3,IS1,j))
      scatfin(1,j+1,j)=scatfin(1,j,j+1)
      elseif(j.eq.ppeg)then
      scatfin(1,j+1,j+1)=-(CR0(j)*CL0(j)*SIGMA(1,IS1,j)+(CR1(j)*CL0(j)+
     +CR0(j)*CL1(j))*SIGMA(2,IS1,j)+CR1(j)*CL1(j)*SIGMA(3,IS1,j))
      scatfin(1,j+1,j+1)=2*scatfin(1,j+1,j+1)
      elseif(j.gt.ppeg)then
      scatfin(1,j+1,j+2)=-(CR0(j)*CL0(j)*SIGMA(1,IS1,j)+(CR1(j)*CL0(j)+
     +CR0(j)*CL1(j))*SIGMA(2,IS1,j)+CR1(j)*CL1(j)*SIGMA(3,IS1,j))
      scatfin(1,j+2,j+1)=scatfin(1,j+1,j+2)
      endif
      enddo
      do j=1,IPM
      sigabsr(j)=CR0(j)*SIGMA(1,IS2,j)+CR1(j)*SIGMA(2,IS2,j)
      sigabsl(j)=CL0(j)*SIGMA(1,IS2,j)+CL1(j)*SIGMA(2,IS2,j)
      enddo
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c            Reconstruct the absoption cross sections                   c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c   reconstruct the new hat cross sections
      sigmaf(1)=sigmar(1)+sigmal(1)
      sigmaf(2)=sigmar(2)
      do j=3,IPM
      if(j.lt.ppeg)then
      sigmaf(j)=sigmar(j)+sigmal(j-1)
      elseif(j.eq.ppeg)then
      sigmaf(j)=sigmal(j-1)
      sigmaf(j+1)=sigmar(j)+sigmal(j)
      sigmaf(j+2)=sigmar(j+1)
      elseif(j.gt.ppeg+1)then
      sigmaf(j+1)=sigmar(j)+sigmal(j-1)
      endif
      enddo
      sigmaf(IPM+2)=sigmal(IPM)
      sigabsf(1)=sigabsr(1)+sigabsl(1)
      sigabsf(2)=sigabsr(2)
      do j=3,IPM
      if(j.lt.ppeg)then
      sigabsf(j)=sigabsr(j)+sigabsl(j-1)
      elseif(j.eq.ppeg)then
```

```
      sigabsf(j)=sigabsl(j-1)
      sigabsf(j+1)=sigabsr(j)+sigabsl(j)
      sigabsf(j+2)=sigabsr(j+1)
      elseif(j.gt.ppeg+1)then
      sigabsf(j+1)=sigabsr(j)+sigabsl(j-1)
      endif
      enddo
      sigabsf(IPM+2)=sigabsl(IPM)
c     Reconstruct the new flux
      do i=1,IPM
      flx_l(i)=(CL0(i)*sigma(1,IS,i)+CL1(i)*sigma(6,IS,i))/100000.0
      flx_r(i)=(CR0(i)*sigma(1,IS,i)+CR1(i)*sigma(6,IS,i))/100000.0
      enddo
      flux(1)=flx_r(1)+flx_l(1)
      flux(2)=flx_r(2)
      do j=3,IPM
      if(j.lt.ppeg)then
      flux(j)=flx_r(j)+flx_l(j-1)
      elseif(j.eq.ppeg)then
      flux(j)=flx_l(j-1)
      flux(j+1)=flx_r(j)+flx_l(j)
      flux(j+2)=flx_r(j+1)
      elseif(j.gt.ppeg+1)then
      flux(j+1)=flx_r(j)+flx_l(j-1)
      endif
      enddo
      flux(IPM+2)=flx_l(IPM)
      do j=1,IPM+2
      sigmaf(j)=sigmaf(j)/flux(j)
      sigabsf(j)=sigabsf(j)/flux(j)
      enddo
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     scale absorption cross sections for calculation       c
c     for multigroup absorption calculation.                c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      do j=1,IPM
      if(j.eq.1)then
      sigabsr(j)=sigabsr(j)/flux(j)
      sigabsl(j)=sigabsl(j)/flux(j)
      elseif(j.lt.ppeg.and.j.ne.1)then
      sigabsr(j)=sigabsr(j)/flux(j)
      sigabsl(j)=sigabsl(j)/flux(j+1)
      elseif(j.eq.ppeg)then
      sigabsr(j)=sigabsr(j)/flux(j+1)
      sigabsl(j)=sigabsl(j)/flux(j+1)
      elseif(j.gt.ppeg)then
      sigabsr(j)=sigabsr(j)/flux(j+1)
      sigabsl(j)=sigabsl(j)/flux(j+2)
      endif
      enddo
      open(25,file='absorption.inp',form='formatted',status='unknown')
      write(25,'(I10)')IPM
      do j=1,IPM
      write(25,'(I10,2E15.8)')j,sigabsl(j),sigabsr(j)
      enddo
      close(25)
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c        Calculate the new scattering matrix.                    c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
  25  continue
      do i=1,ID1(6)+1
        do k=1,IPM
          do j=1,IPM
      if(k.eq.1)then
        if(j.eq.1)then
        scatfin(i,1,1)=scatfin(i,1,1)+scatmat(1,i,1,1)
        elseif(j.lt.ppeg.and.j.ne.1)then
      tempo1=CR0(j)*scatmat(1,i,j,k)+CR1(j)*scatmat(5,i,j,k)
        scatfin(i,j,k)=scatfin(i,j,k)+tempo1
```

```
   tempo4=CL0(j)*scatmat(1,i,j,k)+CL1(j)*scatmat(5,i,j,k)
   scatfin(i,j+1,k)=scatfin(i,j+1,k)+tempo4
   elseif(j.eq.ppeg)then
    tempo1=scatmat(1,i,j,k)
   scatfin(i,j+1,k)=scatfin(i,j+1,k)+tempo1
     elseif(j.gt.ppeg)then
  tempo1=CR0(j)*scatmat(1,i,j,k)+CR1(j)*scatmat(5,i,j,k)
   scatfin(i,j+1,k)=scatfin(i,j+1,k)+tempo1
   tempo2=CL0(j)*scatmat(1,i,j,k)+CL1(j)*scatmat(5,i,j,k)
      scatfin(i,j+2,k)=scatfin(i,j+2,k)+tempo2
 endif
 elseif(k.lt.ppeg.and.k.gt.1)then
   if(j.lt.ppeg)then
 tempo1=CR0(k)*CR0(j)*scatmat(1,i,j,k)+
+  CR1(k)*CR0(j)*scatmat(2,i,j,k)+CR1(k)*CR1(j)*scatmat(4,i,j,k)+
+  CR0(k)*CR1(j)*scatmat(5,i,j,k)
   scatfin(i,j,k)=scatfin(i,j,k)+tempo1
   tempo2=CR0(k)*CL0(j)*scatmat(1,i,j,k)+
+  CR1(k)*CL0(j)*scatmat(2,i,j,k)+CR1(k)*CL1(j)*scatmat(4,i,j,k)+
+  CR0(k)*CL1(j)*scatmat(5,i,j,k)
      scatfin(i,j+1,k)=scatfin(i,j+1,k)+tempo2
   tempo3=CL0(k)*CR0(j)*scatmat(1,i,j,k)+
+  CL1(k)*CR0(j)*scatmat(2,i,j,k)+CL1(k)*CR1(j)*scatmat(4,i,j,k)+
+  CL0(k)*CR1(j)*scatmat(5,i,j,k)
     scatfin(i,j,k+1)=scatfin(i,j,k+1)+tempo3
      tempo4=CL0(k)*CL0(j)*
+  scatmat(1,i,j,k)+CL1(k)*CL0(j)*scatmat(2,i,j,k)+
+  CL1(k)*CL1(j)*scatmat(4,i,j,k)+CL0(k)*CL1(j)*scatmat(5,i,j,k)
   scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo4
   elseif(j.eq.ppeg)then
    tempo1=CR0(k)*CR0(j)*scatmat(1,i,j,k)+
+  CR1(k)*CR0(j)*scatmat(2,i,j,k)+CR1(k)*CR1(j)*scatmat(4,i,j,k)+
+  CR0(k)*CR1(j)*scatmat(5,i,j,k)
   scatfin(i,j+1,k)=scatfin(i,j+1,k)+tempo1
   tempo2=CR0(k)*CL0(j)*scatmat(1,i,j,k)+
+  CR1(k)*CL0(j)*scatmat(2,i,j,k)+CR1(k)*CL1(j)*scatmat(4,i,j,k)+
      scatfin(i,j+1,k)=scatfin(i,j+1,k)+tempo2
   tempo3=CL0(k)*CR0(j)*scatmat(1,i,j,k)+
+  CL1(k)*CR0(j)*scatmat(2,i,j,k)+CL1(k)*CR1(j)*scatmat(4,i,j,k)+
+  CL0(k)*CR1(j)*scatmat(5,i,j,k)
     scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo3
      tempo4=CL0(k)*CL0(j)*
+  scatmat(1,i,j,k)+CL1(k)*CL0(j)*scatmat(2,i,j,k)+
+  CL1(k)*CL1(j)*scatmat(4,i,j,k)+CL0(k)*CL1(j)*scatmat(5,i,j,k)
 scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo4
    elseif(j.gt.ppeg)then
  tempo1=CR0(k)*CR0(j)*scatmat(1,i,j,k)+
+  CR1(k)*CR0(j)*scatmat(2,i,j,k)+CR1(k)*CR1(j)*scatmat(4,i,j,k)+
+  CR0(k)*CR1(j)*scatmat(5,i,j,k)
   scatfin(i,j+1,k)=scatfin(i,j+1,k)+tempo1
   tempo2=CR0(k)*CL0(j)*scatmat(1,i,j,k)+
+  CR1(k)*CL0(j)*scatmat(2,i,j,k)+CR1(k)*CL1(j)*scatmat(4,i,j,k)+
+  CR0(k)*CL1(j)*scatmat(5,i,j,k)
      scatfin(i,j+2,k)=scatfin(i,j+2,k)+tempo2
   tempo3=CL0(k)*CR0(j)*scatmat(1,i,j,k)+
+  CL1(k)*CR0(j)*scatmat(2,i,j,k)+CL1(k)*CR1(j)*scatmat(4,i,j,k)+
+  CL0(k)*CR1(j)*scatmat(5,i,j,k)
     scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo3
      tempo4=CL0(k)*CL0(j)*
+  scatmat(1,i,j,k)+CL1(k)*CL0(j)*scatmat(2,i,j,k)+
+  CL1(k)*CL1(j)*scatmat(4,i,j,k)+CL0(k)*CL1(j)*scatmat(5,i,j,k)
 scatfin(i,j+2,k+1)=scatfin(i,j+2,k+1)+tempo4
 endif
 elseif(k.eq.ppeg)then
 if(j.lt.ppeg)then
   tempo1=CR0(k)*CR0(j)*scatmat(1,i,j,k)+
+  CR1(k)*CR0(j)*scatmat(2,i,j,k)+CR1(k)*CR1(j)*scatmat(4,i,j,k)+
+  CR0(k)*CR1(j)*scatmat(5,i,j,k)
   scatfin(i,j,k+1)=scatfin(i,j,k+1)+tempo1
```

```
   tempo2=CR0(k)*CL0(j)*scatmat(1,i,j,k)+
+  CR1(k)*CL0(j)*scatmat(2,i,j,k)+CR1(k)*CL1(j)*scatmat(4,i,j,k)+
+  CR0(k)*CL1(j)*scatmat(5,i,j,k)
      scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo2
   tempo3=CL0(k)*CR0(j)*scatmat(1,i,j,k)+
+  CL1(k)*CR0(j)*scatmat(2,i,j,k)+CL1(k)*CR1(j)*scatmat(4,i,j,k)+
+  CL0(k)*CR1(j)*scatmat(5,i,j,k)
     scatfin(i,j,k+1)=scatfin(i,j,k+1)+tempo3
      tempo4=CL0(k)*CL0(j)*
+  scatmat(1,i,j,k)+CL1(k)*CL0(j)*scatmat(2,i,j,k)+
+  CL1(k)*CL1(j)*scatmat(4,i,j,k)+CL0(k)*CL1(j)*scatmat(5,i,j,k)
 scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo4
 elseif(j.eq.ppeg)then
   tempo1=CR0(k)*CR0(j)*scatmat(1,i,j,k)+
+  CR1(k)*CR0(j)*scatmat(2,i,j,k)+CR1(k)*CR1(j)*scatmat(4,i,j,k)+
+  CR0(k)*CR1(j)*scatmat(5,i,j,k)
   scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo1
   tempo2=CR0(k)*CL0(j)*scatmat(1,i,j,k)+
+  CR1(k)*CL0(j)*scatmat(2,i,j,k)+CR1(k)*CL1(j)*scatmat(4,i,j,k)+
+  CR0(k)*CL1(j)*scatmat(5,i,j,k)
      scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo2
   tempo3=CL0(k)*CR0(j)*scatmat(1,i,j,k)+
+  CL1(k)*CR0(j)*scatmat(2,i,j,k)+CL1(k)*CR1(j)*scatmat(4,i,j,k)+
+  CL0(k)*CR1(j)*scatmat(5,i,j,k)
     scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo3
      tempo4=CL0(k)*CL0(j)*
+  scatmat(1,i,j,k)+CL1(k)*CL0(j)*scatmat(2,i,j,k)+
+  CL1(k)*CL1(j)*scatmat(4,i,j,k)+CL0(k)*CL1(j)*scatmat(5,i,j,k)
 scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo4
 elseif(j.gt.ppeg)then
   tempo1=CR0(k)*CR0(j)*scatmat(1,i,j,k)+
+  CR1(k)*CR0(j)*scatmat(2,i,j,k)+CR1(k)*CR1(j)*scatmat(4,i,j,k)+
+  CR0(k)*CR1(j)*scatmat(5,i,j,k)
   scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo1
   tempo2=CR0(k)*CL0(j)*scatmat(1,i,j,k)+
+  CR1(k)*CL0(j)*scatmat(2,i,j,k)+CR1(k)*CL1(j)*scatmat(4,i,j,k)+
+  CR0(k)*CL1(j)*scatmat(5,i,j,k)
      scatfin(i,j+2,k+1)=scatfin(i,j+2,k+1)+tempo2
   tempo3=CL0(k)*CR0(j)*scatmat(1,i,j,k)+
+  CL1(k)*CR0(j)*scatmat(2,i,j,k)+CL1(k)*CR1(j)*scatmat(4,i,j,k)+
+  CL0(k)*CR1(j)*scatmat(5,i,j,k)
     scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo3
      tempo4=CL0(k)*CL0(j)*
+  scatmat(1,i,j,k)+CL1(k)*CL0(j)*scatmat(2,i,j,k)+
+  CL1(k)*CL1(j)*scatmat(4,i,j,k)+CL0(k)*CL1(j)*scatmat(5,i,j,k)
 scatfin(i,j+2,k+1)=scatfin(i,j+2,k+1)+tempo4
 endif
 elseif(k.gt.ppeg)then
 if(j.lt.ppeg)then
   tempo1=CR0(k)*CR0(j)*scatmat(1,i,j,k)+
+  CR1(k)*CR0(j)*scatmat(2,i,j,k)+CR1(k)*CR1(j)*scatmat(4,i,j,k)+
+  CR0(k)*CR1(j)*scatmat(5,i,j,k)
   scatfin(i,j,k+1)=scatfin(i,j,k+1)+tempo1
   tempo2=CR0(k)*CL0(j)*scatmat(1,i,j,k)+
+  CR1(k)*CL0(j)*scatmat(2,i,j,k)+CR1(k)*CL1(j)*scatmat(4,i,j,k)+
+  CR0(k)*CL1(j)*scatmat(5,i,j,k)
      scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo2
   tempo3=CL0(k)*CR0(j)*scatmat(1,i,j,k)+
+  CL1(k)*CR0(j)*scatmat(2,i,j,k)+CL1(k)*CR1(j)*scatmat(4,i,j,k)+
+  CL0(k)*CR1(j)*scatmat(5,i,j,k)
     scatfin(i,j,k+2)=scatfin(i,j,k+2)+tempo3
      tempo4=CL0(k)*CL0(j)*
+  scatmat(1,i,j,k)+CL1(k)*CL0(j)*scatmat(2,i,j,k)+
+  CL1(k)*CL1(j)*scatmat(4,i,j,k)+CL0(k)*CL1(j)*scatmat(5,i,j,k)
 scatfin(i,j+1,k+2)=scatfin(i,j+1,k+2)+tempo4
   elseif(j.eq.ppeg)then
   tempo1=CR0(k)*CR0(j)*scatmat(1,i,j,k)+
+  CR1(k)*CR0(j)*scatmat(2,i,j,k)+CR1(k)*CR1(j)*scatmat(4,i,j,k)+
+  CR0(k)*CR1(j)*scatmat(5,i,j,k)
```

```
   scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo1
   tempo2=CR0(k)*CL0(j)*scatmat(1,i,j,k)+
+  CR1(k)*CL0(j)*scatmat(2,i,j,k)+CR1(k)*CL1(j)*scatmat(4,i,j,k)+
+  CR0(k)*CL1(j)*scatmat(5,i,j,k)
      scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo2
   tempo3=CL0(k)*CR0(j)*scatmat(1,i,j,k)+
+  CL1(k)*CR0(j)*scatmat(2,i,j,k)+CL1(k)*CR1(j)*scatmat(4,i,j,k)+
+  CL0(k)*CR1(j)*scatmat(5,i,j,k)
     scatfin(i,j+1,k+2)=scatfin(i,j+1,k+1)+tempo3
      tempo4=CL0(k)*CL0(j)*
+  scatmat(1,i,j,k)+CL1(k)*CL0(j)*scatmat(2,i,j,k)+
+  CL1(k)*CL1(j)*scatmat(4,i,j,k)+CL0(k)*CL1(j)*scatmat(5,i,j,k)
 scatfin(i,j+1,k+2)=scatfin(i,j+1,k+2)+tempo4
 elseif(j.gt.ppeg)then
     tempo1=CR0(k)*CR0(j)*scatmat(1,i,j,k)+
+  CR1(k)*CR0(j)*scatmat(2,i,j,k)+CR1(k)*CR1(j)*scatmat(4,i,j,k)+
+  CR0(k)*CR1(j)*scatmat(5,i,j,k)
   scatfin(i,j+1,k+1)=scatfin(i,j+1,k+1)+tempo1
   tempo2=CR0(k)*CL0(j)*scatmat(1,i,j,k)+
+  CR1(k)*CL0(j)*scatmat(2,i,j,k)+CR1(k)*CL1(j)*scatmat(4,i,j,k)+
+  CR0(k)*CL1(j)*scatmat(5,i,j,k)
      scatfin(i,j+2,k+1)=scatfin(i,j+2,k+1)+tempo2
   tempo3=CL0(k)*CR0(j)*scatmat(1,i,j,k)+
+  CL1(k)*CR0(j)*scatmat(2,i,j,k)+CL1(k)*CR1(j)*scatmat(4,i,j,k)+
+  CL0(k)*CR1(j)*scatmat(5,i,j,k)
     scatfin(i,j+1,k+2)=scatfin(i,j+1,k+2)+tempo3
      tempo4=CL0(k)*CL0(j)*
+  scatmat(1,i,j,k)+CL1(k)*CL0(j)*scatmat(2,i,j,k)+
+  CL1(k)*CL1(j)*scatmat(4,i,j,k)+CL0(k)*CL1(j)*scatmat(5,i,j,k)
 scatfin(i,j+2,k+2)=scatfin(i,j+2,k+2)+tempo4
  endif
  endif
 enddo
 enddo
 enddo
 do j=1,IPM
 if(j.eq.1)then
 scatfin(1,ppeg+1,j)=2*sigma(1,ISP,j)
 elseif(j.lt.ppeg.and.j.ne.1)then
 scatfin(1,ppeg+1,j)=(CR0(j)*sigma(1,ISP,j)+
+CR1(j)*sigma(2,ISP,j))+scatfin(1,ppeg+1,j)
 scatfin(1,ppeg+1,j)=(CR0(j)*sigma(1,ISP,j)
++CR1(j)*sigma(2,ISP,j))+scatfin(1,ppeg+1,j)
 scatfin(1,ppeg+1,j+1)=(CL0(j)*sigma(1,ISP,j)+
+CL1(j)*sigma(2,ISP,j))+scatfin(1,ppeg+1,j+1)
 scatfin(1,ppeg+1,j+1)=(CL0(j)*sigma(1,ISP,j)+
+CL1(j)*sigma(2,ISP,j))+scatfin(1,ppeg+1,j+1)
 elseif(j.eq.ppeg)then
 scatfin(1,ppeg+1,j+1)=(CR0(j)*sigma(1,ISP,j)+
+CR1(j)*sigma(2,ISP,j))+scatfin(1,ppeg+1,j+1)
 scatfin(1,ppeg+1,j+1)=(CR0(j)*sigma(1,ISP,j)
++CR1(j)*sigma(2,ISP,j))+scatfin(1,ppeg+1,j+1)
 scatfin(1,ppeg+1,j+1)=(CL0(j)*sigma(1,ISP,j)+
+CL1(j)*sigma(2,ISP,j))+scatfin(1,ppeg+1,j+1)
 scatfin(1,ppeg+1,j+1)=(CL0(j)*sigma(1,ISP,j)+
+CL1(j)*sigma(2,ISP,j))+scatfin(1,ppeg+1,j+1)
 endif
 end do
 do j=1,IPM
 tempo1=ppf*2.0*(CR0(j)*sigma(1,ISP,j)+CR1(j)*sigma(2,ISP,j))
 tempo2=(1.-ppf)*2.0*(CR0(j)*sigma(1,ISP,j)+CR1(j)*sigma(2,ISP,j))
 tempo3=ppf*2.0*(CL0(j)*sigma(1,ISP,j)+CL1(j)*sigma(2,ISP,j))
 tempo4=(1.-ppf)*2.0*(CL0(j)*sigma(1,ISP,j)+CL1(j)*sigma(2,ISP,j))
 enddo
 do i=1,ID1(6)+1
  do j=1,IPM+2
     do k=1,IPM+2
     scatfin(i,k,j)=scatfin(i,k,j)/flux(j)
   enddo
```

```
       enddo
      enddo
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c        Solve for an infinite Medium.                             c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
        SC(1:390)=0.0
        flux1(1:390)=0.0
        flux2(1:390)=0.0
        alpha=0.01
c     This is for reading fixed source term.
      open(30,file='c:\codes\shielding\MGpa1.inp',form='formatted',
     +status='old')
       read(30,'(a30)')
       read(30,'(I15)')N
c       do j=1,N+2
c       read(30,'(E18.7)')SC(j)
c       enddo
      element=3.29674E-2
       do j=1,IGM
       sigmaf(j)=sigmaf(j)*element
       enddo
       do j=1,IGM
       do i=1,IGM
       scatfin(1,i,j)=-element*scatfin(1,i,j)
       enddo
       enddo
       do j=1,IGM
       scatfin(1,j,j)=scatfin(1,j,j)+sigmaf(j)
       enddo
       count=1
       count2=1
  41   continue
       do i=1,IGM
       ssum=0.0
       do j=Bounds(1,i,1),Bounds(1,i,2)
       ssum=ssum+scatfin(1,i,j)*flux1(j)
       enddo
       flux2(i)=flux1(i)+alpha*(sc(i)-ssum)
       enddo
       err=abs(flux1(1)-flux2(1))
       meter=count/10000
       if(meter.eq.count2)then
       print*,flux1(1),flux1(2),flux1(3),err
       count2=count2+1
       endif
       do j=1,IGM
       flux1(j)=flux2(j)
       enddo
       count=count+1
       if (count.gt.1e4)alpha=0.15
       if(count.le.1E6)then
       goto 41
       endif
      do j=1,N
      if(j.eq.1)then
      mgabs(j)=(flux2(j)*sigabsr(j)+flux2(j)*sigabsl(j))*element
      elseif(j.lt.ppeg.and.j.gt.1)then
      mgabs(j)=(flux2(j)*sigabsr(j)+flux2(j+1)*sigabsl(j))*element
      elseif(j.eq.ppeg)then
      mgabs(j)=(flux2(j+1)*sigabsr(j)+flux2(j+1)*sigabsl(j))*element
      elseif(j.gt.ppeg)then
      mgabs(j)=(flux2(j+1)*sigabsr(j)+flux2(j+2)*sigabsl(j))*element
      endif
      enddo
      open(40,file='c:\codes\shielding\MGsorflx.out',form='formatted',
     +status='unknown',position='append')
      do i=1,IGM
      write(40,'(2x,I5,4x,F15.7)')i,flux2(i)
      enddo
```

```
      close(40)
       k=N/4
       tabs=0.0
       do j=1,8
       absor(j)=0.0
       enddo
       do j=1,4
       do i=1,k
       absor(j)=absor(j)+mgabs((j-1)*k+i)
       enddo
       enddo
       tabs=0.0
       do j=1,8
       tabs=tabs+absor(j)
       enddo
       do j=1,8
      print*,j,absor(j)
       enddo
       print*,tabs
      open(45,file='c:\codes\shielding\MGsorabs.out',form='formatted',
     +status='unknown',position='append')
      write(45,'(2x,I5,5(2x,E12.6))')N,(absor(j),j=1,4),tabs
      close(45)
c  Reproduce the spectrum for hat functions
      open(32,file='c:\codes\shielding\engbd2.inp',form='formatted',
     +status='old')
      read(32,'(I10)')N
      do j=1,N+1
      read(32,'(E17.6)')EB(j)
      enddo
      close(32)
      open(34,file='c:\codes\shielding\mgsourcesp.out',form='formatted',
     +status='unknown',position='append')
      write(34,'(I10)')N
      open(35,file='c:\codes\shielding\New_spectrum.dat',
     +form='formatted',status='old')
      do j=1,2418
      read(35,'(2E18.7)')E(j),SP(j)
      enddo
       do j=1,2418
       do i=1,N
       if(E(j).gt.EB(i+1).and.E(j).le.EB(i))then
       if(i.eq.1)then
       sp(j)=sp(j)*flux1(i)/flux(i)
       write(34,'(E17.6)')sp(j)
       elseif(i.lt.ppeg.and.i.ne.1)then
       sp(j)=sp(j)*((CR0(i)+CR1(i)*E(j))*flux1(i)/flux(i)+
     +(CL0(i)+E(j)*CL1(i))*flux1(i+1)/flux(i+1))
       write(34,'(E17.6)')sp(j)
       elseif(i.eq.ppeg)then
       sp(j)=sp(j)*((CR0(N/2)+CR1(N/2)*E(j))*flux1(i+1)/flux(i+1)+
     +(CL0(i)+E(j)*CL1(i))*flux1(i+1)/flux(i+1))
       write(34,'(E17.6)')sp(j)
       elseif(i.gt.ppeg)then
       sp(j)=sp(j)*((CR0(i)+CR1(i)*E(j))*flux1(i+1)/flux(i+1)+
     +(CL0(i)+E(j)*CL1(i))*flux1(i+2)/flux(i+2))
       write(34,'(E17.6)')sp(j)
       endif
       endif
       enddo
       enddo
       close(22)
  30  continue
        stop
         end
```

Appendix E

Code for Reading Flux Data and Absorption Cross Sections from SCALE Output

Appendix E contains the source code for the WRKLIBFLUX code. Since the conventional multigroup method was benchmarked using the SCALE system from Oak Ridge National Laboratory, the calculation of absorption in the four original energy groups had to be carried out by reading the absorption cross sections and group fluxes from SCALE's output. This code is written to do these tasks and reproduce the calculated spectrum.

```
      program main
      integer IDTAPE,NNUC,IGM,IFTG,MSN,ID1(10),ID3(3),ID5(7),ID7(5)
      integer ID9(1),M,count
      integer MT(20),MAGIC(20,390)
      integer NZ(10),z,MT2D(40),L(100),NT(100)
      real EB(390),UB(390),ID2(2),ID4(2),ID6(1)
      real SIGMA(6,20,390),sigmar(390),sigmal(390),sigmac(390)
      real scatfin(10,390,390),sigmaf(390),sigmat(390)
      real SCATMAT(6,10,390,390),MTS(10),orgscat(10,390,390)
      real temscat1(390),temscat2(390),mgabs(390)
      real flx_l(390),flx_r(390),flux(390),tflux(390)
      real flux1(390),flux2(390),sumflx(390),absor(390)
      integer Strngmat(20,390),Strnglen(20)
      integer BOUNDS(20,390,2),Sum(390),tsum(20),ppeg
      real d(39000),CL0(390),CL1(390),CR0(390),CR1(390)
      real sigabst(390),sigabsl(390),sigabsr(390),sigabsc(390)
      real sigpairl(390),sigpairr(390),sigpairf(390)
      real sigabsf(390),TEMPMT(2000),ppf,SC(390),E(1243),SP(1243)
      real dd(10,39000),totscat(390),MAT(390,390),element
      character*4 TITLE(18),ID8(1)
      character*400 tset
      character*30 coftitle

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     Read the working library after being processed by         c
c     Nitwal.                                                   c
c     Read the guide functions coefficients from file           c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      open(22,file='dumout.out',form='formatted',status='old')
      open(20,file='coeff.inp',form='formatted',status='old')
        read(20,'(a30)')coftitle
        read(20,'(I10)')N
        do j=1,N
       read(20,'(I5,4(1x,E15.8))')KN,CL0(j),CL1(j),CR0(j),CR1(j)
       enddo
        read(20,'(a30)')coftitle
        read(20,'(I5)')ppeg
        read(20,'(a30)')coftitle
        read(20,'(1x,E15.8)')ppf
        close(20)
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     open all three files for the diagonal terms that do not     c
c     have scattering terms.                                      c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      SCATMAT(1:6,1:10,1:390,1:390)=0.0
      orgscat(1:10,1:390,1:390)=0.0
      open(1,file='c:\scale4.4a\tmpdir\ft04f001',form='unformatted',
     +status='old')
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     read first header that has general information about       c
c     the data.                                                 c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      read(jj)IDTAPE,NNUC,IGM,IFTG,MSN,IPM,I1,I2,I3,I4,
     +tset
```

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c  read energy groups boundaries if it is a neutron problem or  c
c  gamma problem.                                               c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
       read(jj)(EB(I),I=1,IPM+1),(UB(I),I=1,IPM+1)
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     Read the fifty word title card.                           c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      read(jj)TITLE,ID1,ID2,ID3,ID4,ID5,ID6,ID7,ID8,ID9
      read(jj)TITLE,ID1,ID2,ID3,ID4,ID5,ID6,ID7,ID8,ID9
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     Read temperature independent cross sections for gamma rays.  c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      if(ID5(6).ne.0)then
        read(jj)(MTS(I),(SIGMA(jj,I,J),J=1,IPM),I=1,ID5(6))
      end if
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     retrieve the  scattering matrix terms and unscramble them.    c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      do j=1,ID1(6)+1
      read(jj)L(j),(d(i),i=1,L(j))
      mw=iset(d(1))
      call unscramble(d,L(j),scatmat,IPM,6,10,j,jj)
      enddo
      enddo
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c         Data up to this point has been read. The next     c
c         step is "unscale" the data.                       c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
           do j=1,ID5(6)
             if(MTS(j).eq.1599.0)then
             IS=j
             endif
           enddo
        do j=1,ID5(6)
        if(MTS(j).eq.516.0)then
        ISP=j
        endif
        enddo
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     Start Scaling the scalar processes.                       c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      open(80,file='groupflxs.inp',form='formatted',status='unknown')
            do i=1,IPM
               WRITE(80,'(E14.7)')SIGMA(1,IS,i)/100000.0
            enddo
       CLOSE(80)
        if(ID5(6).ne.0)then
           do jz=1,6
           do j=1,ID5(6)
             if(j.ne.IS)then
               do i=1,IPM
                SIGMA(jz,j,i)=SIGMA(jz,j,i)*SIGMA(jz,IS,i)/100000.0
               enddo
             endif
          enddo
          enddo
        endif
       endif
        stop
        end
```

Appendix F

Code for Writing NJOY99 Input Deck

Appendix F contains the source code for the WSHEILD code, which writes the input for NJOY99 to process the cross sections. It writes the input deck such that the pair production scattering cross sections can be included or excluded from the processing. This is needed since the pair production cross section needed special treatment.

```
C     Last change:  LFM   7 Aug 2003   12:45 pm

      INTEGER N,model,option
      REAL*8 ebound(1100)
      CHARACTER *80 line

      open(3,file='c:\codes\shielding\engbd.inp',form='formatted',
     +status='old')
      read(3,'(a30)')
      read(3,'(I10)')N
      do j=1,N+1
      read(3,'(1pE15.6)')ebound(j)
      enddo

      open(1,file='c:\codes\shielding\energy.inp',form='formatted',
     +status='old')
      read(1,'(a80)')line
      READ(1,'(24x,I10)')option
      read(1,'(24x,I10)')N
      print*,'Number of Energy Groups  ',N
      READ(1,'(24x,I10)')model

      IF(option.eq.5)then
      N=N+2
      endif

      open(2,file='c:\codes\shielding\shield.dat',form='formatted',
     +status='unknown')
      WRITE(2,'(a6)')'gaminr'
      WRITE(2,'(a11)')'30 31 0 32'
      WRITE(2,'(a12)')'800 1 3 3 1'
      WRITE(2,'(a30)')'''Lead Shielding Problem flat'''
      WRITE(2,'(I15)')N
      do j=1,N+1
      write(2,'(1pE15.6)')ebound(j)
      end do
      WRITE(2,'("23 501  ",a7)')'''total'''
      WRITE(2,'("23 522  ",\,a22)') '''Photoelectric Effect'''
      WRITE(2,'("23 516  ",\,a17)') '''pair production'''
      WRITE(2,'("26 502  ",\,a10)') '''coherent'''
      WRITE(2,'("26 504  ",\,a12)') '''incoherent'''
      if (model.eq.1) then
      WRITE(2,'("26 516  ",\,a17)')'''pair production'''
      end if
      WRITE(2,'("0/")')
      WRITE(2,'("0/")')
      WRITE(2,'("stop")')
      stop
      end
```

Vita

Ibrahim Attieh was born and raised in Saida, Lebanon. He immigrated to Canada with his family at age sixteen to start a new life there. He finished his high school education at Westdale High school in Hamilton, Ontario, Canada. He then pursued a Bachelor of Engineering Degree (Engineering Physics) at MacMaster University, also in Hamilton.

After graduation from university, Mr. Attieh worked first as a Graduate Engineer in Training at the Nuclear Branch of Ontario Hydro, and two years later as an assistant engineer. He worked as a commissioning engineer at Darlington Nuclear Generating Station. He commissioned a number of nuclear and conventional systems, including the containment building of Unit 4.

Mr. Attieh left Ontario Hydro to pursue graduate studies. He obtained a Master of Engineering degree in mechanical engineering from the University of Toronto 1997. In 1998, he joined the Nuclear Engineering department at the University of Tennessee to pursue a Ph.D. in nuclear engineering. He received the Ph.D. degree in May 2004.

1643 4866 44
11/03/04