

Multigroup Cross Section Generation Via  
Monte Carlo Methods

by

Everett Lee Redmond II

B.S., Nuclear Engineering  
Massachusetts Institute of Technology, 1990

M.S., Nuclear Engineering  
Massachusetts Institute of Technology, 1990

Submitted to the Department of Nuclear Engineering  
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY  
at the  
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 1997

Copyright © 1997 Massachusetts Institute of Technology.  
All rights reserved.

Signature of Author: \_\_\_\_\_  
Department of Nuclear Engineering  
May 12, 1997

Certified by: \_\_\_\_\_  
Allan F. Henry, Thesis Supervisor  
Professor of Nuclear Engineering

\_\_\_\_\_  
Jacquelyn C. Yanch, Thesis Reader  
Associate Professor of Nuclear Engineering

Accepted by: \_\_\_\_\_  
Jeffrey P. Freidberg  
Chairman, Department Committee on Graduate Students

MASSACHUSETTS INSTITUTE  
OF TECHNOLOGY

JUL 10 1997 ARCHIVES

LIBRARIES

# Multigroup Cross Section Generation Via Monte Carlo Methods

by  
Everett Lee Redmond II

Submitted to the Department of Nuclear Engineering on May 12, 1997 in  
partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

## ABSTRACT

Monte Carlo methods of performing radiation transport calculations are heavily used in many different applications. However, despite their prevalence, Monte Carlo codes do not eliminate the need for other methods of analysis like discrete ordinates transport codes or even diffusion theory codes. For example: current Monte Carlo codes are not capable of performing transient analysis or continuous energy adjoint calculations.

One of the primary difficulties, however, in using non-continuous energy methods is the need for reliable multigroup cross sections that have been collapsed over an appropriate flux spectrum. Therefore, the purpose of this thesis is to augment the current capabilities of the Monte Carlo code MCNP with a multigroup cross section generation capability. This will make it possible for the user to take advantage of the accuracy and ease of problem definition provided by MCNP while generating multigroup cross sections for other applications.

Currently it is possible to use MCNP to calculate only non-scattering multigroup cross sections. Therefore, the work presented focuses on calculating the group-to-group scattering cross section and the associated Legendre expansion of arbitrary order. The track length particle flux estimator in MCNP was used as the foundation for the cross section generation techniques. Two methods of calculating the group-to-group scattering rate and the Legendre expansion were developed and implemented in MCNP. The advantages and disadvantages of these methods are discussed in detail in the validation section.

The accuracy of the implemented cross section generation techniques is demonstrated with three test problems. Multigroup cross sections were calculated for these test problems and used in different reactor physics codes to reproduce the reference MCNP results.

Thesis Supervisor: Allan F. Henry  
Title: Professor of Nuclear Engineering

## Acknowledgments

I would like to thank Professor Henry for his guidance in this thesis and his willingness to explore new areas of research. I would also like to thank him for the many discussions we have had over the past few years from which I have learned a great deal.

I would also like to thank Professor Yanch for agreeing to be my thesis reader and for the support and guidance she has provided me over the years while working on other projects.

I also thank Professor Lanning, who was involved in the early years of this thesis. I have greatly enjoyed working with him over the many years I have been at MIT.

Erik Iverson is owed a debt of gratitude as well. His friendship helped make this last effort to finish my thesis successful.

I can never repay my family for the support, both emotional and financial, that they have provided me over the many years I have been attending MIT. At times, I thought it would never end and I am sure they had the same feelings. However, I never felt anything but support from them, for which I am truly grateful.

Valerie Schneider has also provided me with an incredible amount of support. She has gracefully tolerated numerous ups and downs in relation to this thesis. Hopefully, I will have the opportunity to return the favor.

Finally, I would like to thank Los Alamos National Laboratory which provided funding for this project. Special thanks is owed to Dr. Robert Little and Dr. John Hendricks who had the confidence in me to help establish the funding for this project. Their guidance and friendship is appreciated.

# Table of Contents

ABSTRACT .....	2
Acknowledgments .....	3
Table of Contents .....	4
List of Figures .....	7
List of Tables.....	9
<b>Chapter 1.....</b>	<b>10</b>
<b>Introduction</b>	
1.1 Previous Work.....	11
1.2 Thesis Organization.....	12
<b>Chapter 2.....</b>	<b>13</b>
<b>The Basics</b>	
2.1 General Notation .....	13
2.2 Basic Concepts .....	14
2.3 Cross Sections Desired.....	16
2.4 Cell Averaged Cross Section.....	18
2.5 Monte Carlo Tallying Methods .....	19
2.5.1 Physical Event Tally Estimator .....	20
2.5.2 Track Length Tally Estimator .....	21
2.5.3 Continuous Energy Cross Sections .....	24
2.6 Particle Current Calculation .....	25
<b>Chapter 3.....</b>	<b>27</b>
<b>MCNP Tracking Methods</b>	
3.1 MCNP Tracking Method.....	27
3.2 Target in Motion.....	29
3.3 $S(\alpha,\beta)$ Scattering .....	30
3.4 Elastic Scattering.....	32
3.5 Law 1 - Tabular Equi-Probable Energy Bins .....	33
3.6 Law 3 - Level Scattering .....	34
3.7 Law 9 - Evaporation Spectrum.....	35
3.8 Law 11 - Energy Dependent Watt Spectrum.....	35
<b>Chapter 4.....</b>	<b>37</b>
<b>Cross Section Generation Implementation</b>	
4.1 Sequence of Steps.....	37
4.2 Data Stored.....	38
4.2.1 Chi Values - Monte Carlo Approach.....	39

4.2.1.1 Single Fissionable Nuclide.....	40
4.2.1.2 All Fissionable Nuclides .....	41
4.2.2 Group-to-Group Scattering - Monte Carlo Approach .....	42
4.2.2.1 Single Nuclide.....	43
4.2.2.2 All Nuclides.....	44
4.3 Calculating Relative Errors .....	44
4.4 The Random Number Sequence.....	49
<b>Chapter 5.....</b>	<b>51</b>
<b>Group-to-Group Scattering - The Explicit Approach</b>	
5.1 Elastic Scattering - Target at Rest .....	52
5.2 Elastic Scattering - Target in Motion .....	55
5.3 Law 1 - Tabular Equi-Probable Energy Bins .....	62
5.4 Law 3 - Level Scattering .....	64
5.5 Law 9 - Evaporation Spectrum.....	64
5.6 Law 11 - Energy Dependent Watt Spectrum.....	65
5.7 Fission Chi .....	65
5.8 S( $\alpha, \beta$ ) Scattering .....	66
5.9 Other Scattering Laws .....	67
<b>Chapter 6.....</b>	<b>68</b>
<b>Legendre Components</b>	
6.1 An Example.....	73
6.2 Monte Carlo Approach - f( $\mu$ ) Estimation .....	74
6.3 Monte Carlo Approach - Direct P <sub>n</sub> Estimation.....	78
6.4 Explicit Approach - f( $\mu$ ) Estimation.....	80
6.4.1 Center of Mass Angular Distribution .....	81
6.4.2 Laboratory Angular Distributions .....	84
6.5 Explicit Approach - Direct P <sub>n</sub> Estimation .....	84
6.5.1 Center of Mass Angular Distribution .....	84
6.5.2 Laboratory Angular Distribution .....	85
6.6 Direct P <sub>n</sub> Estimation - An Alternate Method .....	85
6.7 MCNP Multigroup Libraries .....	88
<b>Chapter 7.....</b>	<b>90</b>
<b>Methods Summary</b>	
7.1 Track Length Estimator.....	90
7.2 Cross Sections Needed .....	92
7.3 Group-to-Group Scattering Rate .....	93
7.3.1 Monte Carlo Approach.....	93
7.3.2 Explicit Approach .....	94
7.4 Legendre Components.....	95
7.4.1 Monte Carlo Approach - f( $\mu$ ) Estimation .....	96
7.4.2 Direct P <sub>n</sub> Calculation.....	97
7.5 Summary of Group-to-Group Calculational Methods.....	97

<b>Chapter 8.....</b>	<b>99</b>
<b>Validation</b>	
8.1 Infinite Slab .....	101
8.1.1 $f(\mu)$ Estimation and Low Probability Events.....	106
8.2 Light Water Reactor Assembly .....	107
8.2.1 CRSRD and Very Low Probability Events .....	111
8.3 Four MITR-II Fuel Elements.....	112
<b>Chapter 9.....</b>	<b>116</b>
<b>Conclusions and Future Work</b>	
<b>References .....</b>	<b>117</b>
<b>Appendix A .....</b>	<b>118</b>
<b>New Input Cards</b>	
<b>Appendix B .....</b>	<b>120</b>
<b>List of Variables</b>	
<b>Appendix C .....</b>	<b>123</b>
<b>Source Code Modifications</b>	

## List of Figures

Figure 2-1: Energy group structure.....	13
Figure 2-2: Heterogeneous two region system in which homogenized cross sections are calculated.....	19
Figure 2-3: An example of particle tracks and interaction sites.....	20
Figure 3-1: An illustration of a collision between a neutron and the target nucleus.....	30
Figure 3-2: Illustration of three equi-probable outgoing energy bins for the two incoming energies shown. The neutron energy before and after the collision are shown in the middle of the figure.....	33
Figure 4-1: An illustration of the array in which four group reaction rate data are stored.....	39
Figure 5-1: Probability distribution function for anisotropic scattering in the center of mass system.....	53
Figure 5-2: Probability distribution function for anisotropic scattering in the center of mass system. The integration of the distribution to determine the fractional area corresponding to specific group-to-group transfers is shown. This integration is performed for the case of elastic scattering with the target-at- rest.....	55
Figure 5-3: Relation of the velocity vectors in the lab system after a scattering event. The vectors $\mathbf{u}$ , $\mathbf{p}$ , and $\mathbf{w}_c$ are coplanar. This figure was borrowed from reference 8.....	56
Figure 5-4: Probability distribution function for anisotropic scattering in the center of mass system. The integration of this distribution to determine the fractional area corresponding to specific group-to-group transfers is shown. This integration is performed for the case of elastic scattering with the target-in- motion.....	62
Figure 5-5: Illustration of three equi-probable outgoing energy bins for the two incoming energies shown. The outgoing energy bins corresponding to the neutron energy are shown in the middle of the figure.....	63
Figure 6-1: Illustration of a scattering event showing the cosine of the laboratory scattering angle.....	68
Figure 6-2: Angular distribution function for target-at-rest elastic scattering in the lab system with Deuterium. This distribution represents scattering from group 18 to 19 with a $1/E$ flux spectrum in group 18 and a constant scattering cross section. An analytical representation and a fourth order Legendre expansion are shown.....	73
Figure 6-3: A graphical representation of a three dimensional array used for storing contributions to the group-to-group scattering rates and associated angular distribution functions. This is a four group example.....	75
Figure 6-4: A plot of the estimated and analytical angular distribution functions. The estimated function was calculated with 25 equi-width cosine bins.....	77
Figure 6-5: A comparison of the reference Legendre expansion from Figure 6-2 with the expansion calculated from the angular distribution function estimated using 25 equi-width cosine bins.....	78

Figure 6-6: A graphical representation of the example center of mass angular distribution. The integrated areas corresponding to specific group-to-group transfers are shown. The fractional equi-probable cosine bins that comprise the integrated areas are labeled.	82
Figure 6-7: A plot of the isotropic center of mass angular distribution function and the associated laboratory angular distribution function for elastic scattering with Deuterium when the target is at rest.	88
Figure 8-1: A two-dimensional drawing of the MCNP model of a CE 16x16 fuel assembly inside a wet storage rack cell. This picture was drawn with the MCNP two-dimensional plotter.	108
Figure 8-2: An enlarged view of the CE 16x16 fuel assembly inside a wet storage rack cell. The boral panels on the cell wall can be seen. This picture was drawn with the MCNP two dimensional plotter.	109
Figure 8-3: MCNP model of four MITR-II fuel elements surrounded by light water. This picture was drawn with the two-dimensional plotter from MCNP.	114

## List of Tables

Table 6-1: Coefficients for the Legendre expansion shown in Figure 6-2.....	74
Table 6-2: Legendre coefficients calculated from estimated angular distribution functions having a different number of equi-width cosine bins.....	78
Table 6-3: Legendre coefficients estimated using direct integration of the angular distribution function at each scattering simulation event. ....	80
Table 6-4: A tabular representation of the data in Figure 6-6. The fractional areas of the regions and the outgoing group they represent are shown. ....	83
Table 7-1: A summary of the methods used for calculating the group-to-group scattering rates and the associated Legendre components. ....	98
Table 8-1: Standard twenty-seven group library format from SCALE 4.3. ....	100
Table 8-2: A three group library format used for cross section generation.....	100
Table 8-3: A comparison of MCNP calculations of an infinite slab. Various cross section generation options were used to see the effect on computer run time..	101
Table 8-4: A comparison of the eigenvalues estimated with different MCNP calculations of the infinite slab. ....	103
Table 8-5: A comparison of the relative error and figure of merit for specific group- to-group cross section bins calculated with different cross section generation techniques. ....	104
Table 8-6: A comparison of ONEDANT and multigroup MCNP calculated values with the continuous energy MCNP reference. The last two columns refer to group fluxes. ....	105
Table 8-7: The angular distribution function for scattering from group 1 to 3 for the infinite slab. The bin boundaries are the cosine of the laboratory scattering angles. ....	107
Table 8-8: A comparison of eigenvalues calculated with the reference MCNP calculation and multigroup MCNP calculations.....	110
Table 8-9: The results of comparing the fluxes calculated with multigroup Monte Carlo to the reference fluxes from the continuous energy Monte Carlo calculation. ....	111
Table 8-10: A comparison of eigenvalues calculated with the reference MCNP calculation and three group QUARTZ calculations.....	113
Table 8-11: The maximum percent error in flux values calculated with QUARTZ and compared to the MCNP reference values. ....	115

## Chapter 1

### Introduction

Multigroup cross sections have many applications in reactor physics. They are the foundation of discrete ordinate and diffusion theory codes as well as nodal diffusion theory codes. Multigroup cross sections also can be used in Monte Carlo codes to decrease the running time of the calculation. In addition, in some Monte Carlo codes, multigroup cross sections are essential for performing adjoint calculations.

The calculation of multigroup cross sections for use in various reactor physics codes is a task that can be rather difficult to perform. The user must calculate an appropriate flux spectrum which is used for weighting and collapsing the continuous energy or fine-group cross sections. This spectrum will depend on the particular mixture of materials and neutron source of the problem being analyzed. Typically, during cross section generation, the user is not able to represent accurately the geometry of the problem being analyzed. One and two dimensional approximations have to be made in order to calculate the flux spectrum which will be used for cross section weighting. These approximations, if made carefully, are adequate to obtain correct multigroup cross sections. However, the effort involved in generating these cross sections is still substantial.

Monte Carlo methods offer the user a great advantage in both physical accuracy and ease of problem set-up. MCNP™ \*<sup>1</sup>, a coupled neutron-photon-electron three-dimensional Monte Carlo radiation transport code from Los Alamos National Laboratory, uses continuous energy cross section data for the radiation transport. It has a full three-dimensional modeling capability with a large number of user definable tallies and source definitions. During problem setup the user specifies a system of heterogeneous cells that contain different materials (combinations of isotopes). During transport, the flux spectrum is properly calculated within each of these cells. A user can easily calculate

---

\* MCNP is a trademark of the Regents of the University of California

various reaction rates and fluxes within these cells using a user-defined energy structure. These reaction rates and fluxes can be used to determine multigroup cross sections. Since the cross section data within MCNP is organized in a fashion designed for efficient radiation transport simulation, it is not currently possible to calculate group-to-group scattering cross sections.

The purpose of this thesis is to exploit the ease and accuracy of Monte Carlo methods (specifically MCNP) in calculating multigroup cross sections for use in various radiation transport codes. The advantage of generating cross sections using a Monte Carlo code is that the flux spectrum is accurately calculated in all parts of the user-specified geometry. This eliminates the need for making one and two dimensional approximations in the cross section developing process. The major disadvantage to using Monte Carlo methods is the amount of preparation time on the part of the user and more importantly the amount of time the calculations take to execute. The advantage in accuracy will hopefully outweigh the disadvantages for certain applications.

Since this thesis is a proof of principle, MCNP was modified to calculate only neutron multigroup cross sections. Modifications to calculate photon multigroup cross sections can be implemented at a later time using the basic principles presented in this thesis.

## 1.1 Previous Work

Obtaining multigroup cross sections from Monte Carlo calculations is not a new idea. The Argonne National Laboratory computer code VIM<sup>2</sup> offers this capability. However, this code does not offer that capability to generate a full set of Legendre components for the scattering cross sections and it is not as widely used in the Nuclear industry as MCNP.

Other users have also modified MCNP to generate multigroup cross sections for use in adjoint calculations.<sup>3</sup> However, this implementation does not take advantage of the continuous energy cross section data format in generating cross sections. In addition these modifications do not offer the capability of generating a Legendre expansion which is essential if the cross sections are to be used in other transport codes.

This thesis will differ from the previous work in the following ways.

- The multigroup cross section generation capability is being added to one of the most widely used Monte Carlo codes in the industry.
- The continuous energy cross section data will be directly manipulated to obtain as much information as possible about the desired multigroup cross sections.
- A user defined number of Legendre components of the scattering cross section can be calculated for use in other reactor physics codes.
- The cross sections being generated are written in the standard ASCII 6E12.5 format which is used by other transport codes.

## 1.2 Thesis Organization

Chapter 2 will describe the fundamental aspects of multigroup cross section generation and how it is implemented in MCNP. Chapter 3 provides a brief description of the particle tracking method used in MCNP and a detailed description of some of the scattering methods. This serves as the foundation for Chapters 4, 5 and 6, which describe the cross section generation implementation, calculation of group-to-group scattering, and determination of the associated Legendre expansion. A brief summary of the methods for determining the group-to-group scattering cross sections is provided in Chapter 7. Validation of the implemented techniques is provided in Chapter 8.

Appendices A, B, and C contain the user manual for the new cross section generation features, a list of variables used in the modifications, and an MCNP patch file containing the modifications for the multigroup cross section generation technique.

## Chapter 2

### The Basics

In this chapter, the basic methods of cross section generation and particle current calculation via Monte Carlo methods will be discussed. The general equations governing cross section generation and the implementation of these equations within MCNP will be described. The basic method of generating particle current crossing a surface between adjacent regions will be also be illustrated.

#### 2.1 General Notation

This section will briefly describe the notation that will be used throughout this thesis to describe energy groups and group scattering.

Figure 2-1 presents a graphical representation of the energy structure used for group cross sections. The indexes listed in the figure will be used in the equations that follow. It is important to note that group 1 will always be the fast group and group G will be the slowest or bottom group. In MCNP, the group structure for the tallies is reversed.

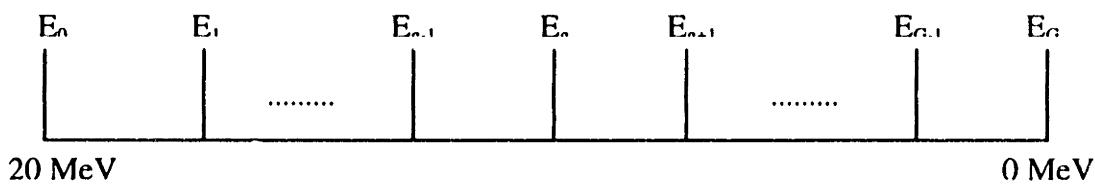


Figure 2-1: Energy group structure.

The following is a list of conventions used in this document.

- $E_0$  is the upper energy
- $E_G$  is the lowest energy
- $E_{g-1}$  is higher in energy than  $E_g$
- Energy group  $g$  contains energies  $E_g$  to  $E_{g-1}$
- The total number of energy groups is  $G$
- Energy  $E'$  is the incoming energy and  $E$  is the outgoing energy
- $\Sigma_{gg'}$  is equivalent to  $\Sigma_{g' \rightarrow g}$  and is defined as the macroscopic scattering cross section from group  $g'$  to  $g$
- The terms region, cell, and volume are used synonymously throughout this thesis to represent a region of single material. For example, the  $UO_2$  of a fuel rod would be a different cell from the zircaloy cladding.

## 2.2 Basic Concepts

Equation 2-1 is a general simplified definition of a multigroup cross section collapsed from continuous energy cross section data.

$$\Sigma_g = \frac{\int_{E_g}^{E_{g-1}} \Sigma(E) \Phi(E) dE}{\int_{E_g}^{E_{g-1}} \Phi(E) dE} \quad (2-1)$$

$\Sigma_g$  = group cross section

$\Sigma(E)$  = macroscopic cross section as a function of energy

$\Phi(E)$  = neutron flux as a function of energy

$$\Sigma(E) = \sum_{j=1}^J n_j \sigma_j(E) \rho$$

The macroscopic cross section is defined as the summation of the microscopic cross sections,  $\sigma(E)$ , ( $\text{cm}^2$ ) of the  $j$  th isotope in the material, multiplied by the associated atom fraction,  $n$ , and multiplied by the density of the material,  $\rho$ , in nuclei per cc.

As stated, the above equation is a simplified representation of cross section collapse. A more detailed derivation of multigroup cross sections will now be presented. Equation 2-2 is the continuous energy one-dimensional Boltzman transport equation.

$$\begin{aligned} \mu \frac{\partial}{\partial z} \Psi(z, \mu, E) + \Sigma_t(z, E) \Psi(z, \mu, E) = \\ \int_0^\infty dE' \int_{-1}^1 \frac{d\mu'}{2} \sum_{j=1}^J f^j(E', E) v^j(E') \Sigma_f^j(z, E') \Psi(z, \mu', E') \\ + \int_0^\infty dE' \int_0^{2\pi} \frac{d\vartheta}{2\pi} \int_{-1}^1 \frac{d\mu'}{2} \Sigma_s(z, E' \rightarrow E, \mu_0) \Psi(z, \mu', E') \end{aligned} \quad (2-2)$$

$\Sigma_t(z, E)$  = total cross section

$\Psi(z, \mu, E)$  = directional flux density

$\mu$  = cosine of angle between neutron direction and z axis

$\Sigma_f^j(z, E')$  = fission cross section for isotope  $j$

$f^j(E', E)$  = fraction of neutrons emitted per unit energy at energy  $E$  from fission of isotope  $j$  due to neutrons at energy  $E'$

$v^j(E')$  = number of neutrons emitted from fission of isotope  $j$  due to neutrons at energy  $E'$

$\Sigma_s(z, E' \rightarrow E, \mu_0)$  = double differential scattering cross section

$\mu_0$  = cosine of angle between neutron at energy  $E$  and neutron at energy  $E'$

$\vartheta$  = azimuthal angle

The scalar group flux is defined by the following equation, where the spatial integration is over some homogenous region,  $\Delta z$ , of interest.

$$\Delta z \phi_g = \int_{E_i}^{E_{i+1}} dE \int_{\Delta z} dz \int_{-1}^1 \frac{d\mu}{2} \Psi(z, \mu, E) \quad (2-3)$$

Given this definition the other multigroup cross sections can easily be defined. The total cross section, nu-sigma fission cross section, chi, and group-to-group transfer cross section are defined below.

$$\Sigma_{tg} = \frac{\int_{E_g}^{E_{g+1}} dE \int_{\Delta z} dz \int_{-1}^1 \frac{d\mu}{2} \Sigma_t(z, E) \Psi(z, \mu, E)}{\Delta z \phi_g}$$

$$v\Sigma_{fg} = \frac{\int_{E_g}^{E_{g+1}} dE \int_{\Delta z} dz \int_{-1}^1 \frac{d\mu}{2} \sum_{j=1}^J v^j \Sigma_f^j(z, E) \Psi(z, \mu, E)}{\Delta z \phi_g}$$

$$X_g = \frac{\int_0^\infty dE' \int_{E_g}^{E_{g+1}} dE \int_{\Delta z} dz \int_{-1}^1 \frac{d\mu}{2} \sum_{j=1}^J f^j(E', E) v^j \Sigma_f^j(z, E') \Psi(z, \mu, E')}{\sum_{g'=1}^G v\Sigma_{fg'} \phi_{g'} \Delta z}$$

$$\Sigma_{sg' \rightarrow g} \equiv \Sigma_{sgg'} = \frac{\int_{E_g}^{E_{g+1}} dE \int_{E_{g'}}^{E_{g'+1}} dE' \int_0^{2\pi} \frac{d\vartheta}{2\pi} \int_{\Delta z} dz \int_{-1}^1 \frac{d\mu}{2} \int_{-1}^1 \frac{d\mu'}{2} \Sigma_s(z, E' \rightarrow E, \mu_0) \Psi(z, \mu', E')}{\Delta z \phi_{g'}}$$

The fission cross section can easily be defined in a manner analogous to the  $v\Sigma_f$  cross section.

## 2.3 Cross Sections Desired

In continuous energy Monte Carlo there are many different types of reactions that occur and therefore many different types of cross sections. For multigroup methods, however, only a limited number of cross sections are needed.

- $\Sigma_{tg}$  = total group cross section
- $\Sigma_{ag}$  = absorption group cross section
- $\Sigma_{fg}$  = fission group cross section
- $v\Sigma_{fg}$  = nu-sigma fission group cross section

- $X_g$  = group chi
- $\Sigma_{sgg'}$  = group-to-group scattering cross section
- $\Sigma_{sgg'}^n$  = Legendre components of the group-to-group scattering cross section - the discussion of this cross section will be reserved for Chapter 6.

For continuous energy, the total cross section is defined to be the sum of all reaction cross sections at a particular energy.

$$\Sigma_t(E) = \Sigma_a(E) + \Sigma_f(E) + \Sigma_s^{elastic}(E) + \Sigma_s^{inelastic}(E) + \Sigma_{(n,2n)}(E) + \dots \quad (2-4)$$

For multigroup cross sections, the total cross section can be defined as :

$$\Sigma_{tg} = \Sigma_{ag} + \Sigma_{fg} + \sum_{g'=1}^G \Sigma_{sg'g} \quad (2-5)$$

As will be shown, this definition depends on the method used to calculate the group-to-group scattering cross section. In comparing equations 2-4 and 2-5, it is obvious that the  $(n,2n)$  term seems to be missing. In actuality, this cross section has been included with the group-to-group scattering cross section. As an explanation, the cross section,  $\Sigma_{(n,2n)gg'}$ , will be defined as the cross section for neutrons emerging into group  $g$  from a neutron originating in group  $g'$ . This cross section can be defined with the following equation.

$$\Sigma_{(n,2n)g' \rightarrow g} \equiv \Sigma_{(n,2n)gg'} = \frac{\int_{E_{g'}}^{E_{g'+1}} dE \int_{E_{g'}}^{E_{g'+1}} dE' \int_{\Delta z} dz \int_{-1}^1 \frac{d\mu'}{2} f_{(n,2n)}(E', E) 2\Sigma_{(n,2n)}(z, E') \Psi(z, \mu', E')}{\Delta z \phi_{g'}} \quad (2-6)$$

The factor of two represents the multiplicity of the reaction, two neutrons out for each neutron in. Since this cross section now resembles the scattering cross section, it is simply added into the appropriate group-to-group scattering cross section. For a single group, the following equation would represent the total cross section when the  $(n,2n)$  reaction is included in the scattering cross section,  $\Sigma_{s-eff}$ .

$$\begin{aligned}\Sigma_t &= \Sigma_a + \Sigma_f + \Sigma_{s\text{-eff}} \\ \Sigma_{s\text{-eff}} &= \Sigma_s + 2\Sigma_{(n,2n)}\end{aligned}\tag{2-7}$$

The factor of two is the same factor of two depicted above in equation 2-6. The implication in equation 2-7 is that the total cross section may not be equal to the sum of the parts, since the multiplicity of the  $(n,2n)$  reaction is included in the scattering cross section. If there is substantial  $(n,2n)$  reaction and if the absorption cross section is defined as true absorption (a neutron enters and no neutrons leave) then the total cross section indeed will be less than the sum of the parts. In this thesis, the absorption cross section being calculated is the true absorption and therefore in some situations, as described above, the total cross section may be less than the sum of the parts.

## 2.4 Cell Averaged Cross Section

Instead of calculating group cross sections in a single material, the user may wish to calculate homogenized group cross sections over a region of heterogeneous materials. Equations 2-8 through 2-10 describe how this is done in order to maintain a reaction rate balance. Figure 2-2 is a simple graphical representation of the two regions mentioned in these equations.

$$(V^1 + V^2) \Sigma_g^{1-2} \phi_g^{1-2} = V^1 \Sigma_g^1 \phi_g^1 + V^2 \Sigma_g^2 \phi_g^2 \tag{2-8}$$

$$\phi_g^{1-2} = \frac{V^1 \phi_g^1 + V^2 \phi_g^2}{V^1 + V^2} \tag{2-9}$$

From these equations the cell averaged cross section can easily be derived. Using equations 2-8 and 2-9, and combining terms, equation 2-10 can be written; this defines the cell averaged cross section for the two-region problem depicted in Figure 2-2.

$$\Sigma_g^{1-2} = \frac{V^1 \Sigma_g^1 \phi_g^1 + V^2 \Sigma_g^2 \phi_g^2}{(V^1 + V^2) \phi_g^{1-2}} = \frac{V^1 \Sigma_g^1 \phi_g^1 + V^2 \Sigma_g^2 \phi_g^2}{V^1 \phi_g^1 + V^2 \phi_g^2} \tag{2-10}$$

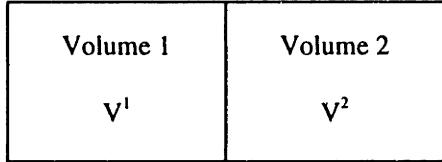


Figure 2-2: Heterogeneous two region system in which homogenized cross sections are calculated.

## 2.5 Monte Carlo Tallying Methods

In Monte Carlo calculations, each particle has associated with it a certain weight,  $W_i$ . This weight is adjusted throughout the calculation in accordance with the physics of the problem. An example of a way in which a particle weight can be modified is the capture event. There are two ways of performing capture in MCNP. The first is called analog capture and is basically a true simulation of the physics: if a capture event is supposed to occur, then the particle is terminated and another particle is started from the source.

The second form of capture is called implicit capture. In this method, a particle is not terminated when a capture event occurs. Rather, the particle weight is reduced by the ratio of the capture cross section to the total cross section times the original weight. If the weight is viewed, in an artificial sense, as a certain number of particles, the resulting weight after implicit capture is the fraction of those particles that survive. This method of capture is the default method of capture in MCNP neutron calculations. Therefore each starting particle continues for many capture events before being terminated by a low weight cutoff. This enables more tally information to be accumulated over the Monte Carlo lifetime of a particle than if the particle were terminated in the fashion of analog capture.

In Monte Carlo calculations there are different ways to accumulate (tally) information for estimating physical quantities such as flux and reaction rate. One way is to accumulate information at each physical event (scattering or capture) that alters the path of a particle. The second method is to accumulate information along each track length of that path. The latter is the method used in MCNP. Both methods will be briefly discussed with

emphasis on cross section generation. The track length method will be discussed in greater detail since it is the method being used to generate cross sections.

### 2.5.1 Physical Event Tally Estimator

Figure 2-3 depicts three cells and illustrates the path of the particle in this example. The track segments are labeled in this figure. A track length segment is defined as the distance a particle travels from an interaction to another interaction or to a cell boundary between volume regions. The birth of the particle occurred at the head of track segment 1 and the termination of the particle occurred at the end of track segment 8.

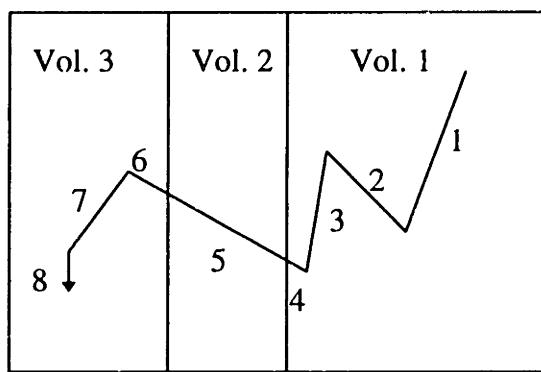


Figure 2-3: An example of particle tracks and interaction sites.

The flux and the various reaction rates can be tallied at each collision site. However, there is one obvious disadvantage to this method. For the particle illustrated in Figure 2-3 there would not be any contribution to the flux or reaction rate tallies in volume 2 since there are no interaction sites in this volume. This drawback means that the statistical error associated with flux and reaction rate tallies for optically thin cells could be quite large.

Because of the drawback mentioned, tallying flux and reaction rates at the collision sites is not the method of choice in MCNP and was not chosen for the implementation of cross section generation.

It should be noted, however, that calculating the group-to-group scattering cross section becomes very straightforward when the tally is performed at collision sites. Since the particle energy is known before the interaction and after the interaction it is very straightforward to tally into the appropriate group-to-group scattering bin. However, this

also means that the contribution is only made to one of the group-to-group scattering bins and not all of them. The result would be that the statistical error associated with the scattering cross sections would be worse than the error associated with the other cross sections that are independent of outgoing energy. This discussion will be revisited in Chapters 4 and 5 when the methods of calculating the scattering cross section are discussed.

### 2.5.2 Track Length Tally Estimator

The following description of a track length estimation of physical quantities is taken from the MCNP manual.<sup>1</sup>

The description will begin with the basic definition of particle flux and proceed from there. In the equations listed below, time is used as a variable and the values are integrated over time. In the case of MCNP calculations and the equations listed below, time refers to the time from the birth of a particle to the death of a particle. This concept of time is not the same as the wall clock time that would be realized during an experiment. All MCNP calculations are static calculations and are not time dependent. The results of an MCNP calculation provide a snap-shot in time of the state of the system being analyzed. Within that snap shot it is assumed that all particles start at the same instant in wall clock time. MCNP quantities can be tallied as a function of time; however, this time is from the birth of the particle where all particles are started at the same wall clock time.

Particle flux is defined as:

$$\Phi(\vec{r}, E, t) = v N(\vec{r}, E, t)$$

$v$  = particle velocity

$N$  = particle density

The time integrated volume averaged group flux,  $\phi_g$ , is:

$$\phi_g = \int \int \int_{V \ t \ E_i}^{E_{g-1}} \Phi(\vec{r}, E, t) dEdt \frac{dV}{V} \equiv \int \int \int_{V \ t \ E_i}^{E_{g-1}} v N(\vec{r}, E, t) dEdt \frac{dV}{V}$$

Since distance,  $ds$ , is equivalent to velocity multiplied by time,  $vdt$ , the following can be written.

$$\phi_g = \int_V \int_{E_t}^{E_{t+1}} \int_{\vec{r}} \Phi(\vec{r}, E, t) dEdt \frac{dV}{V} = \int_V \int_s \int_{E_t}^{E_{t+1}} N(\vec{r}, E, t) dEds \frac{dV}{V}$$

In Monte Carlo calculations, particle density is equivalent to a summation of particle weight per unit volume and distance is equivalent to track length, TL. Therefore the definition above can be represented as:

$$\phi_g = \frac{\int_{E_t}^{E_{t+1}} dE \int_V dV \sum_{i=1}^N WTL_v^i(E)}{V \sum_{i=1}^N W_0^i} \quad (2-11)$$

$WTL_v^i(E)$  = weight of ith particle times the track length of ith particle at energy E in volume region V

V = volume of region of interest

$W_0^i$  = original weight of particle i

N = number of particle histories tracked

The  $W_0$  is the original weight and normalizes the answers to a single starting particle. The weight of the ith particle in the term WTL can change with each track length. For example, in Figure 2-3 the weight of the particle along track length 1 could be different than the weight of the particle along track length 2. The change in weight might occur because of the physics of the problem. Also, the energy of the particle usually changes between track lengths.

A similar equation can be written to define a group reaction rate calculated with a track length estimator.

$$\Sigma_g \phi_g = \frac{\int_{E_t}^{E_{t+1}} dE \int_V dV \sum_{i=1}^N WTL_v^i(E) \Sigma(r, E)}{V \sum_{i=1}^N W_0^i} \quad (2-12)$$

$\Sigma(r, E)$  = cross section as a function of energy and position

The integration over energy and volume in equations 2-11 and 2-12 is performed as a summation. A separate tally bin is created for each energy group and volume region combination. The WTL product is then placed in the appropriate bin for each track length segment that occurs. This method effectively performs the integration over energy and volume. The following terms will be defined to simplify the notation in future equations.

$$\sum_{i=1}^N WTL_{gv}^i = \int_{E_g}^{E_{g+1}} dE \int_V dV \sum_{i=1}^N WTL_v^i(E)$$

$$\sum_{i=1}^N WTL_{gv}^i \Sigma_{gv}^* = \int_{E_g}^{E_{g+1}} dE \int_V dV \sum_{i=1}^N WTL_v^i(E) \Sigma(r, E)$$

Equations 2-11 and 2-12 can then be rewritten in the following form.

$$\phi_g = \frac{\sum_{i=1}^N WTL_{gv}^i}{V \sum_{i=1}^N W_0^i} \quad (2-13)$$

$$\Sigma_g \phi_g = \frac{\sum_{i=1}^N WTL_{gv}^i \Sigma_{gv}^*}{V \sum_{i=1}^N W_0^i} \quad (2-14)$$

Equation 2-13 can be used with some algebraic manipulation to express equation 2-9 in terms of a track length estimator.

$$\phi_g^{1-2} = \frac{\sum_{i=1}^N WTL_{gv^1}^i + \sum_{i=1}^N WTL_{gv^2}^i}{(V^1 + V^2) \sum_{i=1}^N W_0^i} = \frac{\sum_{i=1}^N WTL_{gv^1v^2}^i}{(V^1 + V^2) \sum_{i=1}^N W_0^i} \quad (2-15)$$

Using these definitions, equation 2-10 can be written in the following form. It can then be simplified to provide equation 2-16.

$$\Sigma_g^{1-2} = \frac{V^1 \frac{\sum_{i=1}^N WTL_{g,v^1}^i \Sigma_{g,v^1}^*}{\sum_{i=1}^N W_0^i} + V^2 \frac{\sum_{i=1}^N WTL_{g,v^2}^i \Sigma_{g,v^2}^*}{\sum_{i=1}^N W_0^i}}{(V^1 + V^2) \frac{\sum_{i=1}^N WTL_{g,v^1v^2}^i}{\sum_{i=1}^N W_0^i}}$$

$$\Sigma_g^{1-2} = \frac{\sum_{i=1}^N WTL_{g,v^1}^i \Sigma_{g,v^1}^* + \sum_{i=1}^N WTL_{g,v^2}^i \Sigma_{g,v^2}^*}{\sum_{i=1}^N WTL_{g,v^1v^2}^i} \quad (2-16)$$

There are a couple important points to note about equation 2-16. The first is that this equation does not explicitly depend on the summation of all starting weights. The second, and more important point, is that this equation does not depend on the volume of regions 1 and 2 and the combined region 1-2. The normal track length estimate of the flux requires that the user know the volume of the region of interest or that MCNP is able to calculate the volume. The latter occurs only if the region is one of a select few shapes and rotationally symmetric. Therefore, for odd shaped volumes, the user could expend considerable effort in calculating the region volumes. Since equation 2-16 does not depend on the volume, the user is free to tally over any combination of regions without regard to the calculation of the volumes.

Equation 2-16 is an accurate representation of the manner in which cross section generation has been implemented in MCNP for this thesis. For each region of interest, the numerator of equation 2-16 is tallied for all reaction rates and the denominator of equation 2-16 is tallied. The actual division only occurs at the end of the calculation before the final data are output.

### 2.5.3 Continuous Energy Cross Sections

Equation 2-16 describes the basic method of calculating reaction rates and macroscopic cross sections using the track length method of tallying. The only remaining

question is where the  $\Sigma(r, E)$  comes from. Since MCNP uses microscopic continuous energy cross sections for transport, these same cross sections are used in equation 2-16.

MCNP automatically calculates the macroscopic total and absorption cross sections for each material at each particle energy. Therefore these cross sections are used without modification. The total macroscopic fission cross section is not automatically calculated; therefore, MCNP was modified to calculate this cross section and use it for cross section generation.

The scattering cross sections, which include any reaction where a neutron comes in and a neutron leaves except fission, are not energy to energy cross sections but rather total cross sections. These cross sections represent the probability that a scattering event will occur but do not offer any information about the outcome of the scattering event. The mechanics of the scattering event are handled through the scattering laws that are programmed in MCNP. Therefore, it is not straightforward to calculate group-to-group scattering, since the  $\Sigma_s$  that is available does not describe the energy transfer. Chapters 4 and 5 describe how the  $\Sigma_s$  that is available can be converted to a group-to-group scattering cross section for use in Equation 2-16. This conversion and its implementation are the crux of the thesis.

The calculation of the fission Chi values is very similar to the method used for obtaining the group-to-group scattering cross sections and is also discussed in later chapters.

## 2.6 Particle Current Calculation

MCNP offers the capability to calculate particle current crossing a user defined surface. However, this current is averaged over the entire area of the surface. Since volume regions (cells) in MCNP are created with the intersection and union of surfaces, a single surface will very likely be used to define multiple cells. The result is that the surface average current is not necessarily equivalent to the current exiting or entering a single cell. This makes it extremely difficult for the user to calculate the current going from one cell to another. Since this quantity is essential to the determination of

discontinuity factors for use in Nodal Diffusion Methods,<sup>4,5</sup> MCNP was modified to calculate the current going from one cell to another. Since current is simply particles per unit time, the estimation of the current from one cell to another is a very simple summation defined by the following equation.

$$J_{g,V^1 \rightarrow V^2} = \frac{\sum_{i=1}^N W_{V^1 \rightarrow V^2}^i}{\sum_{i=1}^N W_0}$$

$J_{g,V^1 \rightarrow V^2}$  = the current in group g for particles going from  $V^1$  to  $V^2$

$W_{V^1 \rightarrow V^2}^i$  = the particle weight of the ith particle going from  $V^1$  to  $V^2$

## Chapter 3

### **MCNP Tracking Methods**

This chapter describes the steps MCNP performs to track particles from one location to another. This description is provided as a foundation for the discussions in later chapters.

Many of the MCNP steps mentioned in this chapter are used in the calculation of the group-to-group scattering cross sections. In addition, a few of the MCNP routines used to perform some of these steps were copied and modified for use in cross section generation.

A simple description of the method of particle transport used in MCNP will be provided followed by an expanded discussion of some of the steps. A detailed description of some of the scattering laws will also be provided. The MCNP manual<sup>1</sup> uses the phrase scattering law to define any function or set of data that determines the outgoing energy of a particle from a reaction, including fission and scattering. The equations and general description of these laws were borrowed from the MCNP manual.<sup>1</sup> Only a few of the scattering laws are mentioned because these laws were modified for use in cross section generation. A description of the remaining scattering laws can be found in the MCNP manual.<sup>1</sup>

In this chapter, and future chapters, the term “scattering angle” is used to designate the cosine of the scattering angle. The value of the scattering angle thus ranges between -1 and 1.

#### **3.1 MCNP Tracking Method**

The discussion in this section applies to criticality calculations where the fission reaction is treated as an absorption reaction. If fission were real, the sampling of scattering mentioned below would include the possibility of fission.

The steps listed below illustrate the process MCNP uses for particle transport. The discussion provided is general in nature and does not include extraneous details such as variance reduction techniques. The exact details of certain steps will be expanded in later sections.

Step 1 takes place at either a particle birth site, after a particle interaction, or at a cell boundary.

1. The distance to the next collision is sampled along with the distance to the cell boundary. The shorter distance is chosen and the particle is transported to that location. If the location is a cell boundary, the transport process resumes at step 1.
2. The nuclide used in the sampling of the collision is chosen based on the macroscopic total cross sections and the individual microscopic total cross section for the nuclides present in the material.
3. The velocity of the target nucleus is sampled if the speed of the incoming neutron is sufficiently low or if the target is Hydrogen. The relative velocity of the neutron and the direction of the target nucleus are calculated.
4. The capture event is sampled.

#### *Analog Capture*

The absorption event is sampled and if chosen the particle is terminated. If the absorption event is not chosen the transport continues with step 5.

#### *Implicit Capture*

The weight of the particle is adjusted by the ratio of the absorption cross section to the total cross section and the new weight is given by the following relationship.

$$W_{\text{new}} = W_{\text{old}} \times \left(1 - \frac{\sigma_{\text{abs}}}{\sigma_{\text{total}}}\right)$$

If fission is being treated as capture (as in criticality calculations), the absorption cross section in that equation includes the fission cross section.

The transport continues with step 5.

5. The individual scattering reaction is chosen based on a sampling of the reactions available for the chosen nuclide at the incoming energy of the particle.
6. If the reaction chosen has more than one scattering law associated with it, a single law is sampled based on probabilities provided in the cross section data. The scattering laws define the relationship between incoming particle energy and outgoing particle energy. Some of these laws are discussed in detail later.
7. The outgoing particle lab energy and lab scattering angle are determined using the scattering law chosen in step 6.
8. The particle energy and direction are adjusted using the results from step 7 and transport resumes at step 1.

### **3.2 Target in Motion**

This section expands the discussion in the previous section for the case of target-in-motion. The description provided here was extracted from the MCNP manual <sup>1</sup> and the MCNP FORTRAN source code.

Figure 3.1 shows the relationship between a neutron and the target in motion relative to the laboratory system. The terms listed in the figure are:  $V_n$  is the velocity vector of the neutron,  $V_t$  is the velocity vector of the target nucleus,  $V_{n-rel}$  is the relative velocity vector of the neutron, and  $\mu_t$  is the cosine of the angle between  $V_n$  and  $V_t$ .

In step 3, above, the target velocity,  $V_t$ , and angle,  $\mu_t$ , were calculated. Using these terms the relative velocity, including direction, of the particle is calculated. This, effectively, represents the situation where the target is at rest.

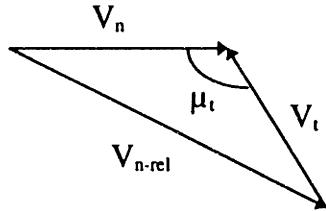


Figure 3-1: An illustration of a collision between a neutron and the target nucleus.

In step 5, above, the individual scattering reaction is chosen. The cross sections used in this choice are evaluated at the relative energy of the neutron. Then in step 7, the scattering is actually performed, as if the target were at rest. For example, if the scattering reaction were elastic, then the simple relationships for scattering in the center of mass coordinates could be used (these are discussed later).

The final energy and direction of the particle are calculated in step 8. This is a two part process when the target is in motion. First, the relative energy and direction of the particle are adjusted using the results of step 7. Finally, the velocity of the target nucleus is used to adjust the new relative velocity of the particle back to the true lab system.

This discussion illustrates the complexity of calculating a scattering event when the target is in motion. The complexity of this calculation will be addressed again in a later chapter.

### 3.3 S( $\alpha, \beta$ ) Scattering

A simple description of some of the data on an S( $\alpha, \beta$ ) library is provided in this section. Only the data that will be referenced in a later chapter are discussed. The reader can consult the MCNP manual<sup>1</sup> for additional information on all of the data formats available in an S( $\alpha, \beta$ ) library.

In step 5 it is possible to avoid the scattering reactions for the chosen nuclide if an S( $\alpha, \beta$ ) library is associated with the isotope. For example, the light water S( $\alpha, \beta$ ) library, *lwtr.01t*, is associated with Hydrogen; therefore, anytime a neutron of energy less than

~4keV scatters with Hydrogen, the  $S(\alpha,\beta)$  cross section data are used instead of the Hydrogen data.

There are two possible forms of scattering in  $S(\alpha,\beta)$  data, elastic and inelastic. This discussion will focus only on the inelastic scattering data. There are also different ways to represent the inelastic scattering data and this discussion will present the one used for the *lwtr.01t*  $S(\alpha,\beta)$  library.

The inelastic data consist of a number of incoming particle energies at which the inelastic cross section is evaluated. For each of these incoming energies, there are a discrete number of outgoing energies. These outgoing energies may or may not be equally probable. For each outgoing energy there are a discrete number of lab scattering angles (cosine of the angle). These angles are equally probable.

The interpolation fraction used in the determination of the outgoing energy is defined as:

$$r = \frac{E_n - E_{in}^i}{E_{in}^{i+1} - E_{in}^i}$$

$E_n$  = neutron incoming energy

$E_{in}^i$  = incoming energy on library that is below  $E_n$

$E_{in}^{i+1}$  = incoming energy on library that is above  $E_n$

The position within the outgoing energies is randomly chosen and then the outgoing energy is interpolated in the following manner.

$$E_{out} = E_{out}^{i,j} + r \times (E_{out}^{i+1,j} - E_{out}^{i,j})$$

$E_{out}^{i,j}$  = the jth outgoing energy associated with  $E_{in}^i$

The lab angle is also interpolated between adjacent tables. First the position within the lab angles is randomly chosen. Finally the lab angle is interpolated between the lab angles associated with  $E_{out}^{i,j}$  and  $E_{out}^{i+1,j}$ .

### 3.4 Elastic Scattering

Elastic scattering in MCNP is performed in the center of mass coordinates. The data used to determine the final particle energy and direction in elastic scattering consist of a table of incoming energies. For each incoming energy there is an associated table of thirty-two equi-probable cosine bins representing the possible center of mass scattering angles.

When elastic scattering occurs, one of the tables of equi-probable cosine bins is picked based on the incoming particle energy. Then one of the thirty-two bins is randomly chosen and a center of mass scattering angle (cosine of the angle) is randomly chosen within that bin.

The outgoing laboratory energy is determined by the following equation.<sup>6</sup>

$$E = E' \times \frac{A^2 + 2A\mu_c + 1}{(A + 1)^2} \quad (3-1)$$

$E$  = outgoing particle energy

$E'$  = incoming particle energy

$$A = \frac{M_t}{M_n}$$

$M_t$  = mass of target

$M_n$  = mass of neutron

$\mu_c$  = cosine of center of mass scattering angle

The cosine of the laboratory scattering angle is determined by the following equation.

$$\mu_L = \frac{A\mu_c + 1}{\sqrt{A^2 + 2A\mu_c + 1}} \quad (3-2)$$

A more detailed discussion of scattering in the center of mass coordinates can be found in reference 6.

### 3.5 Law 1 - Tabular Equi-Probable Energy Bins

This scattering law relates incoming particle energy to outgoing particle energy through equi-probable energy bins. The data for this law specify a certain number of incoming energies. For each incoming energy, a set of equi-probable energy bins represents the possible outgoing energies. In addition, there are thirty-two equi-probable cosine bins in the lab system for each of a certain number of incoming energies. The incoming energy grid used for the outgoing energies is not necessarily the same as the incoming energy grid used for the laboratory angular bins.

Scaled linear interpolation is used to determine the outgoing energy. The easiest way to explain this method is by example.

Figure 3-2 shows equi-probable energy bins for two incoming energies. The particle energy in this example,  $E_n$ , is between the two incoming energies.

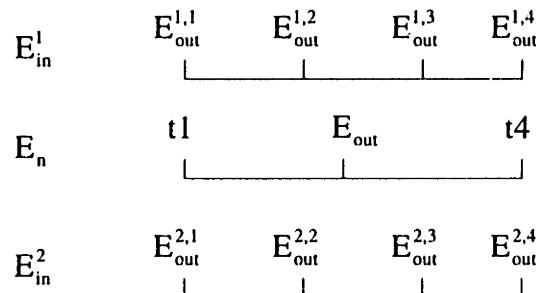


Figure 3-2: Illustration of three equi-probable outgoing energy bins for the two incoming energies shown. The neutron energy before and after the collision are shown in the middle of the figure.

The first step in determining the outgoing energy is to randomly choose one of the three equi-probable outgoing energy bins. For this example, the choice will be the second bin. The following terms are then calculated.

$$r = \frac{E_n - E_{in}^1}{E_{in}^2 - E_{in}^1}$$

$r1$  = random number

r2 = random number

$$t1 = E_{\text{out}}^{1,1} + r \times (E_{\text{out}}^{2,1} - E_{\text{out}}^{1,1})$$

$$t4 = E_{\text{out}}^{1,4} + r \times (E_{\text{out}}^{2,4} - E_{\text{out}}^{1,4})$$

The outgoing energy bins associated with  $E_{\text{in}}^1$  will be used for the final calculation if  $r1 \leq r$ , otherwise the outgoing energy bins associated with  $E_{\text{in}}^2$  will be used. For this example, assume the second set of outgoing energies is chosen. Finally the outgoing energy is chosen using the following equation.

$$E_{\text{out}} = t1 + (t4 - t1) \times \frac{[E_{\text{out}}^{2,2} + r2 \times (E_{\text{out}}^{2,3} - E_{\text{out}}^{2,2})] - E_{\text{out}}^{2,1}}{E_{\text{out}}^{2,4} - E_{\text{out}}^{2,1}} \quad (3-3)$$

After the outgoing energy is determined, the outgoing laboratory scattering angle is calculated by randomly choosing one of the thirty-two equi-probable cosine bins and randomly choosing a cosine within that bin.

### 3.6 Law 3 - Level Scattering

This scattering law is very straightforward. The data for this law consist of two values, A1 and A2, and a set of thirty-two equi-probable cosine bins in the center of mass system. With the values provided, the outgoing energy in the center of mass coordinates can be calculated. Using the center of mass outgoing energy and a center of mass scattering angle randomly chosen from the thirty-two equi-probable bins, the outgoing energy in the laboratory system and the lab scattering angle can be calculated.

The equations used to perform these calculations are listed here.

A1 = an energy value; A2 = a scaling factor

$$E_{\text{out}}^{\text{cm}} = A2 \times (E' - A1)$$

$$E_{\text{out}}^{\text{lab}} = E_{\text{out}}^{\text{cm}} + \frac{E' + 2\mu_c(A+1)\sqrt{E' \times E_{\text{out}}^{\text{cm}}}}{(A+1)^2} \quad (3-4)$$

$$\mu_L = \mu_c \sqrt{\frac{E_{out}^{cm}}{E_{out}^{lab}}} + \frac{1}{A+1} \sqrt{\frac{E'}{E_{out}^{lab}}} \quad (3-5)$$

### 3.7 Law 9 - Evaporation Spectrum

This scattering law is used for inelastic scattering and also used to determine the outgoing energy of neutrons from fission reactions. The data for this law consist of values of the variable T tabulated as a function of incoming energy, E', and a single restriction energy. Interpolation is used between values of T to determine the exact value for the neutron incoming energy.

The following equations define the relationship between incoming particle energy and outgoing particle energy.

$$T \equiv T(E')$$

$$f(E' \rightarrow E) = \frac{1}{C} E e^{-\frac{E}{T}} \quad (3-6)$$

$$0 \leq E \leq E' - U$$

$$U = \text{restriction energy}$$

$$C = T^2 \left[ 1 - e^{-\frac{(E'-U)}{T}} \left( 1 + \frac{E' - U}{T} \right) \right] \quad (3-7)$$

The sampling of this relationship to determine the outgoing laboratory energy is described in the MCNP manual.<sup>1</sup>

When this law is used for scattering, the laboratory scattering angle is sampled from thirty-two equi-probable scattering bins.

### 3.8 Law 11 - Energy Dependent Watt Spectrum

This scattering law is used to determine the outgoing energy of neutrons from fission reactions. The data for this law consist of values of two variables, a and b, tabulated as a function of incoming energy and a single restriction energy. Interpolation is used between

values of the variables to determine the exact value appropriate to a particular neutron incoming energy.

The following equations define the relationship between incoming particle energy and outgoing particle energy.

$$a \equiv a(E')$$

$$b \equiv b(E')$$

$$f(E' \rightarrow E) = \frac{1}{C} e^{\frac{-E}{a}} \sinh(\sqrt{bE}) \quad (3-8)$$

$$0 \leq E \leq E' - U$$

$U$  = restriction energy

$$C = -a e^{\frac{-(E'-U)}{a}} \sinh(\sqrt{b(E'-U)}) + \frac{1}{2} \sqrt{\frac{\pi a^3 b}{4}} e^{\frac{ab}{4}} \left[ \operatorname{erf}\left(\sqrt{\frac{(E'-U)}{a}} - \sqrt{\frac{ab}{4}}\right) + \operatorname{erf}\left(\sqrt{\frac{(E'-U)}{a}} + \sqrt{\frac{ab}{4}}\right) \right] \quad (3-9)$$

The sampling of this relationship to determine the outgoing laboratory energy is described in the MCNP manual.<sup>1</sup>

When this law is used for scattering, the laboratory scattering angle is sampled from thirty-two equi-probable scattering bins.

## Chapter 4

# Cross Section Generation Implementation

Chapter 2 described the basics of the track length estimator and how it is used to estimate cross sections. This chapter will describe the implementation of this estimator in MCNP for the purpose of cross section generation.

The details of which values are stored and where the values are calculated will be discussed. The estimation of the relative errors for the calculated cross sections is also presented. The effect cross section generation has on the random number sequence will also be addressed.

### 4.1 Sequence of Steps

Chapter 3 described the methodology used in MCNP for tracking particles from one location to another. Since this tracking method uses all of the cross sections desired, it seemed logical to copy a number of the MCNP subroutines and adopt them for use in cross section generation. Some of the subroutines were copied and modified and others were simply called when appropriate.

The following steps illustrate the procedure used for calculating cross sections during the tracking of an individual particle history. Some of these steps will be very similar to the steps described in Section 3.1 and in fact a number of the same subroutines were used to perform these steps. Further explanation of some of these steps is provided in the next section.

1. After a particle is transported to a collision site or cell boundary, the cross section generation control subroutine is called. This call occurs at approximately the same location as the call to the regular MCNP tally subroutine.
2. The group to which the particle belongs is determined based on the incoming energy of the particle.

3. The flux, total reaction rate, absorption rate, fission rate, and nu times the fission rate are stored in the temporary array in the appropriate group location.
4. If fission is possible, either one of the fissionable isotopes is chosen and the chi distribution is calculated or the chi distribution is calculated for all fissionable isotopes. The choice is based on user input, the rationale for which will be provided later.
5. Either an individual nuclide is chosen or all nuclides are used for simulating scattering events. The choice is based on user input. The subroutine controlling the scattering calculations is then called. The simulated events that are mentioned in these steps and in this thesis are used only for cross section generation and do not affect normal particle transport.
6. The velocity of the target nucleus is sampled if the speed of the incoming neutron is sufficiently low or if the target is Hydrogen. The angle between the target velocity and the direction of the target nucleus is calculated. If required, the relative velocity of the neutron is also calculated.
7. The group-to-group scattering rate contribution is calculated and stored in the appropriate location. If the user requested, data for the estimation of the angular distribution of the scattered neutrons is also calculated and stored.
8. The storage of cross section data is complete and particle tracking resumes.

At the end of a particle history, the cross section generation subroutine is called again and all of the data in the temporary array are transferred to the primary array. At this time, the values needed for the determination of the relative error are also calculated.

## 4.2 Data Stored

Figure 4.1 is an illustration of the array used to store the various reaction rate data calculated. There is a separate array for every cell in which the calculation is being performed.

						$\phi_g \Sigma_{sg' \rightarrow g}$			
		$\phi_g \Sigma_{ag}$	$\phi_g \Sigma_{fg}$	$\phi_g v \Sigma_{fg}$	group to				
		flux	$X_g$	$\phi_g \Sigma_{fg}$	$\phi_g v \Sigma_{fg}$	1	2	3	4
group from	1								
	2								
	3								
	4								

Figure 4-1: An illustration of the array in which four group reaction rate data are stored.

In step 3, listed above, the flux and the total, absorption, fission, and nu times fission reaction rates are stored. The following is a list of these specific quantities as they are calculated and saved.

- Flux = WTL
- Total = WTL $\Sigma_t$
- Absorption = WTL $\Sigma_a$
- Fission = WTL $\Sigma_f$
- Nu times fission = WTL $v\Sigma_f$

The macroscopic total cross section is recalculated every time a particle changes energy. The macroscopic fission cross section and nu times fission are not automatically calculated in MCNP. Therefore, the subroutine that generates the macroscopic total cross section was modified to generate the fission and nu times fission cross sections when the total cross section is calculated.

#### 4.2.1 Chi Values - Monte Carlo Approach

Step 4, of Section 4.1, determines the chi value for materials with fissionable nuclides. The calculation of this value is not as straightforward as the other quantities

since the continuous energy cross section data are not in a readily accessible format for the determination of this quantity.

In MCNP, the energy distribution of the neutrons emerging from fission is usually determined with laws 9 or 11, as discussed in the last chapter. It is obvious from the discussion that these scattering laws do not readily dictate the fraction of neutrons emerging from fission that belong to a certain group.

Therefore, a Monte Carlo approach is taken to determine the group chi values. The same MCNP routines used to determine the energy of the neutrons emerging from fission during transport are used for this calculation. The basic concept behind the Monte Carlo approach is to simulate fission events and observe the outgoing energies. These outgoing energies will then determine the groups into which the data will be stored.

This approach to estimating the chi values can be performed using a single fissionable nuclide in the material or all of the fissionable nuclides. The latter provides more information for each track length segment which improves statistical convergence. However, the run time also increases. Both ways of executing the Monte Carlo approach are discussed below.

Chapter 5 describes an alternative to this Monte Carlo approach that offers improved statistical convergence.

#### ***4.2.1.1 Single Fissionable Nuclide***

In the single nuclide Monte Carlo approach, the fission reaction rate will be stored multiple times in the appropriate group location for the chi value. This reaction rate is given by:

$$WTL\Sigma_f$$

The first step, when using a single nuclide for the estimation of the Chi, is to randomly choose a fissionable nuclide. The probability for choosing a specific nuclide is:

$$\frac{n^j \sigma_f^j}{\sum_{j=1}^J n^j \sigma_f^j}$$

$n^j$  = atom fraction for fissionable nuclide j

After the nuclide is chosen, a single fission reaction is sampled (ex. (n,f), (n,n'f), (n,2nf), etc.). The reaction is chosen based on the individual fission reaction cross sections. Next, an integer number of neutrons emerging from this fission reaction is chosen. The formula used to determine the integer number of neutrons is:

number of neutrons emerging  $\equiv NE = \text{int}(v + r1)$

where

$\text{int}()$  = the fortran integer function that truncates a real number (ex.  $\text{int}(2.7) = 2$ )

$r1$  = random number between 0 and 1

$v$  = average number of outgoing neutrons per fission - this value is dependant on the incoming energy of the neutron

The chosen reaction is then sampled  $NE$  times. After sampling the reaction, the reaction rate value shown above is stored  $NE$  times in the chi group bins according to the energies of the  $NE$  outgoing particles.

The result is that  $nu$  times the fission rate is stored, on average,  $nu$  times to represent the chi values. Since only a couple of the chi group bins receive data for each track length, the statistical uncertainty associated with some of the chi group bins can be rather large.

#### 4.2.1.2 All Fissionable Nuclides

There are a few differences between sampling the fission reaction from a single fissionable nuclide or all of the fissionable nuclides.

The first difference is the stored value. When all fissionable nuclides are sampled, the individual reaction rates for each of the nuclides is stored. These values are:

$$WTLn^j\sigma_f^j\rho$$

$\rho$  = atom density of material

The fission reactions are sampled, for each fissionable nuclide, and the individual reaction rates are stored NE times in the same manner as described above. Since every fissionable nuclide is sampled, there is more information being stored in the chi group bins and possibly more bins obtaining information in this situation than in the case of sampling a single nuclide. There is, of course, a penalty. The amount of computer time to sample the chi values will increase proportionally to the number of nuclides sampled. This increase in computer time may outweigh the benefits of sampling more than one nuclide.

#### 4.2.2 Group-to-Group Scattering - Monte Carlo Approach

Chapter 3 described the format of the scattering laws that determine the outgoing neutron energy and direction in the lab system. The scattering cross sections that are available on the MCNP data libraries simply represent the probability of a scattering event occurring. The individual scattering laws that are associated with the cross sections actually determine the energy and angular distribution of the outgoing particles. Because of this, it is not straightforward to determine the group-to-group reaction rate.

However, it is possible to use some of the MCNP subroutines to simulate scattering events and thereby estimate the group-to-group scattering rates in a Monte Carlo fashion. Described in Section 4.1, steps 5, 6, and 7 control the calculation of the group-to-group scattering reaction rates. These steps are very similar to the steps used during particle transport to simulate a scattering event. Step 5 indicates that either a single nuclide is chosen for the scattering simulation or, since particle tracking is not involved, all nuclides are used. Both methods will be described.

Chapter 5 describes an alternative to this Monte Carlo approach that offers improved statistical convergence.

#### **4.2.2.1 Single Nuclide**

When a single nuclide is being used for the scattering simulation, the datum that will be stored in the appropriate group-to-group scattering bin is the total scattering reaction rate. This value is:

$$\text{WTL}\Sigma_s = \text{WTL}(\Sigma_t - \Sigma_a - \Sigma_f)$$

The first step is to choose an individual nuclide for performing the scattering simulation. This choice is based on the individual total scattering cross sections. The probability of choosing a specific nuclide is:

$$\frac{n^j \sigma_s^j}{\sum_{j=1}^J n^j \sigma_s^j} = \frac{n^j (\sigma_t^j - \sigma_a^j - \sigma_f^j) \rho}{\Sigma_t - \Sigma_a - \Sigma_f}$$

After the nuclide is chosen, steps 6 and 7 are performed. A scattering reaction is randomly chosen based on the individual scattering cross sections and the appropriate scattering law is also chosen. The scattering law is used to simulate a scattering event and determine what the outgoing particle energy would be if this event were real. The incoming energy (the energy of the particle track) and the outgoing energy, together, determine which group-to-group scattering bin the reaction rate data is stored in.

This Monte Carlo approach provides a tally in only one of the possible group-to-group scattering bins for each track length. Therefore, the statistical uncertainty associated with some of the group-to-group scattering bins may be quite large. The method will, however, produce accurate results if enough particles are transported to sample sufficiently the scattering laws and incoming and outgoing energies.

If the scattering reaction that is sampled happens to be one that has more neutrons emerging than entering (ex.  $n,2n$ ), the scattering law is sampled as many times as there are outgoing neutrons. The scattering rate is then stored in the appropriate group-to-group scattering bins according to the outgoing energies. By this method, the multiplicity of the reaction is accounted for in the multigroup scattering rate as described in equation 2-7.

#### **4.2.2.2 All Nuclides**

An alternative approach to using a single nuclide for estimating the scattering rate is to use all nuclides in the material. When this is done, the value that is stored in the appropriate group-to-group bin will depend on the nuclide being sampled. This value is equal to:

$$WTLn^j \sigma_s^j \rho = WTLn^j (\sigma_t^j - \sigma_a^j - \sigma_f^j) \rho$$

Since all nuclides are being sampled, the total contribution being made to the scattering rate is

$$WTL\Sigma_s = \sum_{j=1}^J WTLn^j \sigma_s^j \rho$$

which is identical to the value stored in Section 4.2.2.1. Therefore, the total information being stored is the same whether a single nuclide is sampled or all nuclides are sampled. The difference is in where the information is stored.

By sampling all nuclides, a contribution to the scattering rate would be obtained from every nuclide thereby improving the statistical errors associated with the group-to-group scattering cross sections. The computer run time associated with calculating the scattering contribution would increase proportionally to the number of nuclides in the material.

The sampling of the scattering event for each nuclide and the storage of the data is performed in the same manner as described in the previous sub-section.

### **4.3 Calculating Relative Errors**

In MCNP, the statistical uncertainty associated with a tallied quantity is expressed in terms of the relative error. By definition, the relative error is the standard deviation divided by the mean (calculated value). Since it is helpful to know the uncertainty associated with a calculated value, the relative errors associated with the multigroup cross sections are calculated.

The following abbreviated derivation concludes with equation 4-1 which is the formula to calculate the relative error for a normal tally within MCNP.

$$S_x^2 \equiv \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2$$

$S_x$  ≡ the standard deviation of the population

$$\bar{x} \equiv \frac{\sum_{i=1}^N x_i}{N}$$

$$S_{\bar{x}}^2 \equiv \frac{1}{N} S_x^2$$

$S_{\bar{x}}$  ≡ the standard deviation of the mean

$$R_{\bar{x}}^2 \equiv \frac{S_{\bar{x}}^2}{\bar{x}^2}$$

$R_{\bar{x}}$  ≡ relative error of the mean

Therefore

$$S_{\bar{x}}^2 = \frac{1}{N-1} (\bar{x}^2 - \bar{x}^2)$$

$$R_{\bar{x}}^2 = \frac{1}{N-1} \left( \frac{\left( \sum_{i=1}^N x_i^2 \right) N}{\left( \sum_{i=1}^N x_i \right)^2} - 1 \right)$$

and as  $N$  becomes large

$$R_{\bar{x}}^2 \approx \left( \frac{\sum_{i=1}^N x_i^2}{\left( \sum_{i=1}^N x_i \right)^2} - \frac{1}{N} \right)$$

Finally:

$$R_{\bar{x}} = \sqrt{\frac{\sum_{i=1}^N x_i^2}{\left( \sum_{i=1}^N x_i \right)^2} - \frac{1}{N}} \quad (4-1)$$

The final calculation of the multigroup cross sections depends on two quantities: the flux and the macroscopic reaction rate. The latter is divided by the former to calculate the cross sections. Since both of these quantities are statistical in nature and are not independent, the calculation of the relative error must account for the correlation. Equation 4-5, which is derived below, is the formula used for calculating this relative error.

Since the desired quantity is a function of two variables, the formula for calculating the standard deviation must account for the deviations in these variables. A brief derivation of a general formula for estimating the standard deviation is provided. This derivation was taken from reference 7.

Given

$$\bar{x} = f(\bar{u}, \bar{v})$$

and

$$x_i = f(u_i, v_i)$$

The variance of the final quantity is

$$S_x^2 \equiv \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2 \quad (4-2)$$

In order to estimate this variance, the deviations in the individual variables must be accounted for. This is done by expressing the deviations  $x_i - \bar{x}$  in terms of the deviations  $u_i - \bar{u}$  and  $v_i - \bar{v}$  of the observed parameters.

$$x_i - \bar{x} \equiv (u_i - \bar{u}) \left( \frac{\partial \bar{x}}{\partial \bar{u}} \right) + (v_i - \bar{v}) \left( \frac{\partial \bar{x}}{\partial \bar{v}} \right) \quad (4-3)$$

The partial derivatives are proportionality constants between changes in the desired value,  $x$ , and the individual variables for infinitesimally small changes in these variables.

Combining equations 4-3 and 4-2, the variance of  $x$  can be expressed in terms of the individual variables  $u$  and  $v$ .

$$S_x^2 \equiv \frac{1}{N-1} \sum_{i=1}^N \left[ (u_i - \bar{u}) \left( \frac{\partial \bar{x}}{\partial u} \right) + (v_i - \bar{v}) \left( \frac{\partial \bar{x}}{\partial v} \right) \right]^2$$

$$S_x^2 \equiv \frac{1}{N-1} \sum_{i=1}^N \left[ (u_i - \bar{u})^2 \left( \frac{\partial \bar{x}}{\partial u} \right)^2 + (v_i - \bar{v})^2 \left( \frac{\partial \bar{x}}{\partial v} \right)^2 + 2(u_i - \bar{u})(v_i - \bar{v}) \left( \frac{\partial \bar{x}}{\partial u} \right) \left( \frac{\partial \bar{x}}{\partial v} \right) \right]$$

Using

$$S_u^2 \equiv \frac{1}{N-1} \sum_{i=1}^N (u_i - \bar{u})^2$$

and

$$S_v^2 \equiv \frac{1}{N-1} \sum_{i=1}^N (v_i - \bar{v})^2$$

and defining the covariance

$$S_{uv}^2 \equiv \frac{1}{N-1} \sum_{i=1}^N [(u_i - \bar{u})(v_i - \bar{v})]$$

yields

$$S_x^2 = S_u^2 \left( \frac{\partial \bar{x}}{\partial u} \right)^2 + S_v^2 \left( \frac{\partial \bar{x}}{\partial v} \right)^2 + 2 \times S_{uv}^2 \left( \frac{\partial \bar{x}}{\partial u} \right) \left( \frac{\partial \bar{x}}{\partial v} \right) \quad (4-4)$$

which is the general formula for estimating the associated standard deviation of the population when the desired value is a function of two variables. This formula is then used to derive equation 4-5.

$$\text{macro cross section} = \frac{\text{reaction rate}}{\text{flux}} ; \bar{x} = \frac{\bar{u}}{\bar{v}}$$

$$\frac{\partial \bar{x}}{\partial u} = \frac{1}{\bar{v}} ; \frac{\partial \bar{x}}{\partial v} = -\frac{\bar{u}}{\bar{v}^2}$$

$$S_x^2 = \frac{1}{N-1} \left[ \left( \sum_{i=1}^N (u_i - \bar{u})^2 \right) \left( \frac{1}{\bar{v}} \right)^2 + \left( \sum_{i=1}^N (v_i - \bar{v})^2 \right) \left( -\frac{\bar{u}}{\bar{v}^2} \right)^2 + 2 \left( \sum_{i=1}^N [(u_i - \bar{u})(v_i - \bar{v})] \right) \left( \frac{1}{\bar{v}} \right) \left( -\frac{\bar{u}}{\bar{v}^2} \right) \right]$$

$$S_x^2 = \frac{N}{N-1} \left[ \left( \frac{1}{\bar{v}^2} \right) (\bar{u}^2 - \bar{u}^2) + \left( \frac{\bar{u}^2}{\bar{v}^4} \right) (\bar{v}^2 - \bar{v}^2) - 2 \left( \frac{\bar{u}}{\bar{v}^3} \right) (\bar{u}\bar{v} - \bar{u} \times \bar{v}) \right]$$

$$S_{\bar{x}}^2 = \frac{1}{N-1} \left[ \left( \frac{1}{\bar{v}^2} \right) (\bar{u}^2 - \bar{u}^2) + \left( \frac{\bar{u}^2}{\bar{v}^4} \right) (\bar{v}^2 - \bar{v}^2) - 2 \left( \frac{\bar{u}}{\bar{v}^3} \right) (\bar{u}\bar{v} - \bar{u} \times \bar{v}) \right]$$

$$R_{\bar{x}}^2 = \frac{S_{\bar{x}}^2}{\bar{x}^2} = \frac{S_{\bar{x}}^2}{\left(\frac{\bar{u}}{\bar{v}}\right)^2}$$

$$R_{\bar{x}}^2 = \frac{1}{N-1} \left[ \frac{\bar{u}^2}{\bar{u}^2} + \frac{\bar{v}^2}{\bar{v}^2} - 2 \left( \frac{\bar{u}\bar{v}}{\bar{u} \times \bar{v}} \right) \right]$$

$$R_{\bar{x}}^2 = \frac{N}{N-1} \left[ \frac{\sum_{i=1}^N u_i^2}{\left( \sum_{i=1}^N u_i \right)^2} + \frac{\sum_{i=1}^N v_i^2}{\left( \sum_{i=1}^N v_i \right)^2} - 2 \left( \frac{\sum_{i=1}^N (u_i v_i)}{\sum_{i=1}^N u_i \sum_{i=1}^N v_i} \right) \right]$$

and as N becomes large

$$R_{\bar{x}}^2 \approx \frac{\sum_{i=1}^N u_i^2}{\left( \sum_{i=1}^N u_i \right)^2} + \frac{\sum_{i=1}^N v_i^2}{\left( \sum_{i=1}^N v_i \right)^2} - 2 \left( \frac{\sum_{i=1}^N (u_i v_i)}{\sum_{i=1}^N u_i \sum_{i=1}^N v_i} \right)$$

Finally:

$$R_{\bar{x}} = \sqrt{\frac{\sum_{i=1}^N u_i^2}{\left( \sum_{i=1}^N u_i \right)^2} + \frac{\sum_{i=1}^N v_i^2}{\left( \sum_{i=1}^N v_i \right)^2} - 2 \left( \frac{\sum_{i=1}^N (u_i v_i)}{\sum_{i=1}^N u_i \sum_{i=1}^N v_i} \right)} \quad (4-5)$$

If all  $v_i$  are identical

$$v_i = c$$

then

$$R_{\bar{x}} = \sqrt{\frac{\sum_{i=1}^N u_i^2}{\left(\sum_{i=1}^N u_i\right)^2} + \frac{\sum_{i=1}^N c^2}{\left(\sum_{i=1}^N c\right)^2} - 2 \left( \frac{\sum_{i=1}^N (u_i c)}{\sum_{i=1}^N u_i \sum_{i=1}^N c} \right)}$$

$$R_{\bar{x}} = \sqrt{\frac{\sum_{i=1}^N u_i^2}{\left(\sum_{i=1}^N u_i\right)^2} + \frac{Nc^2}{(Nc)^2} - 2 \left( \frac{c \sum_{i=1}^N u_i}{Nc \sum_{i=1}^N u_i} \right)}$$

$$R_{\bar{x}} = \sqrt{\frac{\sum_{i=1}^N u_i^2}{\left(\sum_{i=1}^N u_i\right)^2} + \frac{1}{N} - 2 \frac{1}{N}}$$

$$R_{\bar{x}} = \sqrt{\frac{\sum_{i=1}^N u_i^2}{\left(\sum_{i=1}^N u_i\right)^2} - \frac{1}{N}}$$

Thus if there is no variation in the  $v_i$  equation 4-5 reduces to equation 4-1 as it should.

At the end of each particle history, the temporary cross section storage array is summed into the primary storage array. At this time the following quantities are also calculated and summed into the primary storage array.

- the square of the flux estimation for that particle history
- the square of the reaction rate estimation for that particle history
- the product of the flux estimation and the reaction rate estimation

These quantities are used in equation 4-5 at the end of the MCNP run to calculate the relative error associated with the multigroup cross sections.

#### 4.4 The Random Number Sequence

The Monte Carlo approach to estimating the chi and group-to-group scattering, as described in Section 4.2, uses random numbers to select the nuclides and sample the

scattering laws. This use of random numbers will alter the random number sequence in the calculation such that the final answers (ex. k-eff) will be statistically equivalent, but not numerically identical, to the answers from the same MCNP run when the cross section generation is not performed.

During the development of the MCNP modifications it was essential to obtain the same answers from a calculation with or without the cross section generation. This was needed as a way of verifying that the modifications did not inadvertently alter the program in an unforeseen fashion. In order to achieve this, the modifications to MCNP were programmed such that the location in the random number sequence was saved when the cross section generation began and was reset to the same location when the cross section generation was finished.

There is a potential problem with resetting the random number sequence after the cross section generation is finished. The possibility exists that some of the same random numbers could be used again in the cross section generation when it is next executed. The use of the same random number more than once probably would not affect the final cross sections; however, this can not be guaranteed. Therefore, the modifications have been programmed with a switch that permits the user to decide if the random number sequence should be reset after the cross section generation is finished.

## Chapter 5

### **Group-to-Group Scattering - The Explicit Approach**

Chapter 4 described the Monte Carlo approach to estimating the group-to-group scattering cross sections. This chapter will describe the explicit approach to calculating the scattering cross section. The difference between these approaches is the amount of information obtained for each track length. In the Monte Carlo approach, a contribution to one group-to-group bin is obtained for each track length. In contrast, the explicit approach calculates contributions to all possible group-to-group bins based on the scattering law chosen.

In MCNP, the scattering cross section represents the probability of the incoming neutron undergoing a scattering event. This cross section does not provide any information about the possible outgoing energies. Instead, these energies are determined by the associated scattering laws. It is possible to use these scattering laws to determine the fraction of neutrons that scatter from a given incoming energy into each of the energy groups. These fractions multiplied by the scattering cross sections represent the group-to-group scattering cross sections for neutrons scattering at the incoming energy.

Chapter 4 described the method of estimating the scattering rates using a single nuclide or all nuclides in the material. This method is unchanged when using the explicit approach. The only difference between the explicit approach and the Monte Carlo approach described in Section 4.2.2 is the datum stored in the cross section array. Section 4.2.2 shows that the value stored in the Monte Carlo approach is

$$WTL\Sigma_s = WTL(\Sigma_t - \Sigma_a - \Sigma_f)$$

when a single nuclide is sampled and

$$WTLn^j\Sigma_s^j\rho = WTLn^j(\Sigma_t^j - \Sigma_a^j - \Sigma_f^j)\rho$$

when all nuclides are sampled.

In the Monte Carlo approach, this datum is stored in a single group-to-group bin determined by the associated scattering law. In the explicit approach, a fraction of this datum is stored in various group-to-group bins determined by the scattering laws.

In the explicit approach, the datum stored when a single nuclide is sampled is:

$$WTL \Sigma_s f_k = WTL(\Sigma_t - \Sigma_a - \Sigma_f) f_k$$

$f_k$  = the fraction of the particles at the incoming energy scattering into group k

When all nuclides are sampled in the explicit approach, the datum stored in each group-to-group scattering bin is:

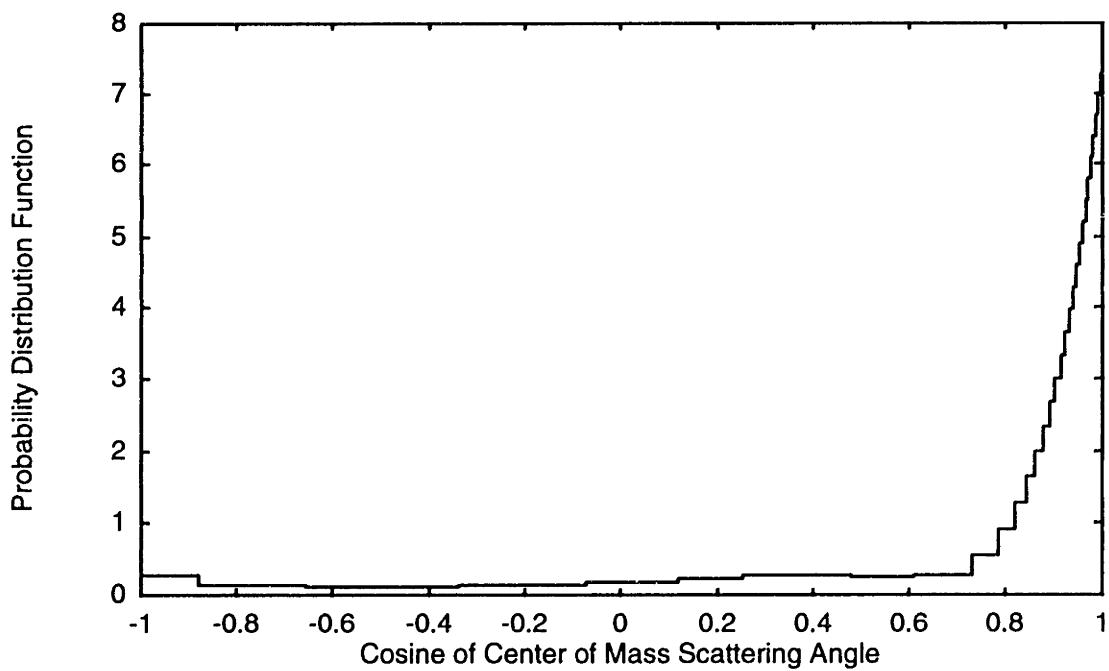
$$WTL n^j \sigma_s^j \rho f_k = WTL n^j (\sigma_t^j - \sigma_a^j - \sigma_f^j) \rho f_k$$

$f_k$  = the fraction of the particles at the incoming energy scattering into group k

The remainder of this chapter will describe how the fractions,  $f_k$ , are determined for each of the scattering laws discussed in Chapter 3.

## 5.1 Elastic Scattering - Target at Rest

Section 3.5 shows the relationship between the scattering angle in the center of mass system and the outgoing energy of the neutron for elastic scattering. The possible scattering angles are represented by thirty-two equi-probable cosine bins in the center of mass system. These bins comprise the probability distribution function (PDF) for scattering at the incoming energy. An example of a PDF for anisotropic scattering in the center of mass system is shown in Figure 5-1. By definition, the integral of the PDF is one. Since each center of mass angle directly relates to an outgoing energy, the PDF can be integrated to determine the fraction,  $f_k$ , of outgoing energies that belong to a certain group.



**Figure 5-1:** Probability distribution function for anisotropic scattering in the center of mass system.

In order to perform this integration, equation 3-1 must be modified to express the center of mass scattering angle in terms of outgoing energy. Equation 5-1 is this relationship.

$$\mu_c = \frac{\frac{E(A+1)^2}{E'} - (A^2 + 1)}{2A} \quad (5-1)$$

$E$  = outgoing particle energy

$E'$  = incoming particle energy

$$A = \frac{M_t}{M_n}$$

$M_t$  = mass of target

$M_n$  = mass of neutron

$\mu_c$  = cosine of center of mass scattering angle

An example of this integration technique will be useful in understanding the method.

In this example there are five energy groups:

Group	Bottom Energy	Top Energy (eV)
1	0.1	0.225
2	0.05	0.1
3	0.03	0.05
4	0.01	0.03
5	0.0	0.01

The incoming neutron energy is 0.04315 eV and therefore this particle belongs to group 3. The target nucleus is Hydrogen and the A value is 0.999167.

The probability distribution function used in this example is shown in Figure 5-1.

The first step in the integration is to determine the outgoing energy corresponding to the highest and lowest center of mass scattering angles using equation 3-1.

Center of Mass Angle	Outgoing Energy (eV)	Group
-1	7.49e-09	5
1	0.04315	3

With this information it is obvious that scattering can occur from starting group 3 to groups 3, 4, and 5. Equation 5-1 is then used to determine the center of mass angles that correspond to the group boundaries for groups 3, 4, and 5.

Using these angles the integration of the PDF is performed to determine the fraction of outgoing energies that belong to a specific group-to-group transfer. Figure 5-2 graphically shows the results of the integration.

The areas shown in Figure 5-2 are the  $f_k$  needed to determine the data that will be stored in the group-to-group bins.

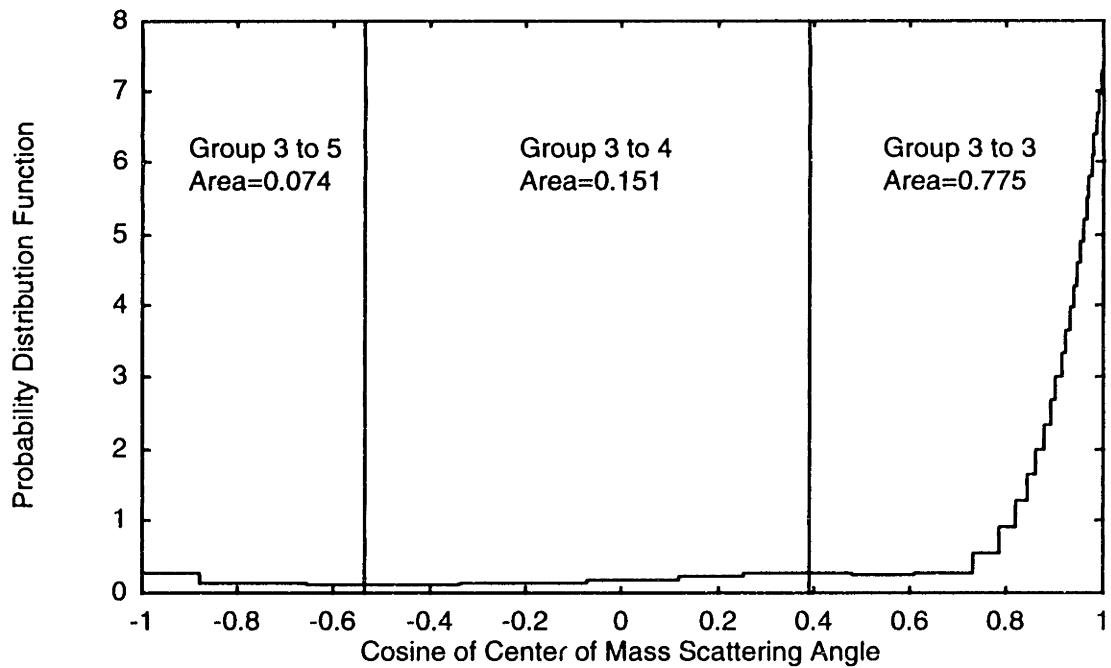


Figure 5-2: Probability distribution function for anisotropic scattering in the center of mass system. The integration of the distribution to determine the fractional area corresponding to specific group-to-group transfers is shown. This integration is performed for the case of elastic scattering with the target-at-rest.

## 5.2 Elastic Scattering - Target in Motion

Section 3.2 described the method used during particle tracking to simulate scattering with a target nucleus in motion. In summary, the relative energy and direction of the neutron are calculated and the scattering is simulated with the target-at-rest. The new relative energy and direction are adjusted to the lab system by adding back the energy and direction of the target.

This method works very well for particle transport but is not readily adaptable to cross section generation. In order to perform the explicit approach as described in Section 5.1, the relationship between center of mass scattering angle, incoming and outgoing energy, and laboratory scattering angle must be known for elastic scattering with the target-in-motion. This relationship is not directly available in the MCNP source code.

Reference 8 provides a detailed discussion of these relationships for elastic scattering with a target-in-motion. Figure 5-3 depicts the relationship between the velocity vectors of the neutron and target in both the laboratory system and the center of mass system.

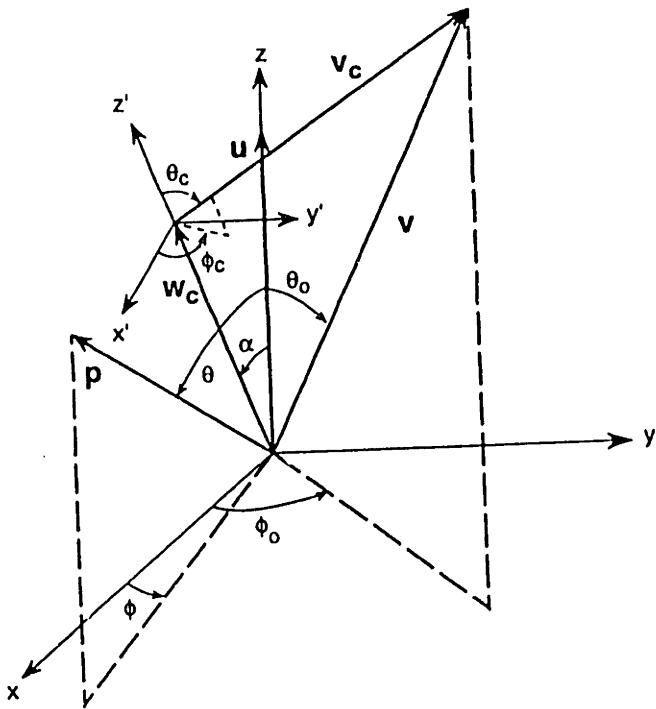


Figure 5-3: Relation of the velocity vectors in the lab system after a scattering event. The vectors  $\mathbf{u}$ ,  $\mathbf{p}$ , and  $\mathbf{w}_c$  are coplanar. This figure was borrowed from reference 8.

In Figure 5-3 the following definitions apply:

$\mathbf{u}$  = incoming neutron velocity

$\mathbf{v}$  = outgoing neutron velocity

$\mathbf{p}$  = target velocity

$\mathbf{w}_c$  = center of mass velocity

$\mathbf{v}_c$  = center of mass velocity of neutron after collision

$\theta_c$  = center of mass scattering angle

$\mu_c \equiv \cos(\theta_c)$

$\phi_c$  = azimuthal scattering angle

$\theta$  = angle between neutron and target

$$\mu_t \equiv \cos(\theta)$$

$\theta_0$  = lab scattering angle

$$\mu_L \equiv \cos(\theta_0)$$

Reference 8 derives the following relationships.

$$w_c = \frac{\sqrt{u^2 + A^2 p^2 + 2Aup\mu_t}}{A+1} \quad (5-2)$$

$$v_c = \frac{A}{A+1} v_r \equiv \frac{A}{A+1} \sqrt{u^2 + p^2 - 2up\mu_t} \quad (5-3)$$

$$(A+1)w_c \cos(\alpha) = u + Ap\mu_t \quad (5-4)$$

$$\mu_L = \frac{\cos(\alpha)(w_c + v_c\mu_c) - \sin(\alpha)v_c\sqrt{1-\mu_c^2} \cos(\phi_c)}{\sqrt{w_c^2 + v_c^2 + 2w_c v_c \mu_c}} \quad (5-5)$$

Using Figure 5-3, the equation defining the outgoing velocity can be written.

$$v^2 = w_c^2 + v_c^2 + 2w_c v_c \mu_c \quad (5-6)$$

In order for these equations to be useful they must be expressed in terms of energy rather than velocity.

Using

$$E' = \frac{1}{2} M_n u^2$$

$$E = \frac{1}{2} M_n v^2$$

$$E_t = \frac{1}{2} M_t p^2$$

equations 5-2 and 5-3 can be written as

$$w_c = \sqrt{\frac{2}{M_n}} \frac{\sqrt{E' + AE_t + 2\mu_t \sqrt{AE'E_t}}}{A+1}$$

$$v_c = \sqrt{\frac{2}{M_t}} \frac{A}{A+1} \sqrt{AE' + E_t - 2\mu_t \sqrt{AE'E_t}}$$

If the following are defined

$$B \equiv E' + AE_t + 2\mu_t \sqrt{AE'E_t}$$

$$C \equiv AE' + E_t - 2\mu_t \sqrt{AE'E_t}$$

equations 5-2 and 5-3 can be simplified to

$$w_c = \sqrt{\frac{2}{M_n}} \frac{\sqrt{B}}{A+1} \quad (5-7)$$

$$v_c = \sqrt{\frac{2}{M_t}} \frac{A}{A+1} \sqrt{C} \quad (5-8)$$

Equation 5-4 can now be simplified and reworked to provide

$$\cos(\alpha) = \frac{\sqrt{E'} + \mu_t \sqrt{AE_t}}{\sqrt{B}}$$

Defining

$$D \equiv \sqrt{E'} + \mu_t \sqrt{AE_t}$$

yields

$$\cos(\alpha) = \frac{D}{\sqrt{B}} \quad (5-9)$$

$$\sin(\alpha) = \sqrt{1 - \frac{D^2}{B}} \quad (5-10)$$

Using equations 5-7 and 5-8, equation 5-6 can be rewritten to provide the following equations which relate outgoing energy and center of mass scattering angle.

$$E = \frac{1}{(A+1)^2} (B + AC + 2\mu_c \sqrt{ABC}) \quad (5-11)$$

$$\mu_c = \frac{(A+1)^2 E - B - AC}{2\sqrt{ABC}} \quad (5-12)$$

Using equations 5-7 through 5-10, equation 5-5 can be rewritten in a simplified form.

$$\mu_L = \frac{D \left( 1 + \frac{\mu_c \sqrt{AC}}{\sqrt{B}} \right) - \sqrt{1 - \frac{D^2}{B}} \sqrt{AC} \sqrt{1 - \mu_c^2} \cos(\phi_c)}{\sqrt{B + AC + 2\mu_c \sqrt{ABC}}} \quad (5-13)$$

Equations 5-11 through 5-13 are the general equations that relate incoming and outgoing energy and the scattering angles in the lab and center of mass systems. When the target is at rest:

$$E_t = 0.0$$

$$\mu_t = 1.0$$

The terms B, C, and D then reduce to

$$B \equiv E'$$

$$C \equiv AE'$$

$$D \equiv \sqrt{E'}$$

and equations 5-11 through 5-13 reduce to

$$E = \frac{1}{(A+1)^2} E' (1 + 2A\mu_c + A^2)$$

$$\mu_c = \frac{(A+1)^2 E - E'(1+A^2)}{2AE'} = \frac{\frac{E}{E'}(A+1)^2 - (A^2 + 1)}{2A}$$

$$\mu_L = \frac{\sqrt{E'}(1+A\mu_c)}{\sqrt{E'(1+A^2+2A\mu_c)}} = \frac{1+A\mu_c}{\sqrt{1+A^2+2A\mu_c}}$$

These equations are identical to equations 5-1, 3-1 and 3-2 as they should be.

Section 5-1 described, by example, the steps involved in the explicit approach to determine the group-to-group scattering rates for the case of elastic scattering with the target-at-rest. In the case of target-in-motion the steps described in Section 5.1 are still used except equations 5-11 and 5-12 are used in place of equations 3-1 and 5-1.

There is one problem that arises when equations 5-11, 5-12, and 5-13 are used. These equations depend on knowing the following information.

- Energy of the target nucleus -  $E_t$
- The angle between the target and the neutron -  $\mu_t$
- The azimuthal angle  $\phi_c$

The solution to this problem is to randomly select these values before the explicit approach is applied. The energy of the target and the angle between the target and neutron are determined using the same subroutine MCNP uses to calculate these values during transport. The azimuthal angle is randomly selected between 0 and  $2\pi$ . Once these values are available the explicit approach can be performed and the PDF can be integrated to determine the fraction of neutrons,  $f_k$ , scattering into the various groups.

An example of this integration is provided here for better understanding of the method. The following five groups are used in this example.

Group	Bottom Energy (eV)	Top Energy (eV)
1	0.1	0.225
2	0.05	0.1
3	0.03	0.05
4	0.01	0.03
5	0.0	0.01

The energy of the incoming neutron is 0.04315 eV, group 3, and the probability distribution function for scattering in the center of mass system is provided in Figure 5-1. The target nucleus is Hydrogen and the A value is 0.999167.

The following values were randomly chosen.

- Energy of the target nucleus = 0.03764 eV
- Cosine of the angle between the target and the neutron = -0.0951

Using these values and equation 5-11, the outgoing energy corresponding to the lowest and highest center of mass scattering angles are determined. These values indicate the possible groups that can be reached by a neutron at this incoming energy scattering with a target defined by the properties listed above.

Center of Mass Angle	Outgoing Energy (eV)	Group
-1	1.81e-08	5
1	0.0806	2

With this information it is obvious that scattering, including up-scattering, can occur from starting group 3 to groups 2, 3, 4, and 5. Equation 5-12 is then used to determine the center of mass angles that correspond to the group boundaries for groups 2, 3, 4, and 5.

Using these angles the integration of the PDF is performed to determine the fraction of outgoing energies that belong to a specific group-to-group transfer. Figure 5-4 graphically shows the results of the integration.

The areas shown in Figure 5-4 are the  $f_k$  needed to determine the data that will be stored in the group-to-group bins.

In essence, by randomly choosing the target parameters, the calculation of the  $f_k$  for the case of elastic scattering with the target-in-motion is a hybrid Monte Carlo/explicit approach.

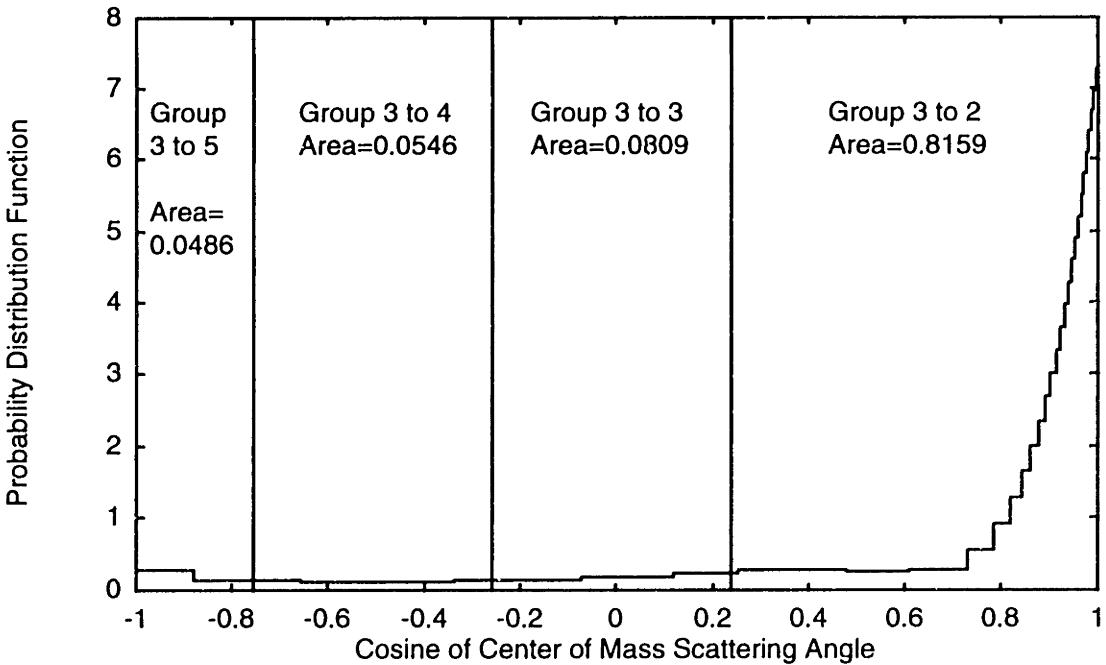


Figure 5-4: Probability distribution function for anisotropic scattering in the center of mass system. The integration of this distribution to determine the fractional area corresponding to specific group-to-group transfers is shown. This integration is performed for the case of elastic scattering with the target-in-motion.

### 5.3 Law 1 - Tabular Equi-Probable Energy Bins

Section 3.5 describes the mechanics of scattering law 1. In this law, the outgoing energy is determined by scaled linear interpolation between sets of equi-probable outgoing energy bins.

Since the possible outgoing energies are provided in a table of equi-probable energy bins, the determination of the fraction of neutrons scattering into a particular energy group is a relatively simple matter. This table is integrated, using the energy group boundaries as limits of integration, to calculate the  $f_k$ .

Before the integration can be performed, the table of possible outgoing energies must be calculated using scaled linear interpolation between adjacent tables. The calculation of this table is very similar to the calculation of a single outgoing energy as discussed in Section 3.5

Figure 5-5 is a graphical representation of two adjacent outgoing energy tables for law 1. The energy table shown in the middle is the scaled linearly interpolated table corresponding to the incoming neutron energy,  $E_n$ .

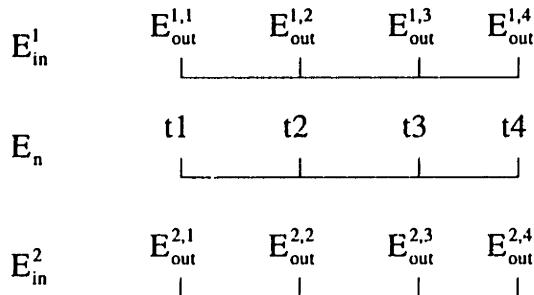


Figure 5-5: Illustration of three equi-probable outgoing energy bins for the two incoming energies shown. The outgoing energy bins corresponding to the neutron energy are shown in the middle of the figure.

The first step in determining the outgoing energy bins is to calculate the following terms.

$$r = \frac{E_n - E_{in}^1}{E_{in}^2 - E_{in}^1}$$

$r1$  = random number

$$t1 = E_{out}^{1,1} + r \times (E_{out}^{2,1} - E_{out}^{1,1})$$

$$t4 = E_{out}^{1,4} + r \times (E_{out}^{2,4} - E_{out}^{1,4})$$

The outgoing energy bins associated with  $E_{in}^1$  will be used for the final calculation if  $r1 \leq r$ , otherwise the outgoing energy bins associated with  $E_{in}^2$  will be used. For this example, assume the second set of outgoing energies is chosen. Finally the outgoing energy bin boundaries are calculated using the following equation.

$$t_i = t1 + (t4 - t1) \times \frac{E_{out}^{2,i} - E_{out}^{2,1}}{E_{out}^{2,4} - E_{out}^{2,1}} \quad (5-14)$$

where  $1 < i < 4$

Once these outgoing energy bin boundaries are known the integration to determine the  $f_k$  is performed and the data are stored in the appropriate group-to-group scattering bins.

## 5.4 Law 3 - Level Scattering

Section 3.6 described law 3 and presented the equations that define this law. As in elastic scattering, the data for this law consist of a table of 32 equi-probable cosine bins, a PDF, in the center of mass system. Since there is a direct relationship between outgoing energy and center of mass scattering angle for this law, the method of integrating the PDF to determine the  $f_k$  values is identical to the method described in Section 5.1.

In order to perform the integration, the equations relating outgoing energy to center of mass scattering angle must be known. Equation 3-4 is the first of these equations, expressing the outgoing energy in terms of the center of mass angle. The corollary to this equation, and the last equation needed, is:

$$\mu_c = \frac{(E_{out}^{lab} - E_{out}^{cm})(A + 1)^2 - E'}{2(A + 1)\sqrt{E'E_{out}^{cm}}} \quad (5-15)$$

## 5.5 Law 9 - Evaporation Spectrum

Equation 3-6 is the probability distribution function for the outgoing energy of a particle after a scattering event has occurred. The PDF can be integrated to yield the cumulative distribution function (CDF) which is shown below. The CDF can then be evaluated at the different energy group boundaries to determine the  $f_k$  for scattering from the incoming energy into group k.

$$T \equiv T(E')$$

$$CDF = \int f(E' \rightarrow E)dE = \frac{-1}{C} e^{\frac{-E}{T}} (E + T)T \quad (5-16)$$

$$0 \leq E \leq E' - U$$

$$U = \text{restriction energy}$$

$$C = T^2 \left[ 1 - e^{-\frac{-(E'-U)}{T}} \left( 1 + \frac{E' - U}{T} \right) \right]$$

## 5.6 Law 11 - Energy Dependent Watt Spectrum

Equation 3-8 is the probability distribution function for the outgoing energy of a particle after a fission event has occurred. The PDF can be integrated to yield the cumulative distribution function (CDF) which is shown below. The CDF can then be evaluated at the different energy group boundaries to determine the fraction of outgoing neutrons that enter group k,  $f_k$ .

$$a \equiv a(E')$$

$$b \equiv b(E')$$

$$\begin{aligned} \text{CDF} = \int f(E' \rightarrow E) dE &= \frac{1}{C} \left[ -a e^{-\frac{E}{a}} \sinh(\sqrt{bE}) \right] \\ &+ \frac{1}{C} \left\{ \frac{1}{2} \sqrt{\frac{\pi a^3 b}{4}} e^{\frac{ab}{4}} \left[ \operatorname{erf}\left(\sqrt{\frac{E}{a}} - \sqrt{\frac{ab}{4}}\right) + \operatorname{erf}\left(\sqrt{\frac{E}{a}} + \sqrt{\frac{ab}{4}}\right) \right] \right\} \end{aligned} \quad (5-17)$$

$$0 \leq E \leq E' - U$$

$$U = \text{restriction energy}$$

$$\begin{aligned} C &= -a e^{-\frac{-(E'-U)}{a}} \sinh(\sqrt{b(E' - U)}) \\ &+ \frac{1}{2} \sqrt{\frac{\pi a^3 b}{4}} e^{\frac{ab}{4}} \left[ \operatorname{erf}\left(\sqrt{\frac{(E' - U)}{a}} - \sqrt{\frac{ab}{4}}\right) + \operatorname{erf}\left(\sqrt{\frac{(E' - U)}{a}} + \sqrt{\frac{ab}{4}}\right) \right] \end{aligned}$$

## 5.7 Fission Chi

As discussed in Sections 3.7 and 3.8, laws 9 and 11 are used for determining outgoing energies from both scattering events and fission events. Therefore, the CDFs can also be used to determine the fraction of fission neutrons that would be born in a given group. This term will also be referred to as  $f_k$ .

In Section 4.2.1, the Monte Carlo approach to estimating the chi value was discussed. If the  $f_k$  are known, a contribution to all possible groups can be made when estimating the

chi value. This is the explicit approach to estimating the chi value. The primary difference between the Monte Carlo approach and the explicit approach is the datum that is stored. Sections 4.2.1.1 and 4.2.1.2 describe the data that are stored in the Monte Carlo approach when single nuclide and all nuclides are used for estimating the chi value.

Using the explicit approach the following datum is stored in the group k bin when a single nuclide is sampled.

$$WTL\Sigma_f v f_k$$

When all nuclides are used in estimating the chi value, the following datum, for nuclide j, is stored in the group k bin.

$$WTLn^j \sigma_f^j \rho v^j f_k$$

$\rho$  = atom density of material

## 5.8 S( $\alpha, \beta$ ) Scattering

Section 3.3 described one of the ways the S( $\alpha, \beta$ ) data can be expressed on the library file. The described data format consists of a discrete number of outgoing energies for various incoming energies. These outgoing energies may or may not be equi-probable.

Since there are a discrete number of outgoing energies, instead of a distribution, it is very easy to determine the fraction of the scattering rate that would correspond to a specific group-to-group transfer. Using the interpolation technique described in Section 3.3, all possible outgoing energies are calculated. Once these energies are known, the fraction,  $f_k$ , of the outgoing energies representing a specific group-to-group transfer is calculated. The following expression defines this fraction.

$P(E_{out}^i)$  = probability of  $E_{out}^i$  being chosen

$f_k = \sum P(E_{out}^i)$  for all  $E_{out}^i$  that belong to group k

## 5.9 Other Scattering Laws

The scattering laws discussed in the previous sections are not all of the scattering laws used in MCNP. However, these laws are frequently used in the continuous energy data based on the ENDF/B-V libraries and therefore were the only scattering laws that were modified for the explicit approach.

In the cross section generation modifications, when a scattering law is used, other than one of the laws mentioned above, the Monte Carlo approach, as described in Chapter 4, is used to estimate the group-to-group scattering rate. This is acceptable for the present thesis. The other scattering laws can easily be modified at a later time for the explicit approach to estimating the group-to-group scattering rate.

There is one exception to this rule: scattering law 44, used with ENDF/B-VI data. It is possible, according to the MCNP source code, for the weight of a particle to be adjusted by this law. Since it is unclear at this time how this may effect the cross section generation, a warning message is printed when this law is encountered.

## Chapter 6

### Legendre Components

The multigroup scattering cross sections that have been discussed do not contain any information about the angular distribution of the emerging particles. This distribution is required to properly represent the nuclear physics of the particle interactions. Otherwise, all scattering would be isotropic in the laboratory system which is not physically correct. Figure 6-1 illustrates a scattering event in the lab system. The angular distribution function in the lab system is the probability distribution of the scattering angle shown in this figure. It is assumed that the azimuthal angular distribution of the scattered particles is always isotropic with respect to the incoming particle direction.

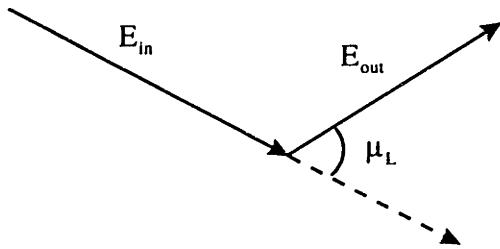


Figure 6-1: Illustration of a scattering event showing the cosine of the laboratory scattering angle.

In multigroup transport methods, the angular distribution is approximated with a Legendre expansion. The following mathematical relationships define the angular distribution of scattered particle and the Legendre expansion used to represent these distributions.

$$f(\mu_0) = \text{laboratory angular distribution of scattering cosines}$$

where

$$\int_{-1}^1 f(\mu_0) d\mu_0 = 1$$

This distribution can be represented with the following expansion in terms of the Legendre polynomials,  $P_n(\mu)$ .

$$f(\mu_0) \approx \sum_{n=0}^N (2n+1)f_n P_n(\mu_0) \quad (6-1)$$

where

$$f_n = \int_{-1}^1 f(\mu_0) P_n(\mu_0) d\mu_0 \quad (6-2)$$

and the Legendre polynomials are defined as

$$P_0(\mu) = 1$$

$$P_1(\mu) = \mu$$

$$(2n+1)\mu P_n(\mu) = (n+1)P_{n+1}(\mu) + nP_{n-1}(\mu)$$

Since the angular distribution represented by equation 6-1 is the distribution of scattering cosines, the multigroup libraries contain the group-to-group scattering cross sections and the Legendre components of these cross sections. The latter are referred to as  $P_n$  cross sections. Typically, multigroup cross section libraries contain scattering cross sections up to an odd Legendre order of  $n=3$  or  $n=5$ . Lower order cross sections,  $n=1$ , are usually not sufficient to represent the angular distribution and higher order cross sections,  $n>5$ , usually do not offer any improved accuracy.

The definition of the  $P_n$  cross sections begins with the transport equation which was shown in Chapter 2. The scattering rate into  $dE$  about  $E$  and  $d\mu$  about  $\mu$ , as shown in the transport equation, is:

$$\int_0^\infty dE' \int_0^{2\pi} \frac{d\varphi'}{2\pi} \int_{-1}^1 d\mu' \Sigma_s(z, E' \rightarrow E, \mu_0) \Psi(z, \mu', E') dEd\mu \quad (6-3)$$

If the integration over  $\mu'$  and  $\varphi'$  is performed while  $\mu_0$  is held constant, equation 6-4 can be written. This is basically the total scattering rate into  $dE$  about  $E$  as a function of  $\mu_0$ . The flux term in equation 6-4 is the energy dependent scalar flux otherwise referred to as the  $P_0$  flux.

$$\int_0^\infty dE' \Sigma_s(z, E' \rightarrow E, \mu_0) \Psi^0(z, E') dE \quad (6-4)$$

The definition for the  $P_n$  cross section is derived by integrating equation 6-3 with respect to  $\mu$ ,  $\mu'$ , and  $\varphi'$  and expanding  $\mu_0$  in terms of  $\mu$  and  $\mu'$ . The integration over incoming and outgoing energies is also performed. Reference 12 defines the resulting equation for the  $P_n$  cross sections as:

$$\Sigma_{sgg'}^n \equiv \frac{\int_{E_g}^{E_{g+1}} dE \int_{E_{g'}}^{E_{g'+1}} dE' \int dz \Sigma_s^n(z, E' \rightarrow E) \Psi^n(z, E')}{\int_{E_{g'}}^{E_{g'+1}} dE' \int dz \Psi^n(z, E')} \quad (6-5)$$

where

$\Sigma_s^n(z, E' \rightarrow E)$  = nth Legendre component of the double differential scattering cross section

$\Psi^n(z, E')$  = nth Legendre component of the directional flux density

and

$$\begin{aligned} \Psi^n(z, E') &= \int_{-1}^1 d\mu' \Psi(z, \mu', E') P_n(\mu') \\ \Sigma_s^n(z, E' \rightarrow E) &= \int_{-1}^1 d\mu_0 \Sigma_s(z, E' \rightarrow E, \mu_0) P_n(\mu_0) \end{aligned} \quad (6-6)$$

Equation 6-5 is the formal definition of the nth Legendre component of the group-to-group scattering cross section. In order to perform the integration shown in the equation, one must be able to calculate the Legendre components of the scattering cross section and the particle flux. The former is not too difficult to do and is discussed below. The latter, however, is practically impossible to calculate beyond a Legendre order of 0. The reason is that if the particle flux is nearly isotropic (which is true in many applications) the Legendre components beyond  $n=0$  will approach zero. This means it would be extremely difficult, if not impossible, to obtain reliable Monte Carlo estimates of these components.

Therefore, the following approximation is made.

$$\Psi^n(z, E') \approx C_n \Psi^0(z, E') = C_n \int_{-1}^1 d\mu' \Psi(z, \mu', E') \quad (6-7)$$

This approximation assumes that the energy dependencies of the  $P_n$  fluxes are proportional to the energy dependence of the  $P_0$  flux. It is believed that this approximation will not introduce significant error in the calculation. The acceptability of this approximation is demonstrated by example in Chapter 8.

Using equations 6-7 and 6-6, equation 6-5 can be rewritten in the following manner where the proportionality constants,  $C_n$ , have been canceled:

$$\Sigma_{sgg'}^n \equiv \frac{\int_{E_g}^{E_{g'+1}} dE \int_{E_{g'}}^{E_{g'+1}} dE' \int_{\Delta z} dz \Psi^0(z, E') \int_{-1}^1 d\mu_0 \Sigma_s(z, E' \rightarrow E, \mu_0) P_n(\mu_0)}{\int_{E_{g'}}^{E_{g'+1}} dE' \int_{\Delta z} dz \Psi^0(z, E')} \quad (6-8)$$

Equation 6-8 also can be written as:

$$\Sigma_{sgg'}^n \equiv \frac{\int_{-1}^1 d\mu_0 \left[ \int_{E_g}^{E_{g'+1}} dE \int_{E_{g'}}^{E_{g'+1}} dE' \int_{\Delta z} dz \Psi^0(z, E') \Sigma_s(z, E' \rightarrow E, \mu_0) \right] P_n(\mu_0)}{\int_{E_{g'}}^{E_{g'+1}} dE' \int_{\Delta z} dz \Psi^0(z, E')} \quad (6-9)$$

Two important points should be mentioned about this equation. The first is that, when  $n=0$ , equation 6-9 is the definition of the group-to-group scattering cross section. In addition, when  $n=0$ , the bracketed term is basically identical to equation 6-4 and represents the group-to-group scattering rate as a function of  $\mu_0$ . The second important point is that the limits of integration over  $\mu_0$  may or may not be -1 to 1 since the scattering angle and the outgoing energy are directly linked by kinematic relationships. However, for convenience, the limits of integration are written as -1 to 1.

The double differential scattering cross section can be written as the product of two functions.

$$\Sigma_s(z, E' \rightarrow E, \mu_0) = \Sigma_s(z, E') f(E' \rightarrow E, \mu_0)$$

where

$f(E' \rightarrow E, \mu_0)$  = the laboratory angular distribution function for scattering at incoming energy  $E'$

and the exact kinematic relationship between the outgoing energy and the scattering angle has been neglected. Using this new definition, equations 6-10 and 6-11 can be written.

$$\Sigma_{sgg'}^n \equiv \frac{\int_{E_g}^{E_{g+1}} dE \int_{E_{g'}}^{E_{g'+1}} dE' \int_{-1}^1 d\mu_0 \Sigma_s(z, E') f(E' \rightarrow E, \mu_0) P_n(\mu_0)}{\int_{E_{g'}}^{E_{g'+1}} dE' \int_{-1}^1 dz \Psi^0(z, E')} \quad (6-10)$$

$$\Sigma_{sgg'}^n \equiv \frac{\int_{-1}^1 d\mu_0 \left[ \int_{E_g}^{E_{g+1}} dE \int_{E_{g'}}^{E_{g'+1}} dE' \int_{-1}^1 dz \Psi^0(z, E') \Sigma_s(z, E') f(E' \rightarrow E, \mu_0) \right] P_n(\mu_0)}{\int_{E_{g'}}^{E_{g'+1}} dE' \int_{-1}^1 dz \Psi^0(z, E')} \quad (6-11)$$

These equations are the definitions of the Legendre components of the scattering cross section that were used in the cross section generation modifications to MCNP. As discussed in Chapter 2, the integrations over space, energy, and  $\mu'$  are performed in a Monte Carlo fashion using summations.

When the Legendre order,  $n$ , is zero, equation 6-10 and 6-11 are the definition of the group-to-group scattering cross section,  $\Sigma_{sgg'}$ . Chapters 4 and 5 described the methods used to calculate this term. Therefore, this chapter will discuss the method of calculating only the higher Legendre components of the scattering cross section. Specifically, the determination of the laboratory angular distribution function and the integration of this function over the variable  $\mu_0$  will be discussed.

## 6.1 An Example

The easiest way to discuss the various methods of calculating the angular distribution function and the Legendre components is to use an example. The MENDF5 multigroup library<sup>9</sup> from Los Alamos National Laboratory contains  $P_4$  expansions for various nuclides. Figure 6-2 contains a plot of a  $P_4$  expansion of the laboratory scattering angle distribution for target-at-rest elastic scattering from group 18 to 19 for Deuterium. Group 18 is defined as the interval 9.12 keV to 24.8 keV and group 19 spans 3.35 keV to 9.12 keV. Figure 6-2 also shows a plot of the analytically calculated angular distribution for this group-to-group transfer. The analytical plot was generated with a small program<sup>11</sup> obtained from the author of reference 10.

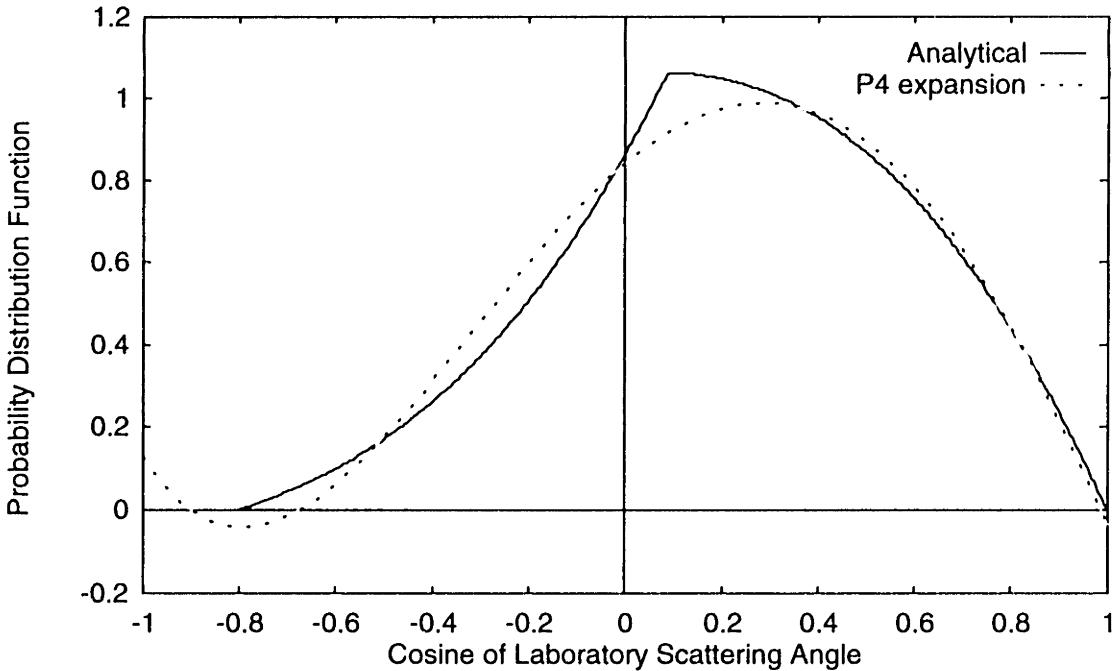


Figure 6-2: Angular distribution function for target-at-rest elastic scattering in the lab system with Deuterium. This distribution represents scattering from group 18 to 19 with a  $1/E$  flux spectrum in group 18 and a constant scattering cross section. An analytical representation and a fourth order Legendre expansion are shown.

The analytical distribution shown in this figure was calculated assuming a  $1/E$  flux spectrum in group 18 and a constant scattering cross section. Isotropic scattering in the center of mass system was also assumed. The Legendre coefficients of the  $P_4$  expansion are provided below and were obtained directly from the MENDF5 library<sup>9</sup>.

Table 6-1: Coefficients for the Legendre expansion shown in Figure 6-2.

P1	2.29096E-01
P2	-2.34511E-01
P3	-1.21645E-01
P4	2.93047E-02

The next couple of sections will discuss different ways of estimating the angular distribution and Legendre components shown in Figure 6-2.

## 6.2 Monte Carlo Approach - $f(\mu)$ Estimation

The bracketed term in equation 6-11 is the scattering rate as a function of  $\mu_0$ . This function of  $\mu_0$  can be accumulated during the Monte Carlo calculation as discussed below.

Section 4.2.2 described the Monte Carlo approach to estimating the group-to-group scattering rate. In summary, a scattering event is simulated and the exiting energy along with the incoming energy are used to determine which group-to-group bin will receive the datum ( $WTL\Sigma_s$  when a single nuclide is sampled). It will be shown that this datum also can be stored to estimate the angular distribution function.

Figure 6-3 is a graphical representation of a three dimensional array that is used to store contributions for the group-to-group scattering rates and the associated angular distribution functions. The first section of boxes (a 4x4x1 array titled scattering rate) represent the location where the group-to-group scattering rate contributions are stored. The next section of boxes (a 4x4x3 array titled bins for  $f(\mu)$ ) represent the locations where contributions are made for estimating the group-to-group angular distribution functions.

In this figure, the bins for the angular distribution function are labeled 1, 2, and N; bins 3 through N-1 are not shown. These bins correspond to equi-width cosine bins in the laboratory system. If N=100 then bin 1 would be the storage location corresponding to laboratory scattering cosines between -1 and -0.98.

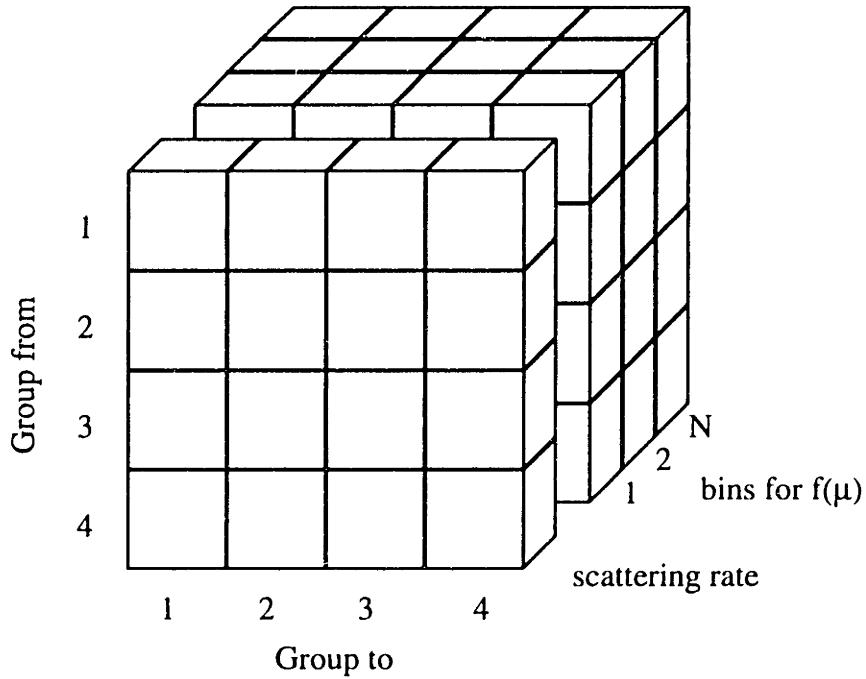


Figure 6-3: A graphical representation of a three dimensional array used for storing contributions to the group-to-group scattering rates and associated angular distribution functions. This is a four group example.

When a scattering event is simulated for cross section generation, the incoming energy is known and the outgoing energy is calculated. These two values determine which group-to-group bin receives the datum. In addition to the outgoing energy, the laboratory scattering angle, as depicted in Figure 6-1, is calculated. This angle determines which of the equi-width cosine bins will receive the datum for the estimation of the angular distribution function. The datum that is stored in this bin is the same datum stored for the scattering rate estimation. This means that the sum of the values stored in bins 1 through  $N$  of the angular distribution bins, for a specific group-to-group transfer, is equal to the value stored in the associated group-to-group scattering rate bin.

After the Monte Carlo calculation is finished, the value in each of the angular distribution bins is divided by the sum of the bins (the value in the associated scattering rate bin) to normalize the area of the distribution to one. The height of each bin is then calculated by dividing the area of the bin (the normalized value in that bin) by the width of the bin (0.02 when  $N=100$ ). With the heights of each bin, the angular distribution

function needed for equation 6-11 is known. This function is then integrated to determine the Legendre polynomial coefficients.

In order to test this method, a simple program was developed to perform the scattering simulation and calculate the angular distribution function for group 18 to 19 scattering of a neutron with Deuterium as described in Section 6.1. At the end of the simulation, the angular distribution function was integrated according to equation 6-2 to obtain the Legendre coefficients. The steps this program followed are:

1. The  $WTL\Sigma_s$  value was 1.0.
2. The energy of the incoming neutron was randomly sampled between energies 9.12 keV and 24.8 keV, group 18, assuming a  $1/E$  spectrum.
3. A center of mass scattering angle was randomly chosen between -1 and 1 (isotropic scattering in the center of mass system).
4. Using equations 3-1 and 3-2, the outgoing energy and the lab scattering angle were calculated.
5. The outgoing energy was checked to see if it was in group 19, 3.35 keV to 9.12 keV. If not, the program returned to step 2 and no contribution was made. If the outgoing energy was in group 19, a unit contribution was made to the bin determined by the laboratory scattering angle.
6. Return to step 2.

This simulation was performed three different times each with 1 million simulated scattering events. The first time, 25 equi-width cosine bins were used for the estimation of the angular distribution function; the second time 50 equi-width cosine bins were used; and the third time 100 equi-width cosine bins were used.

Figure 6-4 shows the results of this simulation when 25 equi-width cosine bins were used. The estimated angular distribution is plotted along with the analytically calculated distribution. This figure clearly indicates that this method produces good results if a sufficient number of scattering events are simulated.

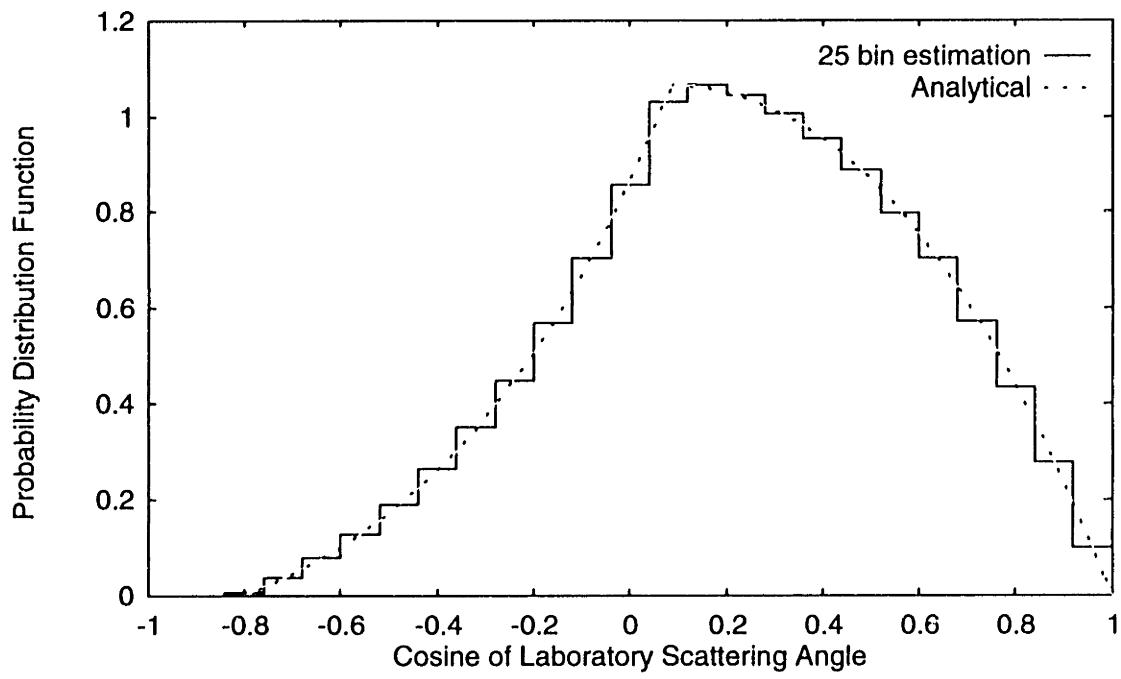


Figure 6-4: A plot of the estimated and analytical angular distribution functions. The estimated function was calculated with 25 equi-width cosine bins.

Table 6-2 shows a comparison of the Legendre components calculated from the 25 bin, 50 bin, and 100 bin estimations. These values are in relatively good agreement (~2%) with the reference values in Table 6-1. Figure 6-5 is a graphical comparison of the reconstructed angular distribution determined from the Legendre components calculated with the 25 bin estimation and from the reference values. This figure shows that, despite a 2% difference in the component values, the curves are basically identical, a result which strongly indicates that this method of estimating the  $f(\mu)$  and the associated Legendre coefficients is acceptable.

Even though the results indicate that 25 equi-width cosine bins are acceptable for the estimation of the angular distribution in this example, 50 bins were used in the MCNP modification as the default. This was done because 50 bins will more accurately represent angular distribution functions that are more narrow in width. For example, group-to-group elastic scattering off of Hydrogen can have an laboratory angular distribution function that spans a very limited range of cosines.

Table 6-2: Legendre coefficients calculated from estimated angular distribution functions having a different number of equi-width cosine bins.

	25 Bins	50 Bins	100 Bins
P1	2.334606E-01	2.334381E-01	2.334492E-01
P2	-2.306115E-01	-2.318418E-01	-2.321382E-01
P3	-1.221845E-01	-1.235910E-01	-1.239458E-01
P4	2.878347E-02	2.808829E-02	2.789943E-02

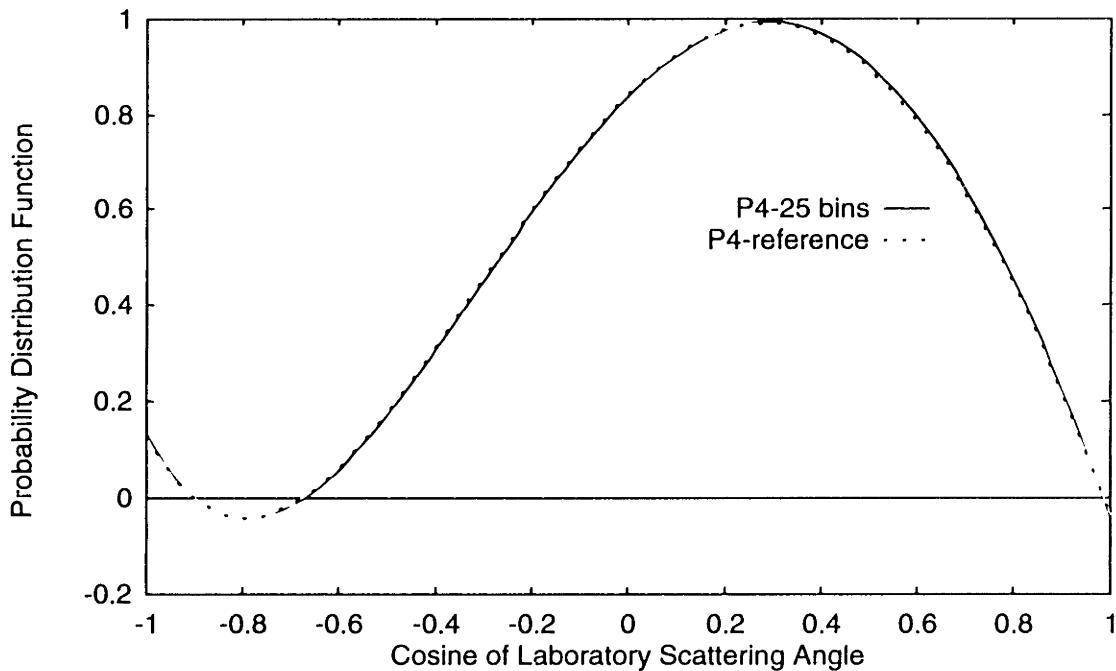


Figure 6-5: A comparison of the reference Legendre expansion from Figure 6-2 with the expansion calculated from the angular distribution function estimated using 25 equi-width cosine bins.

### 6.3 Monte Carlo Approach - Direct $P_n$ Estimation

The approach described in the previous section has been shown to work. However, there is one drawback to this approach: the number of storage locations and the amount of memory needed. If the user wished to calculate 27 group cross section using 50 bins to estimate the angular distribution function, the number of storage locations for the scattering array shown in Figure 6-3 would be  $(27 \times 27 \times 51) 37,179$  per cell. This can get to be a problem because after each history, the temporary storage array is summed into the

permanent array and every single temporary bin must be added into the permanent bin. As a result, a fairly large amount of time is consumed on simple arithmetic.

Equation 6-10 offers the solution to the problem. The Legendre coefficients will be estimated directly instead of estimating the angular distribution function. In this case, the bins 1 through N in Figure 6-3 would be the locations where the estimates for the Legendre coefficients would be stored. Therefore, if the user wished to calculate scattering cross sections with a Legendre order of 7, then N would be 7 in Figure 6-3 and the total number of group-to-group storage locations for generating 27 group cross sections would be reduced to 5,832 (27x27x8).

Equation 6-12, a slightly modified version of equation 6-6, shows the approach to estimating the Legendre components at each simulated scattering event. The left hand side of equation 6-12 is the  $P_n$  coefficient of the datum that is stored when a single nuclide is sampled for the scattering event. This  $P_n$  coefficient value is stored in the appropriate group-to-group bin in the second block of storage bins in Figure 6-3.

$$(WTL\Sigma_s)^n = \int_{-1}^1 WTL\Sigma_s f(\mu_0) P_n(\mu_0) d\mu_0 \quad (6-12)$$

In essence, the Legendre components of each datum being stored are calculated. In this way the Legendre components of the scattering rate are being estimated in the same manner as the scattering rate itself.

The integration in equation 6-12 is performed at each simulated scattering event. In this circumstance, the  $f(\mu)$  is actually a delta function instead of a full distribution since there is only one outgoing laboratory angle,  $\cos^{-1}\mu_L$  (see Figure 6-1). The following equations define the delta function.

$$\delta(x) = 0 ; x \neq 0$$

$$\int f(x)\delta(x)dx = f(0)$$

$$\int f(x)\delta(x-a)dx = f(a)$$

Using these relationships with equation 6-12, the exact manner of calculating the Legendre coefficients of the datum is shown. Equation 6-13 is the definition of the value that is stored in the group-to-group Legendre component bins.

$$\begin{aligned} (\text{WTL}\Sigma_s)^n &= \int_{-1}^1 \text{WTL}\Sigma_s f(\mu_0) P_n(\mu_0) d\mu_0 \\ f(\mu_0) &= \delta(\mu_0 - \mu_L) \\ (\text{WTL}\Sigma_s)^n &= \int_{-1}^1 \text{WTL}\Sigma_s \delta(\mu_0 - \mu_L) P_n(\mu_0) d\mu_0 \\ (\text{WTL}\Sigma_s)^n &= \text{WTL}\Sigma_s P_n(\mu_L) \end{aligned} \quad (6-13)$$

Another simple program was written to test this method. This program was identical to the program written for the previous section, with the following exception. Instead of making a unit contribution ( $\text{WTL}\Sigma_s = 1.0$ ) in a bin corresponding to the  $\mu_L$  calculated, the following values were calculated and stored:  $P_1(\mu_L)$ ,  $P_2(\mu_L)$ ,  $P_3(\mu_L)$ , and  $P_4(\mu_L)$ . At the end of the simulation, the stored values were normalized to the number of simulated particles.

Table 6-3 shows the Legendre coefficients that were calculated with this method. Comparing these results to the values in Table 6-2 clearly indicates that this method is accurate.

Table 6-3: Legendre coefficients estimated using direct integration of the angular distribution function at each scattering simulation event.

P1	2.334486E-01
P2	-2.322440E-01
P3	-1.240736E-01
P4	2.782672E-02

## 6.4 Explicit Approach - $f(\mu)$ Estimation

Chapter 5 described the explicit approach to estimating the group-to-group scattering rate. In this approach, the scattering law is manipulated to determine the contribution to all possible groups from scattering at the incoming energy. For example, Sections 5.1 and

5.2 showed that for elastic scattering the contribution to all groups is determined by integration of the center of mass angular distribution function.

Since the group-to-group scattering rate is being calculated for all possible outgoing groups, it is essential that a similar contribution be made to each associated group-to-group angular distribution function. The only question is how to do this. Section 6.2 explained that, using the Monte Carlo approach, a laboratory scattering angle is known since a scattering event is simulated. However, in the explicit approach described in Chapter 5 a scattering event is not simulated, therefore a single laboratory scattering angle is not known. The solution to this problem is discussed below and an example is used to illustrate the methodology.

#### 6.4.1 Center of Mass Angular Distribution

Elastic scattering, target-at-rest and target-in-motion, and scattering law 3 use a 32 equi-probable bin center of mass angular distribution for the kinematics calculations. As described in Sections 5.1, 5.2, and 5.4, this center of mass angular distribution is integrated to determine the fraction of the outgoing energies that belong to specific groups. In order to estimate the associated laboratory angular distribution, a minimum of 32 randomly chosen laboratory angles are calculated and used to determine which angular distribution bins will receive data. The following example illustrates the technique.

For this example, assume the center of mass angular distribution function is comprised of 4 equi-probable cosine bins instead of 32. The bin boundaries are listed below and represent isotropic scattering in the center of mass system.

The incoming neutron has an energy of  $E_1$ , which is in group 1. Using the method described in Section 5.1, it is determined that the neutron can scatter to groups 1, 2, and 3.

C-M Bin	Lower Boundary	Upper Boundary	Height of Bin
1	-1.0	-0.5	0.5
2	-0.5	0.0	0.5
3	0.0	0.5	0.5
4	0.5	1.0	0.5

Figure 6-6 is a graphical representation of the example center of mass isotropic angular distribution. The angles corresponding to group boundaries and the fractional areas corresponding to specific group-to-group transfers are shown. In this figure the whole and fractional equi-probable bins that comprise the group-to-group transfers are labeled. Regions D and E are the equi-probable bins that represent outgoing energies within group 1. Region C is a fraction of an equi-probable bin and represents outgoing energies within group 2 and regions A and B represent outgoing energies in group 3. Table 6-4 also presents the information shown in this figure.

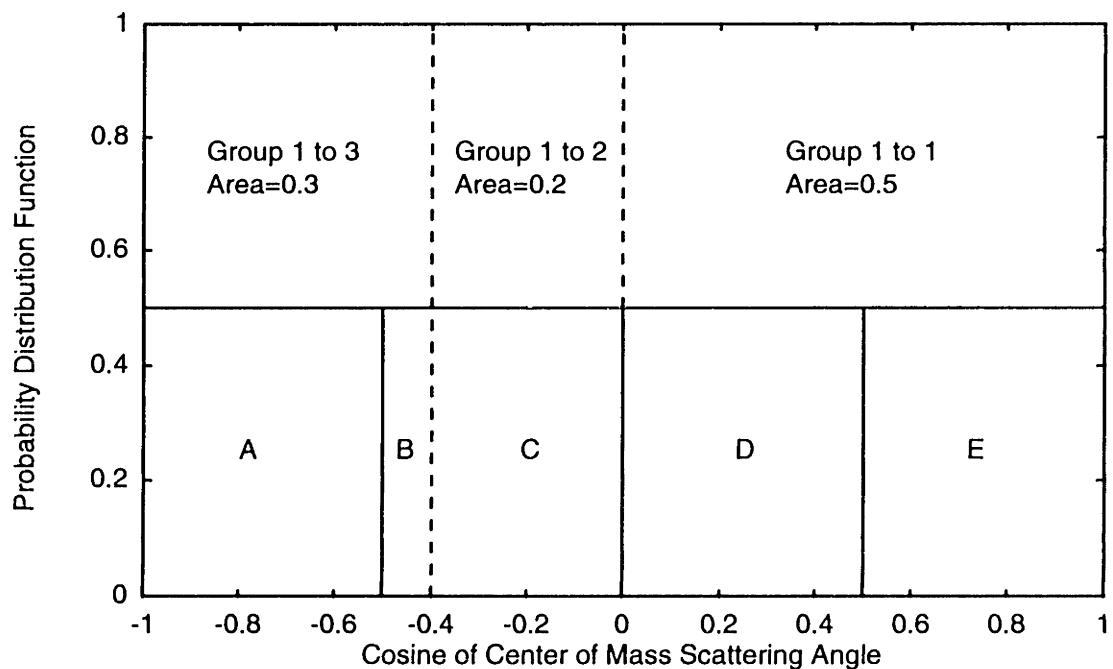


Figure 6-6: A graphical representation of the example center of mass angular distribution. The integrated areas corresponding to specific group-to-group transfers are shown. The fractional equi-probable cosine bins that comprise the integrated areas are labeled.

Once the center of mass angular distribution is integrated (according to the method of Section 5.1), the next task is to store information in the corresponding equi-width cosine bins for the estimation of the angular distribution function. In order to get as much information as possible, each region labeled in Figure 6-6 will be randomly sampled and

a laboratory scattering angle will be determined. This angle will dictate which angular distribution bin, for that specific group-to-group transfer, will receive data.

Table 6-4: A tabular representation of the data in Figure 6-6. The fractional areas of the regions and the outgoing group they represent are shown.

Region	Group to	C-M Angles	Area
A	3	-1.0 to -0.5	0.25
B	3	-0.5 to -0.4	0.05
C	2	-0.4 to 0.0	0.20
D	1	0.0 to 0.5	0.25
E	1	0.5 to 1.0	0.25

The following steps illustrate the process.

1. A random number from 0 to 1 is chosen. In order to conserve computer time, a single random number is used to sample all regions.
2. A center of mass angle is randomly chosen for region A from the angles that region A spans. The random number from step 1 is used for this calculation.
3. Either equation 5-5 (elastic scattering target-in-motion), 3-2 (elastic scattering target-at-rest) or 3-5 (law 3) is used to calculate the laboratory scattering angle.
4. The value  $WT\sum_s$ , times the area of region A (0.25) is stored in the group 1 to 3 angular distribution bin determined by the laboratory angle from step 3.
5. Steps 2 through 4 are repeated for regions B, C, D, and E.

The result of this process is that two angular bins for the group 1 to 3 and the group 1 to 1 transfer will receive information while one angular bin will receive information for the group 1 to 2 transfer.

The basic technique described above would also be applicable to the  $S(\alpha,\beta)$  data where the possible outgoing scattering angles are represented as discrete angles in the laboratory system.

#### **6.4.2 Laboratory Angular Distributions**

Scattering laws 1 and 9 use a laboratory angular distribution consisting of 32 equi-probable cosine bins to determine the scattering angle. The choice of scattering angle is independent of the outgoing energy. Therefore, the same set of laboratory scattering angles are valid for all energy transfers (group-to-group) that may occur with these laws. Sections 5.3 and 5.5 described the methods of determining the energy groups that can be reached by particles with a particular incoming energy.

Since the angular distribution of outgoing particles is already in the laboratory system, it is a simple matter to use this distribution and make a contribution to all equi-width cosine bins of the distribution function being accumulated for each possible group-to-group transfer (see Figure 6-3).

### **6.5 Explicit Approach - Direct $P_n$ Estimation**

Section 6.3 described the method of directly estimating the  $P_n$  components of the scattering cross sections using a delta function while using the Monte Carlo approach to estimate the group-to-group scattering cross section. This same delta function integration technique can be applied when the explicit approach to estimating the scattering cross section, as described in Chapter 5, is used.

The previous section discussed how the angular distribution function is accumulated when the explicit approach to estimating the scattering cross section is performed. As shown, the technique differed when the scattering law being considered used a center of mass versus a laboratory angular distribution function. The method of directly estimating the  $P_n$  components also depends on the original form of the scattering distribution.

#### **6.5.1 Center of Mass Angular Distribution**

Section 6.4.1 described the method of accumulating the lab angular distribution when the possible outgoing scattering angles are provided in the center of mass system. The delta function technique of estimating the  $P_n$  components can also be used in this case. The implementation of this technique is almost identical to the steps shown in

Section 6.4.1. These steps are reiterated here with the appropriate modifications for the delta function integration. Figure 6-6 and Table 6-4 identify the regions mentioned below.

1. A random number from 0 to 1 is chosen. In order to conserve computer time, a single random number is used to sample all regions.
2. A center of mass angle is randomly chosen for region A from the angles that region A spans. The random number from step 1 is used for this calculation.
3. Either equation 5-5 (elastic scattering target-in-motion), 3-2 (elastic scattering target-at-rest) or 3-5 (law 3) is used to calculate the laboratory scattering angle.
4. The value  $WT\Lambda\Sigma_s$  times the area of region A (0.25) is multiplied by the  $P_n$  Legendre polynomial evaluated with the lab angle from step 3. This value is stored in the appropriate bin, corresponding to the Legendre order, for group 1 to 3 scattering. This step is repeated N (the desired Legendre order) times.
5. Steps 2 through 4 are repeated for regions B, C, D, and E.

The basic technique described above would also be applicable to the  $S(\alpha, \beta)$  data where the possible outgoing scattering angles are represented as discrete angles in the laboratory system.

### 6.5.2 Laboratory Angular Distribution

As discussed in Section 6.4.2, certain scattering laws represent the possible outgoing scattering angles in the lab system. Since these scattering angles do not depend on the outgoing energies, the same angular distribution is used for all possible outgoing energies. Therefore, it is straightforward to integrate the angular distribution function and determine the Legendre components using equation 6-12.

## 6.6 Direct $P_n$ Estimation - An Alternate Method

This chapter has focused on the calculation of the Legendre components of the scattering cross section. Equation 6-10 is the basic equation governing the calculation. The crux of equation 6-10 is the knowledge of the laboratory angular distribution

function. In some cases the scattering law represents the outgoing scattering angles in the lab system thereby eliminating the difficulty in determining that function.

However, a larger fraction of the scattering events, including elastic scattering, represent the possible outgoing scattering angles in the center of mass system. These angles are converted to the lab system through the kinematics of the reaction and the equations that have been presented in Chapters 4 and 5. Because of this, approximate techniques of estimating the laboratory angular distribution or directly calculating the  $P_n$  components have been developed. These were described in Sections 6.4.1 and 6.5.1. If the actual laboratory angular distribution function was known, the approximate methods could be avoided and equation 6-12 could be used directly to calculate the Legendre components of the scattering cross section.

The following relationships relate the laboratory angular distribution and the center of mass angular distribution.

$$f(\mu_L)d\mu_L = f(\mu_c)d\mu_c$$

$$f(\mu_L) = f(\mu_c) \frac{d\mu_c}{d\mu_L} \quad (6-14)$$

Since all of the center of mass angular distributions on the MCNP data libraries consist of equi-probable bins, the  $f(\mu_c)$  in equation 6-14 is a constant over specific intervals dictated by the bin boundaries. Because of this, it is possible to determine the  $f(\mu_L)$  and directly integrate it according to equation 6-12. The procedure is shown below for elastic scattering with the target-at-rest.

Equation 3-2 showed the formula for calculating the laboratory scattering angle when the center of mass angle is known for elastic scattering target-at-rest. This equation can be reworked to provide the center of mass angle as a function of the lab angle. The resulting relationship and its first derivative are:

$$\mu_c = \frac{\mu_L^2 - 1}{A} + \frac{\mu_L}{A} \sqrt{\mu_L^2 + A^2 - 1}$$

$$\frac{d\mu_c}{d\mu_L} = \frac{2\mu_L}{A} + \frac{1}{A} \sqrt{\mu_L^2 + A^2 - 1} + \frac{\mu_L^2}{A \sqrt{\mu_L^2 + A^2 - 1}}$$

Insertion of the latter into equation 6-14 yields the equation defining the laboratory angular distribution function:

$$f(\mu_L) = f(\mu_c) \left( \frac{2\mu_L}{A} + \frac{1}{A} \sqrt{\mu_L^2 + A^2 - 1} + \frac{\mu_L^2}{A \sqrt{\mu_L^2 + A^2 - 1}} \right) \quad (6-15)$$

Since  $f(\mu_c)$  is a function consisting of equi-probable cosine bins, its value may change between cosine bins but is constant within a bin. Figure 6-7 is a graph of the center of mass and laboratory angular distribution for elastic scattering, target-at-rest, with Deuterium. Scattering in the center of mass system is isotropic,  $f(\mu_c)=0.5$ , and the lab distribution was calculated with equation 6-15.

With Figure 6-6 taken as an example, it is possible to use equation 6-15 to find the Legendre coefficients for the group-to-group transfers shown in the figure. The following integration results. (The  $f(\mu_c)$  in equation 6-15 has been replaced with 0.5 since the scattering in this example is isotropic in the center of mass system.)

$$f_n = \int_a^b d\mu_L 0.5 \left( \frac{2\mu_L}{A} + \frac{1}{A} \sqrt{\mu_L^2 + A^2 - 1} + \frac{\mu_L^2}{A \sqrt{\mu_L^2 + A^2 - 1}} \right) P_n(\mu_L) \quad (6-16)$$

where equation 3-2 is used to determine

$a =$  lower bound  $\mu_L$  corresponding to the lower bound  $\mu_c$

$b =$  upper bound  $\mu_L$  corresponding to the upper bound  $\mu_c$

The integration shown in equation 6-16 can be performed in closed form. However, as the Legendre order increases, the complexity of the resulting equation increases and the computer time needed to evaluate the expression also increases. In addition, the equation for  $f(\mu_L)$  for the case of elastic scattering with the target-in-motion would be considerably more complex to evaluate. Therefore, it was decided that this method of calculating the Legendre coefficients would not be implemented in MCNP.

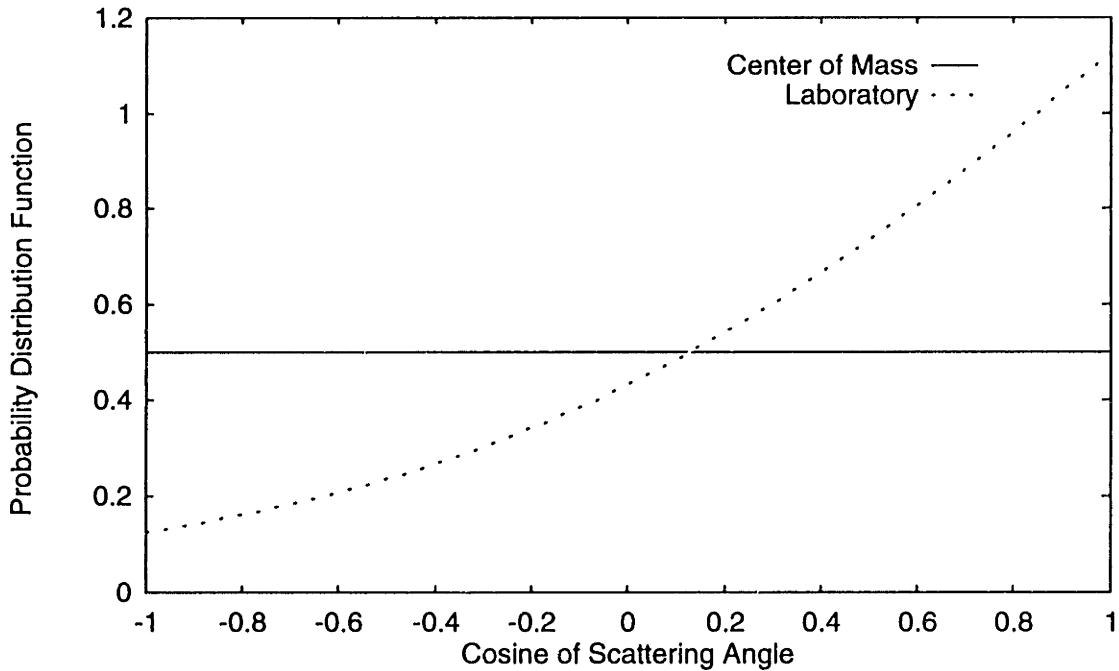


Figure 6-7: A plot of the isotropic center of mass angular distribution function and the associated laboratory angular distribution function for elastic scattering with Deuterium when the target is at rest.

## 6.7 MCNP Multigroup Libraries

Instead of using continuous energy data libraries in MCNP, a multigroup library can be used. One advantage to using a multigroup library is that the problems will run faster. In addition, the multigroup option is currently the only way to perform an adjoint Monte Carlo calculation with MCNP.

The format which MCNP uses for the multigroup library is not one of the standard formats. It does not use a Legendre expansion to represent the angular distribution of the scattered particles. Rather, the distribution is represented with 32 equi-probable cosine bins (the same format used in the continuous energy data). Because of this, the only way to transform a standard multigroup library with a Legendre expansion into the format required by MCNP is to use CRSRD<sup>13</sup>, a Los Alamos cross section processing program.

The cross section generation routines in this thesis do offer the capability to output the multigroup library in MCNP format. However, this option can only be used if the  $f(\mu)$

angular distribution is accumulated in equi-width cosine bins instead of calculating the Legendre components directly. If the equi-width cosine bins exist, they can be integrated to determine the bin boundaries for the required 32 equi-probable cosine bins. If the Legendre components are generated, the auxiliary processing code CRSRD would have to be used to get the cross sections into the MCNP format.

Chapter 8 will identify a potential problem with MCNP formatted multigroup libraries that are created automatically. Therefore, it is recommended that a standard multigroup library (containing Legendre components) be generated and that CRSRD be used to transform this library into the MCNP format.

## Chapter 7

### Methods Summary

This chapter will provide a general summary of the multigroup cross section generation techniques that have been implemented in MCNP. The first item that will be discussed is the track length tally estimator that is used in MCNP to calculate various quantities including reaction rates and flux. Next, the cross sections being calculated will be listed and finally a discussion of the techniques involved in estimating the group-to-group scattering cross section and the associated Legendre components will be provided.

Generating multigroup cross sections is a two step process. In step one, the flux and the various reaction rates are tallied during particle transport. The second step, which is performed after the calculation has been completed, is to divide the reaction rates by the flux, thereby calculating the macroscopic multigroup cross section. The discussion that follows focuses on the generation of the reaction rates with emphasis on the group-to-group scattering rate.

#### 7.1 Track Length Estimator

This discussion of the track length estimator is very similar to the presentation found in the MCNP manual<sup>1</sup>.

Particle flux is defined as:

$$\Phi(\vec{r}, E, t) = vN(\vec{r}, E, t)$$

v = particle velocity

N = particle density

The neutrons comprising a Monte Carlo flux tally appear and survive for a certain time. From this point of view, the time integrated volume averaged group flux,  $\phi_g$ , is:

$$\phi_g = \int \int \int_{V \cap E_g}^{E_{g,i}} \Phi(\vec{r}, E, t) dEdt \frac{dV}{V} \equiv \int \int \int_{V \cap E_g}^{E_{g,i}} vN(\vec{r}, E, t) dEdt \frac{dV}{V}$$

Since distance,  $ds$ , is equivalent to velocity multiplied by time,  $vdt$ , the following can be written:

$$\phi_g = \int \int \int_{V \cap E_g}^{E_{g,i}} \Phi(\vec{r}, E, t) dEdt \frac{dV}{V} = \int \int \int_{V \cap E_g}^{E_{g,i}} N(\vec{r}, E, t) dEds \frac{dV}{V}$$

In Monte Carlo calculations, particle density is equivalent to a summation of particle weight per unit volume and distance is equivalent to track length, TL. Track length is defined as the distance from a surface crossing or particle interaction to the next surface crossing or particle interaction. Therefore the definition above can be represented as:

$$\phi_g = \frac{\int_{E_g}^{E_{g,i}} dE \int_V dV \sum_{i=1}^N WTL_v^i(E)}{V \sum_{i=1}^N W_0^i} \quad (7-1)$$

$WTL_v^i(E)$  = weight of ith particle times the track length of ith particle at energy E in volume region V

V = volume or region of interest

$W_0^i$  = original weight of particle i

N = number of particle histories tracked

The  $W_0$  is the original particle weight and normalizes the answers to a single starting particle.

A similar equation can be written to define a group reaction rate calculated with a track length estimator.

$$\Sigma_g \phi_g = \frac{\int_{E_g}^{E_{g,i}} dE \int_V dV \sum_{i=1}^N WTL_v^i(E) \Sigma(r, E)}{V \sum_{i=1}^N W_0^i} \quad (7-2)$$

$\Sigma(r, E)$  = cross section as a function of energy and position

The integration over energy and volume in equations 7-1 and 7-2 is performed as a summation. A separate tally bin is created for each energy group and volume region combination. The WTL product is then placed in the appropriate bin for each track length segment that occurs. This method effectively performs the integration over energy and volume. Contributions to the tally bins occur at the end of a particle track length (interaction site or surface crossing) and prior to any change in energy or direction that might occur due to an interaction.

Equation 7-2 is the basic equation governing the estimation of reaction rates within MCNP. During particle transport, the weight and the track length of the particle are always available; however, as will be discussed, the energy dependent cross section may not be readily accessible.

## 7.2 Cross Sections Needed

In order for a multigroup library to be useful, the following cross sections must be calculated.

- $\Sigma_{lg}$  = total group cross section
- $\Sigma_{ag}$  = absorption group cross section
- $\Sigma_{fg}$  = fission group cross section
- $v\Sigma_{fg}$  = nu-sigma fission group cross section
- $X_g$  = group chi
- $\Sigma_{sgg'}$  = group-to-group scattering cross section
- $\Sigma_{sgg'}^n$  = Legendre components of the group-to-group scattering cross section

As mentioned above, the reaction rate for these cross sections is calculated using equation 7-2. In MCNP, all of the cross sections needed to calculate these reaction rates are directly available except the group-to-group scattering cross section. Therefore, it is a

simple matter to accumulate, according to equation 7-2, all of the reaction rates in the desired energy groups with the exception of the group-to-group scattering rate.

## 7.3 Group-to-Group Scattering Rate

In MCNP, the scattering cross section is tabulated as a function of energy and represents the probability of a scattering event occurring for a particle with a specific incoming energy. The scattering cross section itself does not provide any information about energy or angular distributions of particles emerging from a scattering event. This information is calculated through the various scattering laws coded in MCNP. A scattering law is a set of data or functions that determine the outgoing energy and angle (relative to the incoming direction) of a particle after an interaction. Probability distributions of the cosine of the scattering angle in either the center of mass or laboratory system are associated with each scattering law and tabulated at various incoming energies. These distributions represent the probability that an incoming particle will scatter through a specific angle relative to the incoming particle direction. Based on the specific scattering law being used, the outgoing energy of the particle may or may not depend on the value of this scattering angle. The exact relationship between the incoming energy, outgoing energy, and scattering angle is determined by the kinematics defined by the scattering laws.

Since the group-to-group scattering cross section is not readily available, the question is what value should be used in equation 7-2 to determine the scattering rate. There are two possible solutions to this problem.

### 7.3.1 Monte Carlo Approach

One way of estimating the group-to-group scattering rate is to simulate a scattering event and observe the outgoing energy. The scattering simulation occurs outside of the normal particle transport and does not affect the particle's flight path. Using the incoming particle energy (the energy of the particle prior to the simulation) and the outgoing energy from the simulation, a tally contribution can be made to the group-to-group scattering bin

determined by these energies. The value of this contribution, according to equation 7-2, would be:

$$WTL\Sigma(E) = WTL\Sigma_s(E^{\text{incoming}})$$

where the spatial dependence notation has been removed from this expression.

The result is that, at every interaction site, only one group-to-group scattering bin receives a contribution. This does not represent the full range of scattering possibilities for a particle with that particular incoming energy. However, if a sufficiently large number of particles are tracked, all of the possible group-to-group scattering bins will receive information and the scattering rates will be calculated correctly.

### 7.3.2 Explicit Approach

The previous sub-section illustrated one method of estimating the group-to-group scattering rate. The disadvantage to that method is that a contribution is made to only one group-to-group scattering bin for each track length. Clearly, it would be better for the statistical convergence of the scattering rates if a contribution could be made to all possible scattering bins when the weight times track length is stored. The following expression represents this contribution.

$$WTL\Sigma(E) = WTL\Sigma_s(E^{\text{incoming}})f_k$$

The  $f_k$  is the fraction of outgoing energies that would enter group  $k$  if the scattering event were simulated an infinite number of times. The  $f_k$  is calculated mathematically by manipulating the particular scattering law in effect rather than simulating the scattering events as in the Monte Carlo approach. Once these  $f_k$  are calculated for all possible outgoing groups, the tally contributions are made to the appropriate group-to-group scattering rate tally bins. This method will result in improved statistical convergence of the scattering rates since all possible group-to-group bins, instead of a single bin, receive a contribution with each track length.

## 7.4 Legendre Components

In order for multigroup cross sections to be useful in various transport codes, the angular distribution of the scattering angles (relative to the particle flight direction) must be known. This distribution is usually represented with an expansion in Legendre polynomials. The order of this expansion is typically 3 or 5.

The following equation defines the Legendre component of the group-to-group scattering cross section.

$$\Sigma_{sgg'}^n \equiv \frac{\int_{E_{t+1}}^{E_{t+1}} dE \int_{E_t}^{E_{t+1}} dE' \int_{\Delta z} dz \Sigma_s^n(z, E' \rightarrow E) \Psi^n(z, E')}{\int_{E_{t+1}}^{E_{t+1}} dE' \int_{\Delta z} dz \Psi^n(z, E')} \quad (7-3)$$

where

$\Sigma_s^n(z, E' \rightarrow E)$  = nth Legendre component of the double differential scattering cross section

$\Psi^n(z, E')$  = nth Legendre component of the directional flux density

and

$$\begin{aligned} \Psi^n(z, E') &= \int_{-1}^1 d\mu \Psi(z, \mu', E') P_n(\mu') \\ \Sigma_s^n(z, E' \rightarrow E) &= \int_{-1}^1 d\mu_0 \Sigma_s(z, E' \rightarrow E, \mu_0) P_n(\mu_0) \end{aligned} \quad (7-4)$$

Because of the extreme difficulty in calculating the Legendre components of order  $n > 0$  (the Legendre coefficients approach zero as the flux becomes isotropic) the following approximation was made.

$$\Psi^n(z, E') \approx C_n \Psi^0(z, E') = C_n \int_{-1}^1 d\mu \Psi(z, \mu', E')$$

This approximation means that a  $P_n$  component of the scattering cross section is being weighted by the energy dependent scalar flux in the same fashion as other cross sections. Using equation 7-4 and the flux approximation, equation 7-3 can be written as:

$$\Sigma_{sgg'}^n \equiv \frac{\int_{E_i}^{E_{i+1}} dE \int_{E_{i'}}^{E_{i'+1}} dE' \int_{\Delta z} dz C_n \Psi^0(z, E') \int_{-1}^1 d\mu_0 \Sigma_s(z, E' \rightarrow E, \mu_0) P_n(\mu_0)}{\int_{E_{i'}}^{E_{i'+1}} dE' \int_{\Delta z} dz C_n \Psi^0(z, E')} \quad (7-5)$$

Equation 7-5 also can be written as:

$$\Sigma_{sgg'}^n \equiv \frac{\int_{-1}^1 d\mu_0 \left[ \int_{E_i}^{E_{i+1}} dE \int_{E_{i'}}^{E_{i'+1}} dE' \int_{\Delta z} dz C_n \Psi^0(z, E') \Sigma_s(z, E' \rightarrow E, \mu_0) \right] P_n(\mu_0)}{\int_{E_{i'}}^{E_{i'+1}} dE' \int_{\Delta z} dz C_n \Psi^0(z, E')} \quad (7-6)$$

The bracketed item in the numerator represents the determination of the scattering rate as a function of  $\mu_0$ . This can be performed in a Monte Carlo fashion during the calculation. Alternatively, equation 7-5 illustrates that the  $P_n$  components of the scattering rate can be calculated directly. Both methods of calculating the  $P_n$  components of the scattering cross section are discussed below.

#### 7.4.1 Monte Carlo Approach - $f(\mu)$ Estimation

The Monte Carlo approach is to accumulate the group-to-group angular distribution function during the particle transport. Associated with each group-to-group scattering rate tally bin is 50 or more equi-width cosine bins which are used to store the angular distribution. This distribution is integrated at the end of the calculation to determine the  $P_n$  components of the scattering cross section.

Section 7.3.1 described the Monte Carlo approach for estimating the scattering cross section. In this approach, a scattering event is simulated to determine which group-to-group scattering bin will receive a contribution. When the outgoing energy is calculated, it is a simple matter to determine the laboratory scattering angle. This scattering angle

dictates which of the equi-width cosine bins will receive the tally contribution. The contribution is the same as the one made to the group-to-group scattering rate bin.

#### 7.4.2 Direct $P_n$ Calculation

Instead of accumulating the laboratory angular distribution function it is possible to use equation 7-5 and determine the  $P_n$  components of each individual scattering contribution. This effectively determines the  $P_n$  component of the scattering rate. Therefore, associated with each scattering rate tally bin is a set of bins, each corresponding to a  $P_n$  component of the scattering rate. At the end of the calculation these values are divided by the flux to yield the desired Legendre components of the cross sections.

### 7.5 Summary of Group-to-Group Calculational Methods

There have been two methods of calculating the group-to-group scattering cross sections and two methods of calculating the Legendre components described. These methods are summarized below in Table 7-1.

The explicit approach to calculating the group-to-group scattering rate is used automatically for elastic scattering, and laws 1, 3, 9, and 11. The Monte Carlo approach is used for the rest of the scattering laws.

The default method of calculating the Legendre components is the  $f(\mu)$  estimation. However, the user can change this with an input parameter. Each method of calculating the Legendre expansion works with either method of estimating the scattering rate.

Table 7-1: A summary of the methods used for calculating the group-to-group scattering rates and the associated Legendre components.

<b>Group-to-Group Scattering Methods</b>	
Monte Carlo Approach	A single scattering event is simulated and based on the incoming and outgoing energies a contribution is made to a single group-to-group scattering bin.
Explicit Approach	A scattering law is sampled and is manipulated to determine the fraction of particles that could scatter into every possible group. A contribution is made to each of these group-to-group scattering bins.
<b>Legendre Component Methods</b>	
f( $\mu$ ) Estimation	The angular distribution is accumulated during the calculation by making a contribution to an equi-width cosine bin associated with a group-to-group scattering bin. This distribution is weighted and integrated at the end of the calculation to determine the Legendre components.
Direct $P_n$ Estimation	The $P_n$ components of the scattering rate are estimated during the calculation by integrating the angular distribution for each scattering cross section.

## **Chapter 8**

### **Validation**

This chapter will describe the validation of the cross section modifications that have been made to MCNP. Examples will be provided to demonstrate that reliable multigroup cross sections can be generated with MCNP.

During the discussion, the different methods of calculating the group-to-group scattering cross section and the associated Legendre components will be referenced. Table 7.1, from the last chapter, provides the reader with a brief summary of these methods.

The first and foremost validation step was to assure that MCNP was not inadvertently affected by the code changes. Therefore, the 25 test problems that are distributed with MCNP were used to verify that the code still performed as expected. The test problems do not indicate whether or not the code modifications are correct, but rather, that the modifications did not damage other portions of the source code.

The next step in the validation process was to set up three test problems and generate multigroup cross sections. The generated cross sections were used in different codes and the results compared to the Monte Carlo reference. The test problems are described below.

The multigroup cross sections were generated in one of two group structures. The first was a standard twenty-seven group structure that is used in SCALE 4.3.<sup>14</sup> The second group structure consisted of three groups. Table 8-1 presents the 27 group structure and Table 8-2 shows the 3 group format.

In the data presented below, all standard deviations are  $1\sigma$ .

Table 8-1: Standard twenty-seven group library format from SCALE 4.3.

<b>Group</b>	<b>Lower Energy (MeV)</b>	<b>Upper Energy (MeV)</b>
1	6.434	20.0
2	3.0	6.434
3	1.85	3.0
4	1.4	1.85
5	9.0E-01	1.4
6	4.0E-01	9.0E-01
7	1.0E-01	4.0E-01
8	1.7E-02	1.0E-01
9	3.0E-03	1.7E-02
10	5.5E-04	3.0E-03
11	1.0E-04	5.5E-04
12	3.0E-05	1.0E-04
13	1.0E-05	3.0E-05
14	3.05E-06	1.0E-05
15	1.77E-06	3.05E-06
16	1.3E-06	1.77E-06
17	1.13E-06	1.3E-06
18	1.0E-06	1.13E-06
19	8.0E-07	1.0E-06
20	4.0E-07	8.0E-07
21	3.25E-07	4.0E-07
22	2.25E-07	3.25E-07
23	1.0E-07	2.25E-07
24	5.0E-08	1.0E-07
25	3.0E-08	5.0E-08
26	1.0E-08	3.0E-08
27	0.0	1.0E-08

Table 8-2: A three group library format used for cross section generation.

<b>Group</b>	<b>Lower Energy (MeV)</b>	<b>Upper Energy (MeV)</b>
1	3.0E-03	20.0
2	4.0E-07	3.0E-03
3	0.0	4.0E-07

## 8.1 Infinite Slab

The first test problem created was an infinite slab of  $^{235}\text{U}$  and Hydrogen. The slab was 30 cm in width and had an atom density of 3.25E-02 atoms/barn-cm. The atom fraction of the  $^{235}\text{U}$  and Hydrogen were 0.02 and 0.98 respectively. This test problem was used for two purposes. The first was to verify that MCNP was creating correct multigroup cross sections and the second was to compare the various methods of cross section generation (see Table 7.1).

The reference k-effective of this system, calculated with MCNP, was  $1.0291 \pm 0.0022$ . The number of active criticality cycles was 100 with 3000 neutrons per cycle.

Table 8-3 compares the run times from various MCNP calculations of the infinite slab. Different cross section generation options were used and the number of groups and the  $P_n$  order were varied to see the effect on computer run time. The number of active criticality cycles was identical in all cases.

Table 8-3: A comparison of MCNP calculations of an infinite slab. Various cross section generation options were used to see the effect on computer run time.

MCNP Run	$\Sigma_s$ Method	$P_n$ Method	Number Groups	$P_n$ Order	Time (mins)
slb400	none	none	-	-	8.0
slb404	Monte Carlo	direct $P_n$	27	7	20.8
slb403	exact	$f(\mu)$ est.	27	7	83.5
slb406	exact all isotopes	direct $P_n$	27	7	47.68
slb402	exact	direct $P_n$	27	7	42.0
slb414	exact	direct $P_n$	27	3	34.8
slb410	exact	direct $P_n$	27	0	28.1
slb412	exact	direct $P_n$	3	7	24.2

There are a few conclusions that can be drawn from the data in Table 8-3.

- The Monte Carlo approach to cross section generation is the fastest. This is reasonable since the Monte Carlo approach is not computationally intensive.
- The  $f(\mu)$  estimation technique for the Legendre expansion results in a noticeably slower calculation than if the direct  $P_n$  method is used. This is probably due to the increased size of the data storage array and the additional arithmetic associated with combining the temporary array with the permanent array after a particle history is complete.
- Sampling all nuclides for scattering, as opposed to a single nuclide, did not increase the run time dramatically. However, it should be noted that there are only two nuclides in this problem and one would expect the run time to increase even more if the number of nuclides were larger.
- Decreasing the  $P_n$  order for the direct  $P_n$  method results in a noticeable decrease in run time. This is due to the elimination of the arithmetic that was performed to evaluate the higher Legendre components.
- Decreasing the number of groups also decreases run time. The reduction in time occurs because of two separate reasons. The first reason is the reduced size of the storage array. The second reason is that, with a fewer number of groups, there is less math involved in determining the exact group-to-group contributions. This is because there are fewer group boundaries that must be considered in the evaluation.

Table 8-4 shows the eigenvalues that were calculated from the MCNP reference calculation and from three of the cross section generating calculations. Since the cross section generation subroutines use random numbers, it is expected that the eigenvalue would be different when cross sections are calculated. However, the eigenvalue should be within three standard deviations of the reference. This is true of the data shown in Table 8-4.

Table 8-4: A comparison of the eigenvalues estimated with different MCNP calculations of the infinite slab.

MCNP Run	$\Sigma_s$ Method	$P_n$ Method	k-eff	Standard Deviation
slb400	none	none	1.0291	0.0022
slb402	exact	direct $P_n$	1.0281	0.0015
slb404	Monte Carlo	direct $P_n$	1.0273	0.0018
slb406	exact all isotopes	direct $P_n$	1.0304	0.0018

Table 8-5 compares the relative error and figure of merit for two of the group-to-group scattering cross sections calculated from different MCNP runs. The relative error is defined as the standard deviation divided by the mean. The figure of merit, which is an MCNP term<sup>1</sup>, is defined with the following equation:

$$FOM = \frac{1}{R^2 T}$$

R = relative error

T = computer time

The relative error can then be written as:

$$R = \frac{1}{\sqrt{FOM \times T}}$$

This clearly indicates that, for a fixed computer time, the calculation with the higher figure of merit will have a lower relative error.

The choice of which group-to-group bins to use in Table 8-5 was arbitrary. The important conclusion to be drawn from Table 8-5 is that the exact method of calculating the group-to-group scattering rate combined with the direct  $P_n$  method is the optimum for generating multigroup cross sections. It is also shown that sampling all of the nuclides in a material further improves the figure of merit (at least in this example).

Table 8-5: A comparison of the relative error and figure of merit for specific group-to-group cross section bins calculated with different cross section generation techniques.

			Relative Error / FOM	
MCNP Run	$\Sigma_s$ Method	$P_n$ Method	Group 1 to 1	Group 23 to 27
slb402	exact	direct $P_n$	0.014 / 120.1	0.0038 / 1679.0
slb403	exact	$f(\mu)$ est.	0.014 / 60.4	0.0038 / 844.5
slb404	Monte Carlo	direct $P_n$	0.022 / 93.9	0.024 / 81.8
slb406	exact all isotopes	direct $P_n$	0.0079 / 333.1	0.0036 / 1548.1

The multigroup cross sections that were calculated from MCNP were used in both ONEDANT<sup>15</sup> and multigroup MCNP calculations of the same geometry. Table 8-6 compares the results of the ONEDANT and multigroup MCNP calculations with the MCNP reference calculation. In addition to the eigenvalue comparison, two values are listed that refer to group fluxes that also were calculated. The first value is the maximum percentage error in any neutron group flux relative to the MCNP reference calculation. Since the reference answers are statistical in nature (calculated with MCNP), the second value presented is the largest number of standard deviations that a calculated group flux is away from the reference value. The following formulae describe these quantities.

$$\% \text{ error} = \frac{|(\text{calc flux}) - (\text{ref. flux})|}{(\text{ref. flux})} \times 100$$

$$\text{num of SD away from ref.} = \frac{|(\text{calc. flux}) - (\text{ref. flux})|}{(\text{SD ref. flux})}$$

SD = standard deviation

If the multigroup calculated fluxes fall within three standard deviations of the reference values, the answers can be considered to be statistically equivalent. Since these comparisons seek to quantify how well the calculations using multigroup cross sections reproduce the reference values, only the maximum error and the largest number of standard deviations away from the reference are presented without regard to which group

these values represent. It is possible that the maximum percent error and the largest number of standard deviations away from the reference relate to different groups.

Table 8-6: A comparison of ONEDANT and multigroup MCNP calculated values with the continuous energy MCNP reference. The last two columns refer to group fluxes.

MCNP reference eigenvalue = $1.0291 \pm 0.00219$					
XS from	Multigroup code	P <sub>n</sub> order	k-eff	Max. % error	Largest # of SD
slb402	ONEDANT	7	1.0280	2.75	2.40
slb403	ONEDANT	7	1.0278	2.74	2.38
slb404	ONEDANT	7	1.0288	2.07	1.95
slb406	ONEDANT	7	1.0280	2.65	2.31
slb406	ONEDANT	5	1.0280	2.65	2.31
slb406	ONEDANT	3	1.0280	2.65	2.30
slb406	ONEDANT	1	1.0168	1.93	6.67
slb406	ONEDANT	0	1.2711	27.09	99.57
slb403	MCNP	7	1.0353	2.70	4.64
slb403 CRSRD	MCNP	7	1.0264	3.34	2.91

A couple of observations can be made based on the data in Table 8-6. The first observation is that the ONEDANT solution is not acceptable if the Legendre order is less than 3. This provides very strong evidence that the angular distribution is being represented correctly despite the flux weighting approximation that was made in Chapter 6. The second observation is that the multigroup group cross sections, when processed with CRSRD<sup>13</sup> (the last item in Table 8-6) and used in MCNP (see section 6.7) are also able to reproduce the reference values. The multigroup MCNP results are discussed further in the subsection below. In conclusion, the data indicate that, for this test problem, the twenty-seven group cross sections were calculated correctly and when used in a transport code the reference values were reproduced.

### **8.1.1 $f(\mu)$ Estimation and Low Probability Events**

As mentioned in Section 6.7, the multigroup cross sections can be directly output in a format for use in MCNP if the  $f(\mu)$  estimation is used for the Legendre components.

Otherwise, the multigroup cross section library can be processed with CRSRD which will convert the Legendre expansions into 32 equi-probable cosine bins for use in multigroup MCNP.

When the  $f(\mu)$  estimation is used, the laboratory angular distribution function is accumulated in 50 or more equi-width cosine bins. These bins are then integrated to yield 32 equi-probable cosine bins for use in multigroup MCNP. The integration is performed such that the first bin that has a contribution will determine the first bin boundary. This is not necessarily the best method of integration because an extremely low probability event may result in a contribution to the tail of the angular distribution which could result in misleading bin boundaries. As an example, Table 8-7 presents the angular distribution function for scattering from group 1 to 3 as calculated in MCNP run slb403. In Table 8-7 it is clearly shown that there are contributions to the  $f(\mu)$  down to -0.96. Therefore, the first bin boundary for the 32 equi-probable bins would begin at -0.96. This is misleading since the tail of the distribution is not adequately sampled and the height of these bins is negligible in comparison to the taller bins.

When the distribution in Table 8-7 is integrated to determine the Legendre components, the bins in the tail of the distribution make a negligible contribution to the integration. This can be seen when the Legendre components are processed with CRSRD and the Legendre expansion is converted into 32 equi-probable bins with the first bin boundary starting at approximately 0.32

The last two lines of data in Table 8-6 show the results when the multigroup library is used in MCNP. In the first case, the data library was output directly from the cross section generation routines and was subject to the problems described above. In the second case (labeled with CRSRD), the Legendre components were processed with CRSRD to yield the 32 equi-probable cosine bins. It does appear from the data that the use of CRSRD

improved the comparison to the reference and eliminated the bin boundary problem described above.

Table 8-7: The angular distribution function for scattering from group 1 to 3 for the infinite slab.  
The bin boundaries are the cosine of the laboratory scattering angles.

<b>Bin Boundaries</b>	<b>Height</b>	<b>Bin Boundaries</b>	<b>Height</b>
-1.00	-0.96	0.00E+00	0.00
-0.96	-0.92	8.87E-04	0.04
-0.92	-0.88	8.87E-04	0.08
-0.88	-0.84	0.00E+00	0.12
-0.84	-0.80	8.87E-04	0.16
-0.80	-0.76	0.00E+00	0.20
-0.76	-0.72	8.87E-04	0.24
-0.72	-0.68	8.87E-04	0.28
-0.68	-0.64	0.00E+00	0.32
-0.64	-0.60	8.87E-04	0.36
-0.60	-0.56	8.87E-04	0.40
-0.56	-0.52	0.00E+00	0.44
-0.52	-0.48	8.87E-04	0.48
-0.48	-0.44	0.00E+00	0.52
-0.44	-0.40	8.87E-04	0.56
-0.40	-0.36	8.87E-04	0.60
-0.36	-0.32	0.00E+00	0.64
-0.32	-0.28	8.87E-04	0.68
-0.28	-0.24	8.87E-04	0.72
-0.24	-0.20	0.00E+00	0.76
-0.20	-0.16	8.87E-04	0.80
-0.16	-0.12	0.00E+00	0.84
-0.12	-0.08	8.87E-04	0.88
-0.08	-0.04	8.87E-04	0.92
-0.04	0.00	0.00E+00	0.96
			1.00
			8.87E-04

## 8.2 Light Water Reactor Assembly

The second test problem was a Combustion Engineering 16x16 PWR fuel assembly located in a wet spent fuel storage rack cell. Reflective boundary conditions were used on all sides of the geometry, effectively creating an infinite storage cell. The fuel in the

assembly was low enriched, 1.7 weight percent  $^{235}\text{U}$ ,  $\text{UO}_2$  with a density of 10.47 gm/cc and the surrounding water had a density of 1.0 gm/cc. The storage rack cell consisted of a stainless steel box with boral panels on all four sides for reactivity control. Figure 8-1 is a drawing of the CE assembly inside the storage rack cell. Figure 8-2 is an enlargement of the upper right hand corner to help show the structure of the storage cell.

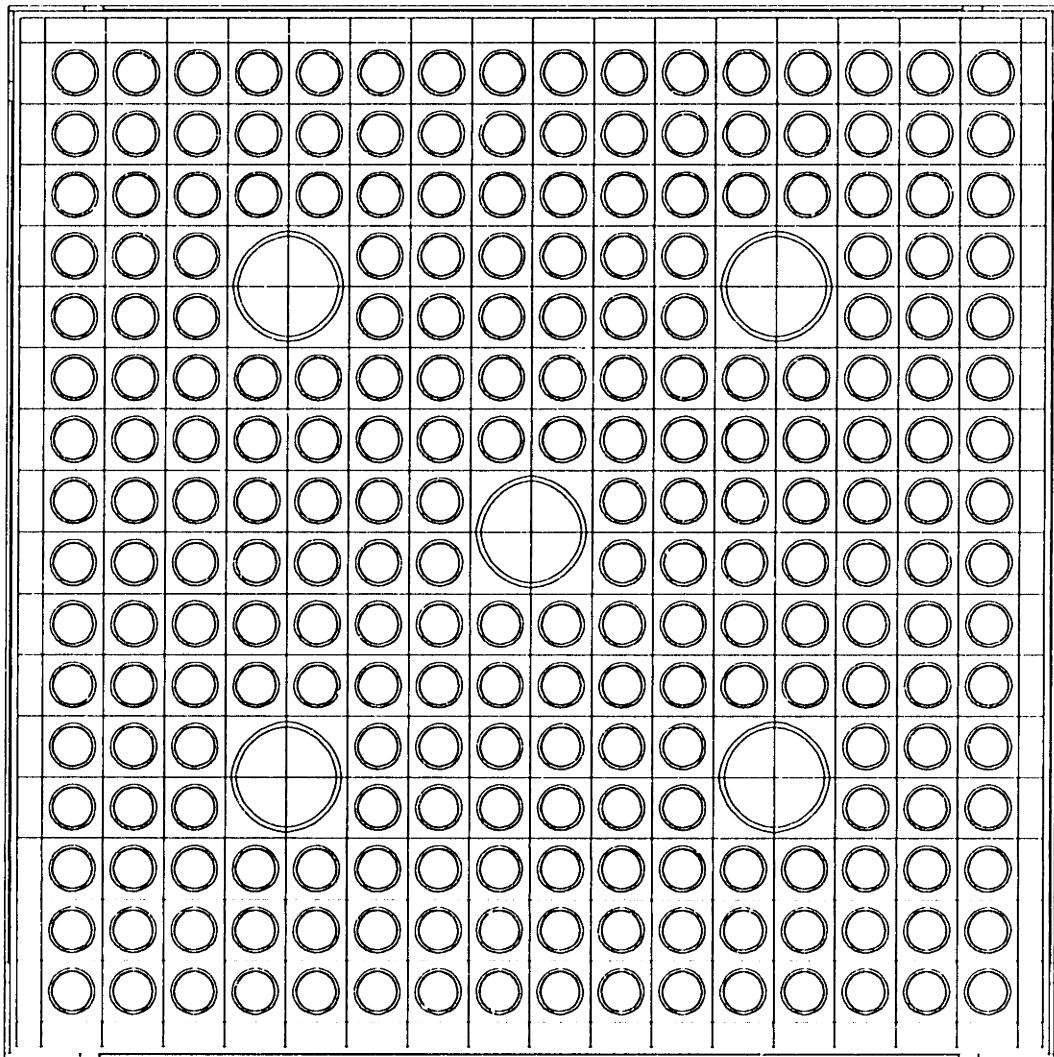


Figure 8-1: A two-dimensional drawing of the MCNP model of a CE 16x16 fuel assembly inside a wet storage rack cell. This picture was drawn with the MCNP two-dimensional plotter.

An MCNP calculation was used to generate twenty-seven group cross sections for each of the materials in the model (fuel, clad, water, stainless steel, boral). The cross sections were averaged over the materials; for example, the calculated multigroup fuel

cross section was an average cross section for all fuel pins. The neutron flux also was calculated in the fuel (averaged over all fuel pins) and averaged over the entire assembly.

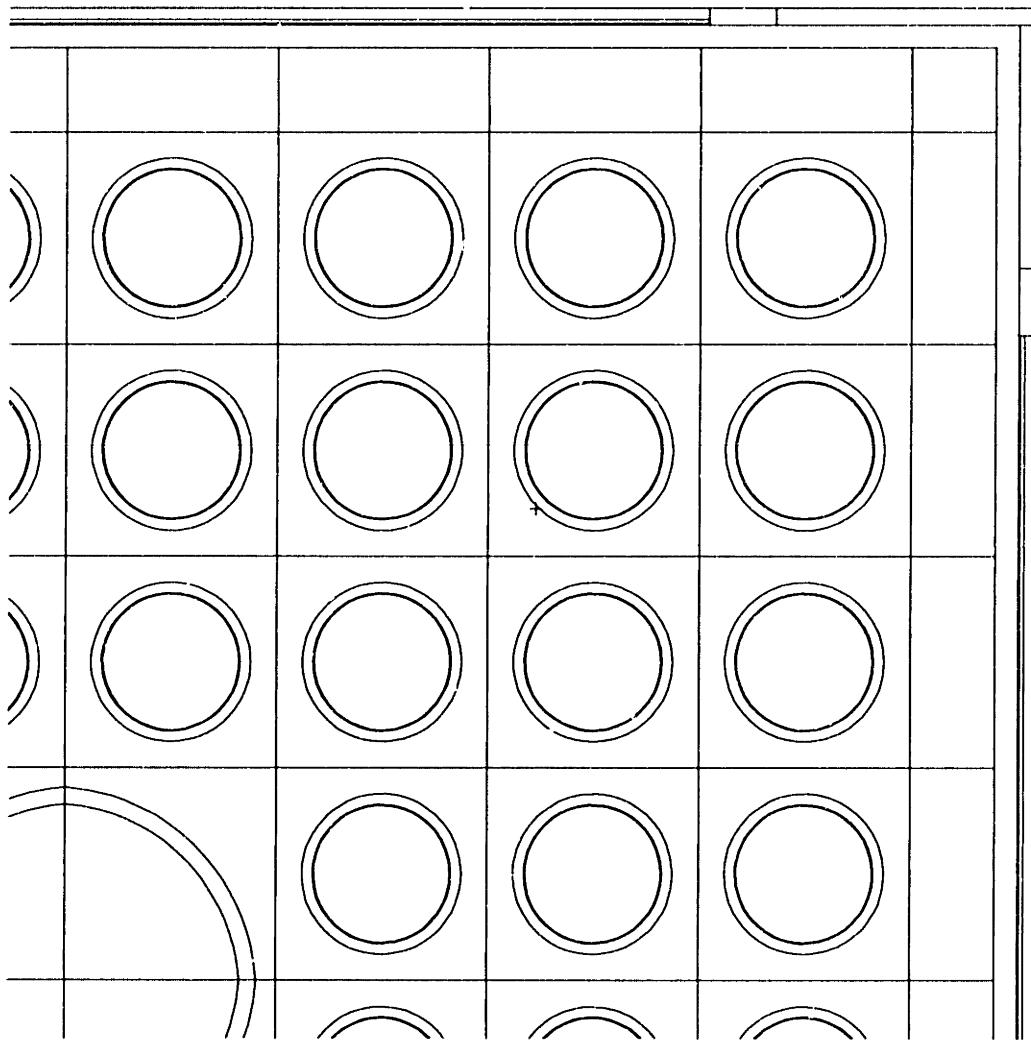


Figure 8-2: An enlarged view of the CE 16x16 fuel assembly inside a wet storage rack cell. The boral panels on the cell wall can be seen. This picture was drawn with the MCNP two dimensional plotter.

Criticality calculations were performed with 100 active cycles and 3000 neutrons per cycle. The cross section generation method used was the exact method in combination with the direct  $P_n$  method of calculating the Legendre components. Two separate cross section generation runs were performed. The first sampled a single nuclide and the second sampled all nuclides. The highest Legendre order calculated was 5 in both cases. The computer run times for the cross section generation calculations increased by a factor

of 5.3 when a single nuclide was sampled and a factor of 6.7 when all nuclides were sampled.

Both multigroup libraries were processed with CRSRD (to convert them to MCNP format) and used in a multigroup MCNP calculation of the same geometry. The eigenvalue, fuel averaged and assembly averaged fluxes were compared to the MCNP reference.

Table 8-8 presents a comparison of the reference eigenvalue with the eigenvalues calculated when the two multigroup libraries were used. The eigenvalues from the multigroup calculations are within 2 standard deviations of the reference which seems to indicate that the multigroup cross sections calculated are correct.

Table 8-8: A comparison of eigenvalues calculated with the reference MCNP calculation and multigroup MCNP calculations.

	k-effective	Standard Deviation
reference	0.90501	0.00109
multigroup	0.90455	0.00117
multigroup all isotopes sampled	0.90352	0.00116

Table 8-9 shows the results of the flux comparisons. The average and maximum percent error are shown as well as the average and maximum number of standard deviations the multigroup answer is away from the MCNP reference. The method of calculating the percent error and the number of standard deviations away from the reference was described in the Section 8.1. The table shows very good agreement between the MCNP calculated results using the multigroup libraries and the results using the continuous energy cross section data. This is further confirmation that the multigroup cross section libraries calculated are correct.

Table 8-9: The results of comparing the fluxes calculated with multigroup Monte Carlo to the reference fluxes from the continuous energy Monte Carlo calculation.

	Average % error	Maximum % error	Average # of SD	Largest # of SD
Fuel Pin Average Flux				
multigroup	0.70	2.18	1.57	5.59
multigroup all isotopes sampled	0.71	2.68	1.49	3.29
Assembly Average Flux				
multigroup	0.42	1.49	1.10	2.87
multigroup all isotopes sampled	0.47	2.02	1.19	2.75

### 8.2.1 CRSRD and Very Low Probability Events

The multigroup libraries created for this problem contained some extremely low probability events. Two such events were up-scattering (with Hydrogen) from group 11 to 10 and from group 10 to 9. The calculated cross section for these events was 8 to 10 orders of magnitude lower than total scattering cross section for those groups. The angular distribution associated with these events was almost a pure delta function with a corresponding laboratory scattering angle (cosine of the angle) of approximately 1.0. The associated  $P_n$  cross sections were almost identical to the  $P_0$  cross section. This created a problem when the multigroup data was processed with CRSRD because the subroutines in CRSRD that convert the Legendre expansion to 32 equi-probable cosine bins can not properly handle a delta function. The result is unusable garbage on the MCNP formatted cross section library.

The solution to this problem is very simple. All scattering cross sections less than 1.0E-06 times the total group scattering cross section are set to zero. This will eliminate the very low probability events that have delta functions for the angular distributions and will not affect the final answers since the cross sections that have been eliminated are insignificant in comparison to the total scattering cross section.

### 8.3 Four MITR-II Fuel Elements

The last test problem consisted of four MITR-II fuel elements arranged in a rhombus surrounded by water. Reflective boundary conditions were used in the axial direction and vacuum boundary conditions were used outside the water reflector. A triangular mesh geometry (triangle height was 8.19 cm) was overlaid onto the four fuel elements and the surrounding water. Figure 8-3 shows the triangular geometry. The four fuel elements can clearly be seen in the center of the picture. These elements were modeled explicitly, including the fuel meat, cladding and water channels.

A criticality calculation was performed using 100 active cycles and 3000 neutrons per cycle. Homogenized three group cross sections were calculated for each of the triangles shown in Figure 8-3. In addition, three group neutron currents were calculated out of the five sides of each triangle. These currents were always outgoing currents and therefore positive or zero in value. The net current for a particular triangle was calculated after the MCNP calculation was finished utilizing the outgoing currents for adjacent cells.

These homogenized three group cross sections and currents were used in a nodal diffusion code, QUARTZ.<sup>4</sup> In addition, the currents and volume averaged fluxes were used to calculate discontinuity factors which were also input into QUARTZ. The discontinuity factors are an integral part of nodal methods and permit the nodal calculation to recreate the reference eigenvalue and fluxes. A comparison of the nodal results to the MCNP reference results are provided below.

Since the currents, fluxes, and cross sections (reaction rates) were calculated with MCNP, a statistical code, it was not expected that, within an energy group, the reaction rates for production and loss of neutrons would balance. The difference between the production rate and the loss rate should be minimal; but because of the statistical nature of MCNP the difference will be non zero. If however, a reaction rate balance is forced by artificially adjusting the total cross section, nodal methods incorporating discontinuity factors should reproduce exactly the reference MCNP values. When a reaction rate balance is forced, the fluxes and currents are taken as constant and the total cross section

is adjusted to compensate for any difference. The following equations illustrate the reaction rate balance technique.

The loss term is:

$$\text{left} = J_g^{\text{out}} + \sum_{tg} \phi_g$$

$J_g^{\text{out}}$  = outgoing net current in group g

The source term is:

$$\text{right} = \sum_{g'=1}^G \sum_{s,g' \rightarrow g} \phi_{g'} + \frac{\sum_{tg} X_g v \sum_{tg} \phi_{g'}}{k - \text{eff}}$$

k-eff = eigenvalue from MCNP reference calculation

Therefore the new total cross section will be defined as:

$$\Sigma_{tg}^{\text{new}} = \Sigma_{tg} - \frac{\text{left} - \text{right}}{\phi_g}$$

Table 8-10 compares the eigenvalues from the reference MCNP calculation and two QUARTZ calculations. The first uses the three group cross sections directly and the second one uses the same cross sections except the total term has been adjusted to force reaction rate balance. The agreement between the QUARTZ and the MCNP values is excellent and when the reaction rate balance is performed the QUARTZ calculation exactly matches the reference.

Table 8-10: A comparison of eigenvalues calculated with the reference MCNP calculation and three group QUARTZ calculations.

	k-effective	Standard Deviation
MCNP reference	0.71325	0.00149
QUARTZ	0.71360	
QUARTZ reaction rate balance	0.71325	

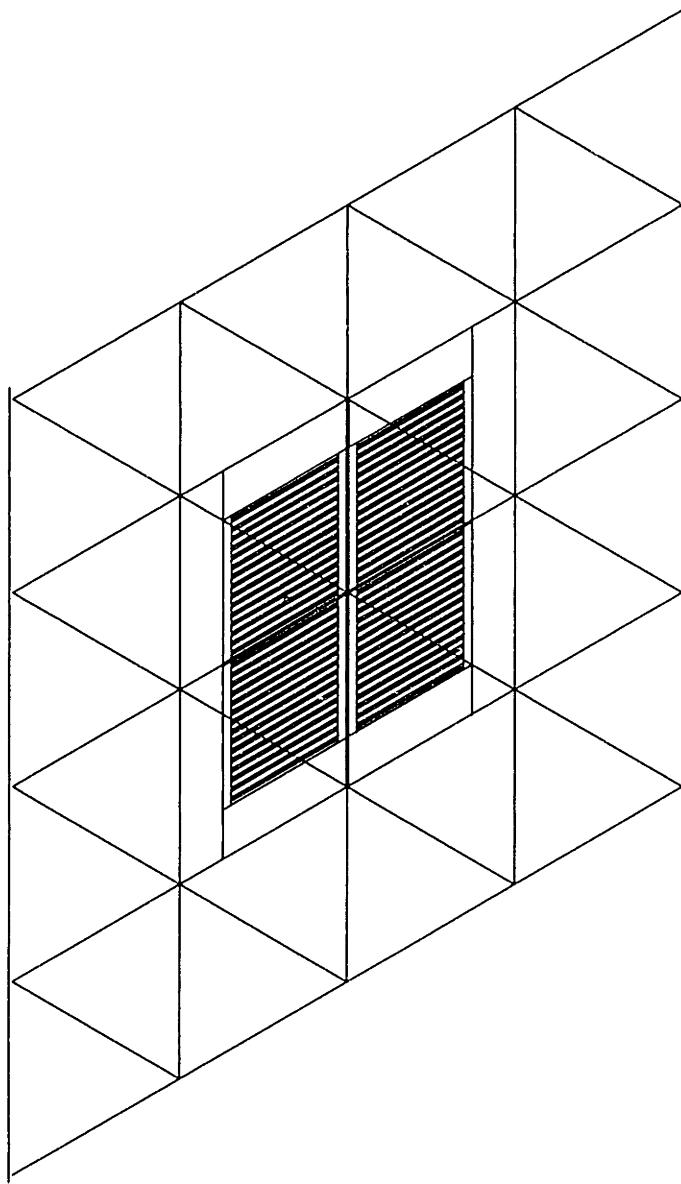


Figure 8-3: MCNP model of four MITR-II fuel elements surrounded by light water. This picture was drawn with the two-dimensional plotter from MCNP.

Table 8-11 compares the fluxes calculated from MCNP and QUARTZ. The maximum percent error in the QUARTZ values relative to the MCNP reference is shown for the flux values in the reflector region and in the core region.

The agreement between the QUARTZ and the MCNP values in the fuel region is very good and in the reflector regions is still quite good. When the reaction rate balance is

used, it is clear that QUARTZ reproduces the MCNP reference values completely. The small percent error shown is due to a slight difference in the normalization constants.

Table 8-11: The maximum percent error in flux values calculated with QUARTZ and compared to the MCNP reference values.

	Maximum % error	
	REFLECTOR	FUEL
QUARTZ	5.10	1.10
QUARTZ reaction rate balance	0.18	0.18

All of these results indicate that the cross section generation modifications to MCNP can be used to calculate accurately homogenized multigroup (or few group) cross sections and currents to be used in nodal methods.

## Chapter 9

### **Conclusions and Future Work**

In view of the very good results from the test problems that were run, it is concluded that the cross section modification to MCNP, while involving an approximation in estimating the Legendre components of the scattering cross section, is successful. These modifications offer the user the capability of generating a multigroup library with an arbitrary number of energy groups and an arbitrary Legendre order for the scattering cross section. In addition the user can generate cell to cell neutron currents. As demonstrated, these cross sections can be used in a variety of reactor physics codes with good agreement being obtained with the reference calculation.

As stated in Chapter 1, this thesis is a proof of principle and therefore MCNP was modified to produce only neutron cross sections as a demonstration that the implemented methods are correct. Future work should focus on enhancing the cross section generation modifications by reworking other scattering laws to perform the explicit approach to cross section generation. In addition, the ability to calculate photon cross sections and photon production from neutron interactions should also be included.

## References

1. J. F. Briesmeister, Editor, "MCNP - A General Monte Carlo N-Particle Transport Code, Version 4A," LA-12625-M, Los Alamos National Laboratory, 1993.
2. R. N. Blomquist, "VIM - Monte Carlo Neutron Transport Code," presented at Nuclear Criticality Technology Safety Project Workshop, Williamsburg, VA, May 10-11, 1994.
3. M. W. Mickael, "A Fast, Adaptive, Semi-Deterministic Weight Window Generator for MCNP," *Nuclear Science and Engineering*, Vol. 119 (34-43), 1995.
4. T. DeLorey, "A Transient, Quadratic Nodal Method for Triangular-Z Geometry," Ph.D. Thesis, Massachusetts Institute of Technology, 1993.
5. J. Hughes, "An Experimental Evaluation of the Instrumented Flux Synthesis Method," Ph.D. Thesis, Massachusetts Institute of Technology, 1995.
6. A. F. Henry, *Nuclear-Reactor Analysis*, The MIT Press, Cambridge MA, 1975.
7. P. R. Bevington and D. K. Robinson, *Data Reduction and Error Analysis for the Physical Sciences*, McGraw-Hill, New York, 1992.
8. K. D. Lathrop, "Mean Cosine for Thermal Energy Elastic Scattering," *Nuclear Science and Engineering*, Vol. 119 (80-86), 1995.
9. R. C. Little and R. E. Seamon, "New MENDF5 and MENDF5G," Los Alamos National Laboratory internal memorandum, X-6:RCL-86-412, 1986.
10. R. S. Baker, "XREP Module," Los Alamos National Laboratory internal memorandum, X-6:RSB-93-208, 1993.
11. Personal communication with R. S. Baker, 1994.
12. G. I. Bell and S. Glasstone, *Nuclear Reactor Theory*, Robert E. Krieger Publishing, Malabar, Florida, 1970.
13. J. C. Wagner, E. L. Redmond II, S. P. Palmtag, and J. S. Hendricks, "MCNP: Multigroup/Adjoint Capabilities," LA-12704, Los Alamos National Laboratory, 1994.
14. "SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation," NUREG/CR-0200, Rev. 4, (ORNL?NUREG/CSD-2/R4), Vols. I, II, and III, April 1995.
15. R. E. Alcouffe et al., "DANTSYS: A Diffusion Accelerated Neutral Particle Transport Code System," LA-12969-M, Los Alamos National Laboratory, 1995.

## Appendix A

### New Input Cards

The following input cards were added to MCNP for use with the new multigroup cross section generation feature.

#### **XSCON - CONTROL CARD FOR CROSS SECTION GENERATION**

Form: XSCON                  leg ixsn ixsp notused con1 con2 con3 con4 msct mggrp mnup con5

leg        = Legendre order to be output  
ixsn      = Number of neutron groups  
ixsp      = Number of photon groups (not implemented)  
notused     = Position must be filled but is not used  
con1      = 0 -  $f(\mu)$  estimation (default)  
              = 1 - direct  $P_n$  estimation  
con2      = 0 - relative error calculated (default)  
              = not equal 0 - relative error not calculated  
con3      = 0 - one isotope to do scattering with (default)  
              = 1 - cycle through all isotopes  
con4      = 0 - track MCNP - reset random numbers (default)  
              = not equal to 0 - use random numbers only once  
msct      = number of bins for  $f(\mu)$  estimation - must be multiple of 10  
              (default = 50)  
mggrp     = highest group to be reached by up-scatter (default = 1)  
mnup      = largest number of up-scatter groups (default = ixsn + ixsp - 1)  
con5      = 1 - do not generate MCNP multigroup file  
              = not equal to 1 - generate a new multigroup file (default)  
              this option only valid if con1=0

#### **XSCEL - CELL ENTRY CARD FOR CROSS SECTION GENERATION**

General Form: XSCEL C<sub>1</sub> C<sub>2</sub> (C<sub>3</sub> ... C<sub>4</sub>) C<sub>5</sub> ... C<sub>N</sub>

C<sub>i</sub> = cells in which multigroup cross sections will be calculated. Cells can be combined by enclosing them in parentheses. Tally cell can be any cell in the geometry. However, tallying within lattices is not supported

## **XSERG - GROUP BOUNDARIES FOR CROSS SECTION GENERATION**

Form: XSERG:N E<sub>1</sub> E<sub>2</sub> ... E<sub>N</sub>

Energy boundaries are input for N-1 groups. Must be increasing in energy.

## **XSCUR - CELL ENTRIES FOR NEUTRON CURRENT CALCULATION**

Form: XSCUR C<sub>1</sub> C<sub>2</sub> C<sub>3</sub> C<sub>4</sub> ... C<sub>N</sub>

These entries are in pairs, therefore there must be an even number of entries  
C<sub>1</sub> C<sub>2</sub> means the current from C<sub>1</sub> to C<sub>2</sub> is desired. MCNP will automatically  
calculate the current from C<sub>2</sub> to C<sub>1</sub> therefore this entry should not be included.  
There is no checking performed to verify that cells C<sub>1</sub> and C<sub>2</sub> are adjacent. If they  
are not - the current will be zero.

## **NEW OUTPUT FILES**

The following new output files are created when cross section generation is used.

XSOUT	output file for cross section
XSREL	output file containing relative errors for cross sections
XSCUR	calculated currents including relative errors
XST1M	type 1 MCNP multigroup file
XSX1M	xsdir file for XST1M

These file names can be changed on the command line or message block using the form  
xsout=?????. The MCNP feature NAME does not work with these files.

## **FORMAT OF OUTPUT DATA**

The xsout file is in the DTF (6E12.5) ASCII format. The xsrel file is in the same format  
but contains only the associated relative errors. The following lists the items stored in the  
first 9 positions

position	item
1	flux
2	chi
3	total scattering
4	total p1 scattering
5	diffusion coefficient. - not correct and probably should not be used
6	fission xs
7	absorption xs
8	nusigf
9	total
10 and beyond	scattering array

## Appendix B

### List of Variables

These are the variables used in the cross section generation patch

#### List of Variables

```
bdiv   - width of equi-cosine bins
bdiv1  - 1/bdiv
cazm   - azimuthal scattering angle for target in motion
crarr  - tally array for neutron current calculation
crcon  - container array for current calculation
ctar   - cosine of angle between target and neutron
etar   - energy of target nucleus
iasz   - length of tally array per bin (note the manner in which data is
         stored in xsarr is described below)
icsz   - length of container array per bin
idacr  - array containing pointers into crcon/crarr plus other info
idatc  - array containing pointers into xscon/xsarr plus other info
idatr  - array containing pointers into xscon/xsarr plus other info
ifltg  - variable to determine if tgtvelx has already been called
ileg   - parameter of max number of leg bins allowed - set to 7
ilen   - length of group data
ispo   - array of pointers for scattering matrix - dimension (mxsd(1)**2,2)
ispos  - position of within group scattering
itpos  - position of total xs
ixsf   - flag indicating that cross section generation occurring
ixsn   - number of neutron groups to be calculated
ixsp   - number of photon groups to be calculated
lacr   - offset for idacr
lawelr - used to determine if law reworked
lcexs  - array of pointers into idatc
lcra   - offset for crarr
lcrc   - offset for crcon
lcrxs  - array of pointers into idacr
lidc   - offset for idatc
lidr   - offset for idatr
lispc  - offset for ispo
llce   - offset for lcexs
llcr   - offset for lcrxs
lloc   - offset for locxs
locxs  - array of pointers into idatr
lxnt   - offset for xsnt
lxpt   - offset for xspt
lxsa   - offset for xsarr
lxsc   - offset for xscon
lxsn   - offset for xsen
lxsp   - offset for xsep
mgrp   - highest group to up-scatter to
mnup   - largest number of up-scatter groups
msct   - number of bins for legendre determination
mxsc   - entries from xscon card - length = 11
mxsd   - pre-calculated values
nbin   - parameter for maximum number of equi-cosine bins - 100
nbixs  - number of tally bins on xscl card
ncrxs  - number of entries on xscl card
nenxs  - number of entries on xscl card
xsarr  - tally array for cross section gen
xscon  - container array for cross section gen tallies
xsen   - array containing neutron energies
xsep   - array containing photon energies
xsnt   - temporary array for neutron stuff
xspt   - temporary array for photon stuff

lcexs - dimensioned to nbixs
locxs - dimensioned to mxa
lcrxs - dimensioned to mxa
```

```

ispo - dimensioned to (mxsd(1)**2,2)
idatac array
N IP IC(i) ... IC(N)

N = number of cells in this bin
IP = pointer into xscon/xsarr where data for this bin is stored
IC = user cell number that defines this bin

idatr array
IC N IP(i) ... IP(N)

IC = internal cell number, idatr(locxs(i))=i
N = number of bins that this cell is in
IP = pointer into xscon/xsarr where data for first bin is

idacr array
IIC N IC(i) ... IC(N) IP(i) ... IP(N)

IIC = internal cell number idacr(lcrxs(i))=i
N = number of cells that are adjacent to IIC that current desired
IC = adjacent cell for current calculation from IIC to IC
IP = pointer into crarr for current calc from IIC to IC

ispo array
(1,(ifr-1)*mxsd(1)+ito) = location of data after mxsd(1)*6 in a bin in xsarr
or xscon array

(2,(ifr-1)*mxsd(1)+ito) = location of angular bin data after mxsd(1)*6 in
a bin in xsarr or xscon array (only used if
mxsc(5)=0

```

#### Storage in xsarr

data	- bins (all)  -
sq data	- bins (all except Pn bins)  -
cor data	- bins (cross sections only)  -

```

# of      data per bin = mxsd(1)*6+mxsd(2)+(mxsd(2)*msct or mxsd(2)*mxsc(1))
# of    sq data per bin = mxsd(1)*6+mxsd(2)
# of cor data per bin = mxsd(1)*4+mxsd(2)

mxsd(1) = total number of groups
mxsd(2) = number of scattering positions used
mxsd(3) = icsz*nbinxs = total length of container array
mxsd(4) = mxsd(1)*4+mxsd(2) = length of correlation data per bin
mxsd(5) = mxsd(1)*6+mxsd(2)
mxsd(6) = ncrxs*mxsd(1)

```

#### Data store

position	item
1	flux
2	chi
3	sigf
4	siga
5	nusigf
6	sigtot
7 and beyond	scattering array

Data written to ascii file

```
position           item
1      flux
2      chi
3      total scattering
4      total p1 scattering
5      diffusion coef. - probably wrong
6      fission xs
7      absorption xs
8      nusigf
9      total
10 and beyond    scattering array
```

## Appendix C

### Source Code Modifications

The following modifications are in the standard MCNP patch file format. A few of the MCNP subroutines have been copied into these modifications and renamed. In these subroutines, the new coding can be easily identified, in most cases, as the uppercase text.

```
*ident zcxs1
*d,zc4a.9
 3 nbmx=100,ncolor=10,ndef=19,novr=5,nptra=13,nsp=602,nsp12=nsp+12,
*d,zc4a.12
 6 iut=45,iuz=46,iuk=47,iu3=48,iu4=49,iupw=50,iupc=51,iuxo=61,
 6 iuxr=62,iuxc=63,iuxt=64,iuxx=65,ileg=7,nbin=100,
*/
/* ---- vv comdeck
*ident vvxsl
*d,vv4a.13
 1 rsss,xsdir,com,dumn1,dumn2,xsout,xsrel,xscur,xst1m,xsx1m,
 1 isub(ndef)
*d,vv4a.15
 1 plotm,rsss,xsdir,com,dumn1,dumn2,xsout,xsrel,xscur,xst1m,xsx1m
*/
*addfile,vv
*comdeck cmx
  common /scatd/etar,ctar,cazm,w4,ergolde
  common /scati/ ifltg,im1,ig,lawelr
*comdeck cmx1
  common /scatd1/ a2,a3,a4,a5,b,c,d,d1,d2,d3,d4,d5
*/ ---- cm comdeck
*ident cmxsl
*/
/* statically allocated common
*d,cm4a.1,cm4a.2
  parameter (nfixcm=maxi+3*maxv+mtop+mipt*(24+mxdt+7*mxidx)+nsp+80,
  1 lfixcm=3*mxdt+mink+11*mipt+2*maxv+2*maxf+299)
*d,cm4a.5
  2 hsb(nsp),bdiv,bdiv1,
*i,cm4a.7
  7 nbinsx,ixsf,ixsn,ixsp,nenxs,icsz,iasz,mxsc(12),mxsd(6),ncrxs,
  7 msct,mgrp,mnup,
*/
/* pointers
*i,cm4a.14
  5 llce,lloc,lldr,lidc,lxsn,lxsp,lxsc,lxsa,llcr,lacr,lcrc,lcra,
  5 lisp,lxnt,lxpt,
*d,cm4a.29
  2 hsb(nsp),bdiv,bdiv1,
*i,cm4a.34
  7 nbinsx,ixsf,ixsn,ixsp,nenxs,icsz,iasz,mxsc(12),mxsd(6),ncrxs,
  7 msct,mgrp,mnup,
*i,cm4a.41
  5 llce,lloc,lldr,lidc,lxsn,lxsp,lxsc,lxsa,llcr,lacr,lcrc,lcra,
  5 lisp,lxnt,lxpt,
*/
*d,cm4a.61
  7 (kstt,lstt),(kxsc,lxsc),(kxsa,lxsa),(kcrc,lcrc),(kcra,lcra)
/* add sigf and a sigf variable
*d,cm.175
  2 rnrtc,rnrtc0,sff(3,maxf),siga,sigf,asigf,smultc(3),ssr,stp,
  2 sumktc(3),
*/
*d,cm4a.68
*if def,multt.or.multp,4
*i,cm4a.69
  8 kxsc,kxsa,kcrc,kcra,
```

```

*d,cm4a.70
    parameter (ntskcm=102*mipt+40*mxlv+3*maxf+mlgc+186,
*d,cm4a.72
    1 ltskcm=mipt*(2+8*mwdx)+4*mxlv+2*mlgc+maxf+89)
*/
/* fixed dynamically allocated common
*i,cm.213
    4 xsen(1),xsep(1),xsnt(1),xspt(1),
*i,cm4a.75
    6 lcexs(1),locxs(1),idatr(1),idatc(1),lcrxs(1),idacr(1),ispo(2,1),
*d,cm4a.76
*if -def,pointer,10
*i,cm.224
    3 xsen,xsep,xsnt,xspt,
*d,cm4a.77
    6 ksu,lcexs,locxs,idatr,idatc,lcrxs,idacr,ispo,
*d,cm4a.78
*if def,pointer,22
*i,cm.239
    4 (kdy,xsen),(kdy,xsep),(kdy,xsnt),(kdy,xspt),
*d,cm4a.79
    6 (kdy,ksu),(kdy,lcexs),(kdy,locxs),(kdy,idatr),(kdy,idatc),
    6 (kdy,lcrxs),(kdy,idacr),(kdy,ispo),
*i,cm.282
c      cross section generation tallies
    dimension xscon(1),xsarr(1),crcon(1),crarr(1)
*if -def,pointer,1
    equivalence (das,xscon,xsarr,crcon,crarr)
*if def,pointer,1
    pointer (kdy,xscon),(kdy,xsarr),(kdy,crcon),(kdy,crarr)
*/
/* ----- blkdat
*ident bdxs1
*d,bd4a.10
    1 'srctp','plotm','rssa','xsdir','com','dumn1','dumn2','xsout',
    1 'xsrel','xscur','xstlm','xsxlm',//,
*/
/* ----- mcnp
*ident mcs1
*i,mc.32
c      iuxo      cross section output file      s/f  output only
c      iuxr      cross sect. rel error file   s/f  output only
c      iuxc      cross sect. current file   s/f  output only
c      iuxt      cross sect. type 1 file    s/f  output only
c      iuxx      cross sect. type 1 rel err. s/f  output only
*/
/* ----- tpefil
*ident tfxs1
*i,tf.27
    do 42 i=1,(icsz*nbinxs)
 42 xscon(lxsc+i)=0.
    do 43 i=1,iasz*nbinxs
 43 xsarr(lxsa+i)=0.
    do 44 i=1,ncrxs*mxsd(1)
 44 crcon(lcrc+i)=0.
    do 45 i=1,ncrxs*mxsd(1)*2
 45 crarr(lcra+i)=0.
*/
/* ----- runtpw
*ident rrxs1
*i,rr4a.3
    call fastdr(iu,xsarr(lxsa+1),iasz*nbinxs)
    call fastdr(iu,crarr(lcra+1),ncrxs*mxsd(1)*2)
*/
/* ----- runtpw
*ident rwxsl
*i,rw4a.2
    call fastdw(iu,xsarr(lxsa+1),iasz*nbinxs)
    call fastdw(iu,crarr(lcra+1),ncrxs*mxsd(1)*2)
*/
/* ----- jc comdeck
*d,jc4a.1
    parameter (nkcd=91,ntalmx=100,mopts=5)
*/
/* ----- ibldat
*ident ibxs1
*i,ib4a.7
    data cnm(88),(krq(i,88),i=1,7) //'xscon',0,0, 0,1, 2, 12,0/
    data cnm(89),(krq(i,89),i=1,7) //'xscel',0,0, 0,0, 0, 0,0/
    data cnm(90),(krq(i,90),i=1,7) //'xserg',0,1, 0,0, 1, 0,0/

```

```

      data cnm(91),(krq(i,91),i=1,7)/*xscur',0,0, 0,0, 2,    0,0/
*/
*----- imcn
*ident imxs1
*i,im.148
c      calculate array sizes for xs calculation
  if(ixsf.ne.0)then
  mxsd(1)=ixsn+ixsp
  ic=1
c  set up ispo array
  do 273 i=1,mxsd(1)
  do 272 j=1,mxsd(1)
  ip=(i-1)*mxsd(1)+j
  if(j.lt.i)then
  if(j.lt.mgrp)go to 272
  if((i-j).gt.mnup)go to 272
  endif
  ispo(lisp+1,ip)=ic
  ic=ic+1
272 continue
273 continue
  mxsd(2)=ic-1
c  bin method of legendre calc
  if(mxsc(5).eq.0.and.mxsc(1).ne.0)then
  ic=ic-msct
  do 275 i=1,mxsd(1)
  do 274 j=1,mxsd(1)
  ip=(i-1)*mxsd(1)+j
  if(ispo(lisp+1,ip).eq.0)go to 274
  ic=ic+msct
  ispo(lisp+2,ip)=ic
274 continue
275 continue
  endif
  icsz=mxsd(1)*6+mxsd(2)*(1+mxsc(1))
  if(mxsc(5).eq.0.and.mxsc(1).ne.0)icsz=icsz+mxsd(2)*
  1 (msct-mxsc(1))
c  the part after icsz+mxsd(4) is for correlation data
c  between val and flux
c  no rel error on legendre moment calculated for technical and
c  space limitation reasons
  mxsd(3)=icsz*nbinxs
  if(mxsc(6).eq.0)mxsd(4)=mxsd(1)*4+mxsd(2)
  if(mxsc(6).eq.0)mxsd(5)=mxsd(1)*6+mxsd(2)
  mxsd(6)=ncrxs*mxsd(1)
  iasz=icsz+mxsd(4)+mxsd(5)
  bdiv=2.0d0/msct
  bdiv1=one/bdiv
c      this is done to have a way of determining if the associated
c      energy card was input
  ixsn=-ixsn
  ixsp=-ixsp
  endif
*i,im.162
  if(ixsf.ne.0.and.ixsf.ne.2.and.ixsf.ne.12)
  1 call erprnt(1,1,0,0,0,0,0,
  2 '47hxcon and xscel must both be present for xs gen')
  if(ixsn.lt.0)call erprnt(1,1,0,0,0,0,0,
  1 '22hxserg:n is not present')
  if(ixsp.lt.0)call erprnt(1,1,0,0,0,0,0,
  1 '22hxserg:p is not present')
*d,im4a.63
c
  lxsc=ltal+(nmxf*mxf+ktls)*(mtasks+1)+3
  lxsa=lxsc+icsz*nbinxs*(mtasks+1)
  lcrc=lxsa+iasz*nbinxs*(mtasks+1)
  lcra=lcrc+mxsd(6)*(mtasks+1)
  lgbn=lcra+mxsd(6)*2*(mtasks+1)
*/
*----- newcd1
*ident nfxs1
*d,nf.70
  2      9910,9920,9930,340,340,110,350)ica-55
*i,nf.168
c >>>  cross section generation cards           xscon,xscel
  340 ixsf=ixsf+1
c      need to accumulate total fission for xs gen
  itfxs=1
  return
c >>>  cross section generation cards           xscur

```

```

350 ixsf=ixsf+10
      return
c
*/
/* ----- nxtit1
*ident nxxs1
*d,nx.14
      2      9910,9920,9930,520,540,10,10)ica-55
*i,nx.244
c >>>> cross section generation cards           xscon
  520 mxsc(nwc)=abs(iitm)
      if(nwc.eq.2)ixsn=abs(iitm)
c   photon cross sections not handled yet
      if(nwc.eq.3)ixsp=0
      if(nwc.eq.4)mxsc(nwc)=0
      if(nwc.eq.9)msct=abs(iitm)
      if(nwc.eq.10)mgrp=abs(iitm)
      if(nwc.eq.11)mnup=abs(iitm)
      return
c
c >>>> cross section generation cards           xscel
c   m1c  character counter
c   m2c  number of bins
c   m3c  number of entries
c   m4c  flag to indicate that a character has occurred
  540 go to (550,560,570) kitm+1
c
  550 if(m4c.ne.0)go to 555
      m4c=1
      m2c=m2c+1
      return
  555 m4c=0
  560 return
  570 m3c=m3c+1
      if(m4c.eq.0)m2c=m2c+1
      return
c
*/
/* ----- oldcd1
*ident olxs1
*d,ol.16
      2      9910,9920,9930,360,370,10,380)ica-55
*i,ol.156
c >>>> cross section generation cards           xscon
  360 if(msct.eq.0)msct=50
      if(mgrp.eq.0)mgrp=1
      if(mnup.eq.0)mnup=ixsn+ixsp-1
      return
c >>>> cross section generation cards           xscel
  370 nbins=m2c
      nenxs=m3c
      return
c >>>> cross section generation cards           xscur
  380 ncrxs=nwc
      return
c
*/
/* ----- setdas
*ident sdxs1
*d,sd.90
      linsn=lxnm+nmat1*ke
      lxsp=linsn+ixsn+1
      lxnt=lxsp+ixsp+1
      lxpt=lxnt+ixsn
      lfcdg=lxpt+ixsp
*d,sd.134
      llce=lnxs+16*mxe1
      llloc=llce+nbins
      llisp=llloc+mxa
      lidr=llisp+((ixsn+ixsp)**2)*2
      lidc=lidr+nenxs*3+1
      llcr=lidc+nbins*(2+nenxs)
      lacr=llcr*mxa
      lfcdj=lacr+4*ncrxs
*/
/* ----- newcrd
*ident nexs1
*d,ne.94
      2      9910,9920,9930,150,640,650,680)ica-55
*i,ne4a.65

```

```

c
c >>>> cross section generation cards                                xscel
c           m1c is a counter for the position within xscon
c           m6c is a counter for the position within idatc
c           m7c is a counter for the position within idatr
 640 m1c=1
  m6c=1
  m7c=1
  return
c
c >>>> cross section generation cards                                xserv
 650 do 660 i=1,mipt
 660 if(nqp(i).eq.1)m1c=m1c+1
    if(m1c.gt.1)call erprnt(2,1,0,0,0,0,0,1,
    1 '39honly one particle designator is allowed')
    if(nqp(2).eq.1)call erprnt(2,2,0,0,0,0,0,1,
    1 '37hxgen for photons is not supported')
    if(nqp(3).eq.1)call erprnt(2,2,0,0,0,0,0,1,
    1 '37hxgen for electrons is not supported')
    if(nqp(1).eq.1)ixsn=abs(ixsn)
    if(nqp(2).eq.1)ixsp=abs(ixsp)
    if(nqp(1).eq.1.and.kpt(1).eq.0)go to 670
    if(nqp(2).eq.1.and.kpt(2).eq.0)go to 670
    return
 670 call erprnt(2,2,0,0,0,0,0,1,
  1 '43hthis card is extraneous and will be ignored')
  return
c
c >>>> cross section generation cards                                xscur
c           m1c=cell current is exiting
c           m2c=cell current is entering
c           m3c=counter from 1 to 2
c           m4c=position within idacr
c           m5c=position within crcon
c
 680 ir=mod(ncrxs,2)
    if(ir.ne.0)call erprnt(2,1,0,0,0,0,0,1,
    1 '39hthere must be an even number of entries')
    m3c=0
    m4c=1
    m5c=1
    return
/*
/* ----- chekit
*ident cexs1
*d,ce.29
 2      9910,9920,9930,1180,1190,10,1240)ica-55
*i,ce4a.241
c
c >>>> cross section generation cards                                xscon
 1180 if(iitm.lt.0) go to 9010
    if((nwc.ge.5.and.nwc.le.7).and.(iitm.gt.1)) go to 9010
    if(nwc.eq.1.and.iitm.gt.ileg)call erprnt(2,1,1,iitm,0,0,0,0,
    1 '44hthe parameter ileg (max leg order) must be >,i5')
    if(nwc.eq.9)then
      if(mod(msct,10).ne.0)call erprnt(2,1,1,iitm,0,0,0,0,
      1 '43hthe value of msct must be multiple of ten: ,i5')
      if(msct.gt.nbin)call erprnt(2,1,1,iitm,0,0,0,0,
      1 '45hthe parameter nbin (num of cos bin) must be >,i5')
    endif
    if(nwc.eq.10)then
      iz=abs(ixsn)+abs(ixsp)
      if(mgrp.gt.iz) go to 9010
    endif
    if(nwc.eq.11)then
      iz=abs(ixsn)+abs(ixsp)
      if(mnup.gt.(iz-1))go to 9010
    endif
    return
c
c >>>> cross section generation cards                                xscel
c           m4c is for parenthesis order
c           m8c indicates if this is a multi cell bin (1) or single (0)
 1190 go to(1200,1210,1220)kitm+1
 1200 k=index('()',hitm(1:1))
    if(k.eq.0)go to 9010
    if(k.eq.1)m4c=m4c+1
    if(k.eq.2)m4c=m4c-1
    if(m4c.lt.0.or.m4c.gt.1)go to 1230
    m8c=m4c

```

```

        return
1210 call erprnt(2,1,0,0,0,0,0,1,'25hentries must be integers.')
        return
1220 m5c=namchg(1,iitm)
        if(m5c.eq.0)call erprnt(2,1,1,iitm,0,0,0,1,
1 '19hinvalid cell entry ,i5')
        return
1230 call erprnt(2,1,0,0,0,0,0,1,'26hinvalid parentheses order.')
        return
c
c >>>> cross section generation cards
1240 m2c=namchg(1,iitm)                                xscur
        if(m2c.eq.0)call erprnt(2,1,1,iitm,0,0,0,1,
1 '19hinvalid cell entry ,i5')
        m3c=m3c+1
        if(m3c.eq.3)m3c=1
        if(m3c.eq.1)m1c=m2c
        return
c
*/
/* ----- nextit
*ident nyxs1
*c,ny4a.1
2      1480,1490,1510,9910,9920,9930,10,1620,1730,1740)ica-55
*i,ny4a.125
c
c >>>> cross section generation cards
c           m1c is a counter for the position within xscon
c           m2c is the number of the bin currently being done
c           m3c is a count of the number of cells in this bin
c           m4c is for parenthesis order
c           m5c is used for internal cell number
c           m6c is counter for position within idatc
c           m7c is counter for position within idatr
c           m8c indicates if this is a multi cell bin (1) or single (0)
c
c           process parentheses first
1620 if(kitm.ne.0)go to 1640
c           open parenthesis
        if(m8c.ne.1)go to 1630
        m3c=0
        m2c=m2c+1
        lcexs(l1ce+m2c)=m6c
        idatc(lidc+m6c+1)=m1c
        return
c           close parenthesis
1630 idatc(lidc+m6c)=m3c
        m6c=m6c+m3c+2
        m1c=m1c+icsz
        return
c           case of single cell per bin
1640 if(m8c.ne.0)go to 1650
        m2c=m2c+1
        lcexs(l1ce+m2c)=m6c
        idatc(lidc+m6c)=1
        idatc(lidc+m6c+1)=m1c
        idatc(lidc+m6c+2)=iitm
        m6c=m6c+3
        go to 1660
c           case of multiple cells per bin
1650 m3c=m3c+1
        idatc(lidc+m6c+m3c+1)=iitm
c           fill idatr array
1660 if(locxs(lloc+m5c).ne.0)go to 1670
        locxs(lloc+m5c)=m7c
        idatr(lidr+m7c)=m5c
        idatr(lidr+m7c+1)=1
        idatr(lidr+m7c+2)=m1c
        m7c=m7c+3
        go to 1720
c           fill idatr when data already exists for this cell
c           readjust size of idatr array
c           id = beginning of data in idatr for this cell
c           id1 = position where new data will be stored
c           id2 = (end position of data in idatr) plus one
c           id3 = counter
1670 id=locxs(lloc+m5c)
        id1=id+idatr(lidr+id+1)+2
        id2=id1-1
1680 id2=id2+1

```

```

        if(idatr(lidr+id2).ne.0)go to 1680
        if(id2.eq.id1)go to 1700
c          shift everything by one
        id3=id2
1690  id3=id3-1
        idatr(lidr+id3+1)=idatr(lidr+id3)
        if(id3.ne.id1)go to 1690
c          store new data
1700  idatr(lidr+id1)=mlc
        idatr(lidr+id1+1)=idatr(lidr+id1+1)+1
c          realign locxs array
1710  id1=id1+1
        if(id1.gt.id2)go to 1715
        locxs(lloc+idatr(lidr+id1))=id1
        id1=id1+idatr(lidr+id1+1)+1
        go to 1710
1715  m7c=m7c+1
1720  if(m8c.eq.0)mlc=mlc+icsz
        return
c
c >>>> cross section generation cards
1730  if(nqp(1).eq.1.and.nwc.le.(ixsn+1))xsen(lxsn+nwc)=ritm           xseng
      if(nqp(2).eq.1.and.nwc.le.(ixsp+1))xsep(lxsp+nwc)=ritm
      return
c
c >>>> cross section generation cards
1740  if(m3c.eq.1)return                                     xscur
c
        m10c=0
1742  if(lcrxs(llcr+m1c).ne.0)go to 1750
        lcrxs(llcr+m1c)=m4c
c          set first position equal to cell exiting
        idacr(lacr+m4c)=m1c
        idacr(lacr+m4c+1)=1
        idacr(lacr+m4c+2)=m2c
        idacr(lacr+m4c+3)=m5c
        m4c=m4c+4
c          safety check
        if(lacr+m4c.ge.1fcadj)call erprnt(1,1,0,0,0,0,0,1,
1 '34error in allocation of idacr array' )
        if(ics.eq.-1)return
c
        m5c=m5c+mxsd(1)
        go to 1800
1750  m6c=lcrxs(llcr+m1c)
c          search to see if this pairing already exists
        do 1755 i=1,idacr(lacr+m6c+1)
        if(m2c.eq.idacr(lacr+m6c+1+i))call erprnt(2,1,2,mlc,m2c,0,0,1,
1 '28hthis pairing already exists ,i5,3h : ,i5')
        if(ics.eq.-1)return
1755  continue
c          need to shift all the data after current cell by 2
c          find empty data
c          id1=position of first data to be shifted
c          id2=position to store data in
        id1=m6c+idacr(lacr+m6c+1)*2+2
        id2=id1-1
1760  id2=id2+1
c          safety check
        if(lacr+id2.ge.1fcadj)call erprnt(1,1,0,0,0,0,0,1,
1 '34error in allocation of idacr array' )
        if(ics.eq.-1)return
c
        if(idacr(lacr+id2).ne.0)go to 1760
        id3=id2
        id2=id2+2
        id4=id2
        if(id3.eq.id1)go to 1773
1765  id2=id2-1
        id3=id3-1
        idacr(lacr+id2)=idacr(lacr+id3)
        if(id3.ne.id1)go to 1765
c          everything is now shifted by two and must realign lcrxs array
1770  lcrxs(llcr+idacr(lacr+id2))=id2
        id2=id2+idacr(lacr+id2+1)*2+2
        if(id2.ne.id4)go to 1770
c          time to store new data
1773  idacr(lacr+id1+1)=m5c
        m5c=m5c+mxsd(1)
        do 1775 i=1,idacr(lacr+m6c+1)

```

```

        idacr(lacr+id1)=idacr(lacr+id1-1)
1775  id1=id1-1
        idacr(lacr+id1)=m2c
        m4c=m4c+2
        idacr(lacr+m6c+1)=idacr(lacr+m6c+1)+1
c      do reverse pairing
1800  m10c=m10c+1
        if(m10c.eq.2) return
        m6c=m1c
        m1c=m2c
        m2c=m6c
        go to 1742
c
*/
/* ----- oldcrd
*ident ocxs1
*d,oc.16
    2      9910,9920,9930,10,730,740,10)ica-55
*i,oc4a.90
c
c >>>> cross section generation cards
730 if(m4c.ne.0)call erprnt(2,1,0,0,0,0,0,1,
1 '26hinvalid parenthesis order.')
return
c
c >>>> cross section generation cards
740 if(nqp(1).eq.1)m1c=lxsn+1
        if(nqp(2).eq.1)m1c=lxsp+1
        if(m1c.gt.1.and.nwc.ne.m1c)call erprnt(2,1,2,nwc,m1c,0,0,0,
1 '20hnumber of entries is,i3,22h, and should have been,i3')
        if(nwc.lt.m1c)m1c=nwc
c      check array
        if(nqp(2).eq.1)go to 760
        do 750 i=2,m1c
750 if(xsen(lxsn+i).le.xsen(lxsn+i-1))go to 9010
        return
760 do 770 i=2,m1c
770 if(xsep(lxsp+i).le.xsep(lxsp+i-1))go to 9010
        return
c
*/
/* ----- xact
*ident xaxs1
*d,xa4a.5
    1 (2*nspt+5)*ntal)*(mtasks+1)*ndp2+
    2 ((icsz+iasz)*nbixns*(mtasks+1)*ndp2+3*ndp2)
    3 .mxsd(6)*3*(mtasks+1)*ndp2,
*/
/* ----- utask
*ident utxs1
*i,ut.64
    kxsc=lxsc+(ktask+1)*icsz*nbixns
    kxsa=lxsa+(ktask+1)*iasz*nbixns
    kcrc=lcrc+(ktask+1)*mxsd(6)
    kcra=lcra+(ktask+1)*mxsd(6)*2
*i,ut4a.15
    do 56 i=1,icsz*nbixns
56  xscon(kxsc+i)=0.
    do 57 i=1,iasz*nbixns
57  xsarr(kxsa+i)=0.
    do 58 i=1,mxsd(6)
58  crcon(kcrc+i)=0.
    do 59 i=1,mxsd(6)*2
59  crarr(kcra+i)=0.
*/
/* ----- vtask
*ident vtxs1
*i,vt.111
    do 290 i=1,iasz*nbixns
    xsarr(lxsa+i)=xsarr(lxsa+i)+xsarr(kxsa+i)
290  xsarr(kxsa+i)=0.
    do 295 i=1,mxsd(6)*2
    crarr(lcra+i)=crarr(lcra+i)+crarr(kcra+i)
295  crarr(kcra+i)=0.
*/
/* ----- msgcon
*ident mexs1
*d,me4a.247
    if(ixsf.eq.0)go to 362
    call mgetd(xsarr(kxsa+1),iasz*nbixns,i)

```

```

    call mgetd(crarr(kcra+1),mxsd(6)*2,i)
362 if(ntal.eq.0)go to 365
*d,me4a.410
    if(ixsf.eq.0)go to 502
    call mputd(xsarr(kxsa+1),iasz*nbinxs,i)
    call mputd(crarr(kcra+1),mxsd(6)*2,i)
502 if(ntal.eq.0)go to 505
*/
/* ----- hstory
*ident hsxs1
*i,hs.108
c      track length method of cross section generation
    if(mxsc(1+ipt).ne.0)then
        l=locxs(lloc+icl)
        if(l.ne.0) call mulstor(1,l,d)
        do 125 ie=0,lev-1
            l=locxs(lloc+int(udt(7,ie)))
125 if(l.ne.0) call mulstor(1,l,d)
        endif
c
*i,hs.274
    if(ixsf.ne.0)call mulstor(3,0,zero)
*/
/* ----- surfac
*ident suxs1
*i,su.5
    dimension lp(0:mxlv)
    do 5 ie=0,mxlv
5 lp(ie)=0
*d,su.27,su4a.11
c      check if cell leaving is requesting current tally
    if(ixsf.ne.12.or.mxsc(1+ipt).eq.0)go to 57
c
    11=lcrxs(llcr+icl)
    if(11.ne.0) lp(lev)=11
    do 53 ie=0,lev-1
        11=lcrxs(llcr+int(udt(7,ie)))
53 if(11.ne.0) lp(ie)=11
    call newcel(cs)
c      check new cell to see if receiving current
    do 56 ie=0,mxlv
    if(lp(ie).eq.0)go to 56
c      im = number of possible entering cells
    im=idacr(lacr+lp(ie)+1)
c      check cell going into to see if one of any level
    do 55 i1=1,im
c      ih = cell going into
    ih=idacr(lacr+lp(ie)+1+i1)
    11=iap
    if(11.eq.ih)call mulstor(2,idacr(lacr+lp(ie)+1+i1+im),zero)
    do 54 i2=0,lev-1
        11=int(udt(7,i2))
54 if(11.eq.ih)call mulstor(2,idacr(lacr+lp(ie)+1+i1+im),zero)
55 continue
56 continue
    go to 58
57 call newcel(cs)
58 if(kdb.ne.0)go to 215
*/
/* ----- acetot
*ident atxs1
/* add sigf and asigf variable
*i,at.24
    sigf=0.
    asigf=0.
*d,at.155,at.156
290 do 300 m=jmd(1jmd+mk),jmd(1jmd+mk+1)-1
*i,at.163
    sigf=sigf+rtc(krtc+8,iex)*fme(lfme+m)
    asigf=asigf+rtc(krtc+10,iex)*rtc(krtc+8,iex)*fme(lfme+m)
c      use rtc(9) for sigf storage and do not clear rtc(10)
    rtc(krtc+9,iex)=rtc(krtc+8,iex)
    if(lfc1(lfc1+icl).eq.0) rtc(krtc+8,iex)=0.
*/
/* ----- photot
*ident ptxs1
*i,pt.7
    sigf=0.
    asigf=0.
*/

```

```

/* ----- output
*ident ouxs1
*i,ou.44
c      call output for xs gen
      if(ixst.ne.0)call xsoutp
*/
*addfile,za
*deck mulst
      subroutine mulstor(ip,im,dt)
c
c      this subroutine stores the multigroup data into the appropriate
c      arrays
c      ip
c      1 - flux, total, abs, fiss, nu*fiss
c      2 - storage of current data
c      3 - flush container array
c      im = pointer into idatr
c      dt = track length
c
*ccall cm
*call cmx
c
c      dimension vr(3)
c      return if kcode problem is not settled
c      if(nsr.eq.71.and.kcy.le.ikz)return
c      im1=im+1
c
c      go to (100,200,300) ip
900  return
c
c      energies are stored in monotonically increasing order
c      with all group boundaries given
c      partial photon group programming
100  mk=mat(lmat+ic1)
      if(mk.eq.0)return
      if(ipt.eq.2) go to 115
      if(erg.lt.xsen(lxsn+1))return
      do 105 i=2,ixsn+1
105  if(erg.lt.xsen(lxsn+i))go to 110
      return
110  ig=i-1
c      groups are reversed
c      fast=1
c      thermal=n
c      ig=ixsn+1-ig
      go to 130
115  if(erg.lt.xsep(lxsp+1))return
      do 120 i=2,ixsp+1
120  if(erg.lt.xsep(lxsp+i))go to 125
      return
c      photon groups are below neutron groups
125  ig=1-1
      ig=ixsp+1-ig+ixsn
130  it=mxsd(1)
c      flux - flux estimator
      w1=wgt*dt
c      need macroscopic cross section reaction rate
      w2=w1*rho(lrho+ic1)
c      various reactions
      w3=w2*sigf
      w4=w2*siga
      w5=w2*asigf
      w6=w2*totm
c      cycle through all tally bins that contain this cell
      do 135 i=1,idatr(lidr+im1)
      jp=kxsc+idatr(lidr+im1+i)+ig-1
      xscon(jp)=xscon(jp)+w1
      jp=jp+it*2
      xscon(jp)=xscon(jp)+w3
      jp=jp+it
      xscon(jp)=xscon(jp)+w4
      jp=jp+it
      xscon(jp)=xscon(jp)+w5
      jp=jp+it
135  xscon(jp)=xscon(jp)+w6
c      doing chi value and then scattering array
c      save the random number stuff to be able to reset history
      if(mxsc(8).eq.0)then
      ranbelr=ranb
      ranselr=rans

```

```

rnrtcelr=rnrtc
endif
wgtolde=wgt
ergolde=erg
uold=uuu
vold=vvv
wold=www
m=jmd(ljmd+mk)
m1=m
c skip chi if no fission cross section
if(sigf.eq.0.)go to 170
if(mxsc(7).eq.1)go to 163
c calculate chi for a single isotope
if(npq(lnpq+mk).eq.1)go to 145
c=rang()*sigf
c the -2 is taking advantage of fortran
do 140 m=m1,jmd(ljmd+mk+1)-2
c select fissionable nucleus
c=c-rtc(krtc+9,lme(llme+1,m))*fme(lfme+m)
140 if(c.lt.0.)go to 145
145 iex=lme(llme+1,m)
c use mcnp routine to get all information needed
rtc(krtc+2,iex)=rtc(krtc+1,iex)
krtc(kktc+2,iex)=kktc(kktc+1,iex)
t1=rtc(krtc+8,iex)
rtc(krtc+8,iex)=rtc(krtc+9,iex)
c must set fission to capture temporarily
j1=lfcl(lfcl+ic1)
lfcl(lfcl+ic1)=-1
c
c iflag=1 for fission
c iflag is used to determine if the law choosen was reworked in this
c patch of if it wasn't - please refer to the xsigen manual - the
c celr coding can be uncommented to see which laws are used that are
c not reworked - be warned though that the output file that those
c few lines of code creates could be rather large.
c
c
iflag=1
cdbg set iflag to 0 for single event collision sampling
cdbg iflag=0
call acecolx(2,iflag)
lfcl(lfcl+ic1)=j1
rtc(krtc+8,iex)=t1
celr
celr if(iflag.ne.-1)write(97,'(a)'//'fiss',lawelr
if(iflag.ne.-1)idum(lawelr)=idum(lawelr)+1
celr
if(iflag.eq.-1)then
do 147 j=1,idatr(lidr+im1)
jp=kxsc+idatr(lidr+im1+j)+it
do 146 i=1,ixsn
ii=ixsn+1-i
xscon(jp+ii-1)=xscon(jp+ii-1)+xsnt(lxnt+i)*w5
146 continue
147 continue
go to 170
endif
ni=cmult
do 162 i=1,ni
if(colout(1,i).lt.xsen(ixsn+1))go to 162
do 150 j=2,ixsn+1
150 if(colout(1,i).lt.xsen(ixsn+j))go to 155
go to 162
155 ig1=j-1
ig1=ixsn+1-ig1
do 160 j=1,idatr(lidr+im1)
jp=kxsc+idatr(lidr+im1+j)+it+ig1-1
160 xscon(jp)=xscon(jp)+w3
162 continue
go to 170
c
c cycle through all isotopes as directed by user
c
163 do 169 m=m1,jmd(ljmd+mk+1)-1
iex=lme(llme+1,m)
fisse=rtc(krtc+9,iex)*fme(lfme+m)
if(fisse.eq.0.)go to 169
c iflag=1 for fission
iflag=1

```

```

cdbg set iflag to 0 for single event collision sampling
cdbg iflag=0
w4=fisse*w2
rtc(krtc+2,iex)=rtc(krtc+1,iex)
ktc(kktc+2,iex)=ktc(kktc+1,iex)
tmpe=rtc(krtc+8,iex)
rtc(krtc+8,iex)=rtc(krtc+9,iex)
itmp=lfcl(lxfc+ic1)
lfcl(lxfc+ic1)=-1
call acecolx(2,iflag)
lfcl(lxfc+ic1)=itmp
rtc(krtc+8,iex)=tmpe
celr
celr      if(iflag.ne.-1)write(97,*)'fiss',lawelr
celr      if(iflag.ne.-1)idum(lawelr)=idum(lawelr)+1
celr      if(iflag.eq.-1)then
celr          w4=w4*rtc(krtc+10,iex)
celr          do 149 j=1,idatr(lidr+im1)
celr              jp=kxsc+idatr(lidr+im1+j)+it
celr              do 148 i=1,ixsn
celr                  ii=ixsn+1-i
celr                  xscon(jp+ii-1)=xscon(jp+ii-1)+xsnt(lxnt+i)*w1
celr              continue
celr          continue
celr          go to 170
celr      endif
celr      ni=cmult
celr      do 168 i=1,ni
celr          if(colout(1,i).lt.xsen(lxsn+1))go to 168
celr          do 164 j=2,ixsn+1
celr              if(colout(1,i).lt.xsen(lxsn+j))go to 165
celr              go to 168
celr          ig1=j-1
celr          ig1=ixsn+1-ig1
celr          do 166 j=1,idatr(lidr+im1)
celr              jp=kxsc+idatr(lidr+im1+j)+it+ig1-1
celr          xscon(jp)=xscon(jp)+w4
celr      continue
celr      continue
c
c      do scattering stuff
170 if(mxsc(7).eq.1)go to 182
c
c      pick an isotope to do the scattering with
m=m1
c      calculate scattering cross section and the value to be stored
w3=totm-sigma-sigf
w4=w3*w2
c
c      choose an isotope to do scattering with
if(npq(lnpq+mk).eq.1)go to 175
c=rang()*w3
do 173 m=m1,jmd(ljmd+mk+1)-2
c=c-(rtc(krtc+5,lme(llme+1,m))-rtc(krtc+9,lme(llme+1,m))-1
   rtc(krtc+3,lme(llme+1,m))*fme(lfme+m)
173 if(c.lt.0.)go to 175
175 iex=lme(llme+1,m)
iet=lmt(llmt+m)
if(ergolde.gt.esa(lesa+iet))iet=0
erg=ergolde
wgt=wgtolde
uuu=uold
vvv=vold
www=wold
call egout
go to 197
c
c      cycle through all isotopes in material mk
182 do 190 m=m1,jmd(ljmd+mk+1)-1
iex=lme(llme+1,m)
iet=lmt(llmt+m)
if(ergolde.gt.esa(lesa+iet))iet=0
tote=rtc(krtc+5,iex)*fme(lfme+m)
abse=rtc(krtc+3,iex)*fme(lfme+m)
fisse=rtc(krtc+9,iex)*fme(lfme+m)
w4=(tote-abse-fisse)
w4=w4*w2
erg=ergolde

```

```

wgt=wgtolde
uuu=uold
vvv=vold
www=wold
call egout
190 continue
c   reset everything before returning
197 if(mxsc(8).eq.0)then
  ranb=ranbelr
  rans=ranselr
  rnrtc=rnrtcelr
  endif
  wgt=wgtolde
  erg=ergolde
  uuu=uold
  vvv=vold
  www=wold
  ntyn=0
  return
c
c   store current information
c   photon groups not coded yet
200 if(ipt.eq.2) return
  if(erg.lt.xsen(lxsn+1).or.erg.gt.xsen(lxsn+ixsn+1))return
c   binary search method
  ky=1
  ky=ixsn+1
201 if((ky-kv).eq.1) go to 203
  ku=(kv+ky)/2
  if(erg.lt.xsen(lxsn+ku))go to 202
  kv=ku
  go to 201
202 ky=ku
  go to 201
203 ig=kv
c   groups are reversed
  ig=ixsn+1-ig
c   go to 230
c  215 if(erg.lt.xsep(lxsp+1))return
c   do 220 i=2,ixsp+1
c  220 if(erg.lt.xsep(lxsp+i))go to 225
c   return
c   photon groups are below neutron groups
c  225 ig=i-1
c   ig=ixsp+1-ig+ixsn
230 crcon(kcrc+im+ig-1)=crcon(kcrc+im+ig-1)+wgt
  return
c
c   flush container array
cdir$ ivdep
300 do 310 i=1,mxsd(3)
*if -def,cray,1   omit the check, in order for the loop to vectorize.
  if(xscon(kxsc+i).eq.0.)go to 310
  xsarr(kxsa+i)=xsarr(kxsa+i)+xscon(kxsc+i)
310 continue
  if(mxsc(6).ne.0)go to 365
  iq=kxsa+mxsd(3)
  do 320 i=1,nbinxs
    ip=kxsc+(i-1)*icsz
    do 315 j=1,mxsd(5)
      iq=iq+1
      ip=ip+1
      if(xscon(ip).eq.0.)go to 315
      xsarr(iq)=xsarr(iq)+xscon(ip)**2
315 continue
320 continue
c   do correlation data
c   it = position of first data for each bin within xscon
c   flux information
c   iu = position of fiss cross section within xscon
c   iv = position of first correlation data for each bin
c   within xsarr
c   iw = pos of current data within xscon
c   ix = pos of current data within xsarr
c
  it=kxsc
  iu=kxsc+mxsd(1)*2
  iv=kxsa+mxsd(3)+mxsd(5)*nbinxs
  do 360 i=1,nbinxs
    iw=iu

```

```

      ix=iv
c      doing fiss,abs,nu*fiss,total
      do 330 j=1,4
cdir$ ivdep
      do 325 k=1,mxsd(1)
      iw=iw+1
      ix=ix+1
*if -def,cray,1
      if(xscon(iw).eq.0.)go to 325
c      multiply reaction rate by flux for correlation info
      xsarr(ix)=xsarr(ix)+xscon(iw) *
      1 xscon(it+k)
325 continue
330 continue
c      doing scattering array
c      group from
      do 350 j=1,mxsd(1)
cdir$ ivdep
c      group to
      do 340 k=1,mxsd(1)
      is=isp0(lisp+1,(j-1)*mxsd(1)+k)
*if -def,cray,1
      if(is.eq.0)go to 340
      xsarr(ix+is)=xsarr(ix+is)+xscon(iw+is) *
      1 xscon(it+j)
340 continue
350 continue
      it=it+icsz
      iu=iu+icsz
      iv=iv+mxsd(4)
360 continue
c      zero out container array
cdir$ ivdep
      do 370 i=1,mxsd(3)
370 xscon(kxsc+i)=0.
c      do current array now
      if(ixsf.ne.12)return
cdir$ ivdep
      do 380 i=1,mxsd(6)
*if -def,cray,1    omit the check, in order for the loop to vectorize.
      if(crcon(kcrc+i).eq.0.)go to 380
      crarr(kcra+i)=crarr(kcra+i)+crcon(kcrc+i)
380 continue
cdir$ ivdep
      if(mxsc(6).eq.0)then
      do 385 i=1,mxsd(6)
*if -def,cray,1    omit the check, in order for the loop to vectorize.
      if(crcon(kcrc+i).eq.0.)go to 385
      crarr(kcra+mxsd(6)+i)=crarr(kcra+mxsd(6)+i) +
      1 crcon(kcrc+i)**2
385 continue
      endif
cdir$ ivdep
      do 390 i=1,mxsd(6)
390 crcon(kcrc+i)=0.
      return
      end
*deck ego
      subroutine egout
c
c      this subroutine calculates the angle and the outgoing energy
c      for scattering and stores the appropriate values
c
c      w4=scattering contribution
c      iml=position within idatr
c      ig=start group
*call cm
*call cmx
c
      dimension vr(3)
      ifltg=0
c      sample target velocity for the scattering treatment
      e2=erg
      r2=rtc(krtc+1,iex)
      rtc(krtc+2,iex)=rtc(krtc+1,iex)
      k2=ktc(kktc+1,iex)
      ktc(kktc+2,iex)=ktc(kktc+1,iex)
      ssr=0.
      if(iet.eq.0.and.(awm(lawn+iex).le.1.or.erg.le.400.*ttn))
      1      call tgtvelx(1,e2,r2,ar,vr,k2)

```

```

        if(iet.ne.0)go to 250
        erg=e2
        rtc(krtc+2,iex)=r2
        ktc(kktc+2,iex)=k2
c      must set fission to capture temporarily
        j1=lfcl(lfcl+ic1)
        lfcl(lfcl+ic1)=-1
        iflag=2
cdbg  set iflag to 0 for single event collision sampling
cdbg  iflag=0
      call acecolx(0,iflag)
      lfcl(lfcl+ic1)=j1
celr
celr      if(iflag.ne.-2)write(97,'(scat',lawelr
celr      if(iflag.ne.-2)idum(lawelr)=idum(lawelr)+1
celr      if(iflag.eq.-2)go to 260
c      do normal scattering and finish target velocity if necessary
      if(ifltg.eq.1)call tgtvelx(2,e2,r2,ar,vr,k2)
      wgt=wgt
      ni=cmult
      do 240 k=1,ni
      if(ssr.ne.0.)go to 200
      cosval=colout(2,k)
      erg=colout(1,k)
      call scatst(cosval)
      go to 240
200  s=1./ssr
      vr(1)=s*vr(1)
      vr(2)=s*vr(2)
      vr(3)=s*vr(3)
      call rotasz(colout(2,k),vr,uuu,irt)
      s=sqrt(colout(1,k)*ar)
      uuu=s*uuu+vtr(1)
      vvv=s*vvv+vtr(2)
      www=s*www+vtr(3)
      t=uuu**2+vvv**2+www**2
      s=1./sqrt(t)
      uuu=uuu*s
      vvv=vvv*s
      www=www*s
      erg=t/ar
      cosval=uuu*uold+vvv*vold+www*wold
      call scatst(cosval)
240  continue
      go to 260
250  iflag=3
cdbg  set iflag to 0 for single event collision sampling
cdbg  iflag=0
      call sabcolx(iflag)
      if(iflag.eq.-3)return
      cosval=uuu*uold+vvv*vold+www*wold
      call scatst(cosval)
260  return
      end
*deck sca
      subroutine scatst(cosval)
c
c      this subroutines stores the scattering stuff
c      w4=scattering contribution
c      iml=position within idatr
c      ig=start group
c      igr=group to
c
*call cm
*call cmx
c
      dimension pv(0:ileg)
c
c      photon not coded yet
100  if(ipt.eq.2)return
      if(erg.lt.xsen(lxsn+1).or.erg.gt.xsen(lxsn+ixsn+1))return
c      binary search method
      kv=1
      ky=ixsn+1
101  if((ky-kv).eq.1) go to 103
      ku=(kv+ky)/2
      if(erg.lt.xsen(lxsn+ku))go to 102
      kv=ku
      go to 101

```

```

102  ky=ku
     go to 101
103  ig1=kv
     ig1=ixsn+1-ig1
c      go to 130
c  115 if(erg.lt.xsep(lxsp+1))return
c      do 120 i=2,ixsp+1
c  120 if(erg.gt.xsep(lxsp+i))go to 125
c      return
c      photon groups are below neutron groups
c  125 ig1=i-1
c      ig1=ixsp+1-ig1+ixsn
130  it=mxsd(1)
c      ig=group from, ig1=group to
ip=ispo(lisp+1,(ig-1)*it+ig1)
if(ip.eq.0)then
write(iuo,131)ig,ig1
write(jtty,131)ig,ig1
return
131  format('XSERR: scattering occurred from grp ',i5,' to ',i5,
1 ' but not stored')
endif
ip=it*6+ip
iq=it*6+ispo(lisp+2,(ig-1)*it+ig1)
if(mxsc(5).eq.1)go to 270
c      determine which of msct bins cosval belongs to
c      do angular distribution data
c      mu component
c      note:the bin boundaries for the equi cosine bins will go
c      from -1 to 1
cosv=cosval+1.0d0
k=cosv*bdiv1+1
if(k.gt.msct)then
     write(iuo,*)'WARNING - k > msct'
endif
iq=iq+k-1
go to 280
c      do the legendre order stuff
c      Pi, i goes from 1 to ipn
270  pv(0)=one
pv(1)=cosval
do 275 k=1,mxsc(1)-1
pv(k+1)=((2.0d0*k+one)*cosval*pv(k)-k*pv(k-1))/(k+one)
275 continue
c      cycle through all tally bins that contain this cell
280  do 295 i=1,idatr(lidr+im1)
ii=kxsc+idatr(lidr+im1+i)
c      the -1 is required because idatr starts at 1 and ip is rel to 0
jp=ii+ip-1
xscon(jp)=xscon(jp)+w4
if(mxsc(5).eq.1)go to 285
jp=ii+iq-1
xscon(jp)=xscon(jp)+w4
go to 295
285  do 290 j=1,mxsc(1)
jp=jp+mxsd(2)
290  xscon(jp)=xscon(jp)+w4*pv(j)
295 continue
      return
end
*deck xspl
      subroutine xsoutp
c
c      this subroutine dumps the newly generated multigroup xs
c
*call cm
c
      character hm*8,hm1*8,a(6)*12,a12*12,hzaid*10,dir*80,a21(3)*21
dimension func(nbin+1),e(6),y(6),pv(ileg,nbin+1),nx1(16)
dimension jx1(32),nl(13)
common/plvs/v(ileg+1)
data a12/'          '
data nl/1,2,3,4,5,7,9,11,22,24,44,66,67/
c
c      sfrac = fraction of total cross section below which
c      scattering cross sections will be set to 0
      sfrac=1.0d-06
c
400 format(a10,f10.6,13x,a10,////)
401 format(8i9)

```

```

402 format(4(1pe20.13))
403 format(a80)
404 format(a6.5x,f10.6)
405 format(3(a21,2x))
406 format(a21)
c
      do 5 i=1,6
 5 a(i)=a12
c      calculate the tally normalization factor.
c      return if kcode problem is not settled
      if(nsr.eq.71.and.kcy.le.ikz) return
      t=max(1,nps)
      if(nsr.eq.6.and.nrrs.ge.nrss)t=max(1,np1)
      if(nsr.eq.6.and.nrrs.lt.nrss)t=max(1,npsr)
      if(nsr.ne.71)go to 10
      t=n3rck*(kc2+1-ikz)-wt0*nsa
      if\knrm.ne.0)t=nps-nskk
10 fp=1./max(t,one)
c
c      initialize func and pv and fnc array
c      func is the array containing the boundaries for equi-cosine bins
c      pv array contains the integral of Pl(mu) evaluated on bin
c      boundaries
      if(mxsc(1).eq.0)go to 19
      func(1)=-1.0d0
      do 11 i=1,msct
      func(i+1)=func(i)+bdiv
      do 13 i=1,msct+1
      v(1)=func(i)
      do 12 k=2,mxsc(1)+1
      v(k)=v(k-1)*func(i)
      pv(k-1,i)=pvint(k-1)
12     continue
13     continue
      do 15 i=1,msct
      do 14 k=1,mxsc(1)
      pv(k,i)=pv(k,i+1)-pv(k,i)
14     continue
15     continue
c
19 itpos=9
c      data storage positions
c      1      2      3      4      5      6      7      8      9
c      flux chi totcat totpl diff fiss abs nuf tot
c      scan for number of upscatter and downscatter groups
      iupscat=0
      idnscat=0
      it=mxsd(1)*6
      do 35 i=1,nbinxs
c      j is group from, k is group to
      do 30 j=1,mxsd(1)
      do 25 k=1,mxsd(1)
      is=ispo(lisp+1,(j-1)*mxsd(1)+k)
      if(is.eq.0)go to 25
      if(xsarr(lxsa+it+is).eq.0.0)go to 25
c      upscatter
      if(j.le.k)go to 20
      if(iupscat.lt.(j-k))iupscat=j-k
c      downscatter
20 if(j.ge.k)go to 25
      if(idnscat.lt.(k-j))idnscat=k-j
25 continue
30 continue
      it=it+icsz
35 continue
      ispos=itpos+iupscat+1
      ilen=ispos+idnscat
c
c      open cross section output files
      hm=xsout
      call unique(hm,jtty)
      open(iuxo,file=hm,status='new')
      if(mxsc(6).eq.0)then
      hm1=xsrel
      call unique(hm1,jtty)
      open(iuxr,file=hm1,status='new')
      endif
c
c      write a listing of what each bin is and the various information
c      needed - this is written to the output file

```

```

write(iuo,300)
write(iuo,302)hm
if(mxsc(6).eq.0)write(iuo,303)hm1
write(iuo,320)itpos,ispos,ilen,mxsd(1),mxsc(1)
write(iuo,304)
do 41 i=1,nbinxs
it=lcexs(lcce+i)
is=idatc(lidc+it)
iu=1
40 iv=is
if(iv.gt.(iu+14))iv=iu+14
if(iu.eq.1)write(iuo,310)i,(idatc(lidc+it+1+j),j=iu,iv)
if(iu.ne.1)write(iuo,311)(idatc(lidc+it+1+j),j=iu,iv)
iu=iv+1
if(iu.le.is)go to 40
41 continue

do 36 i=1,13
if(idum(i).ne.0)go to 37
36 continue
go to 39
37 write(iuo,305)
do 38 i=1,13
if(idum(i).ne.0)write(iuo,306)nl(i),idum(i)
38 continue
39 continue
c
300 format('Cross Section Generation Notes')
302 format('Cross section file = ',a8)
303 format('Relative error file = ',a8)
304 format('Bin - Corresponding cells')
305 format('Scat. Law not mod - num. hits')
306 format(i2,15x,i12)
310 format(i5,3x,15(i5,1x))
311 format(8x,15(i5,1x))
320 format('DTF file info : itpos = ',i5,' ispos = ',i5,
1 ' ilen = ',i5,' ngroup = ',i5,' leg order = ',i5)
800 format(' P',i1,' bin=',i5,'c itpos=',i5,' ispos=',i5,
1 ' ilen=',i5,' ngroup=',i5,' leg order=',i5)
c
c      output data
c      cycle through all bins
do 180 i=1,nbinxs
write(iuxo,800)0,i,itpos,ispos,ilen,mxsd(1),mxsc(1)
if(mxsc(6).eq.0)then
write(iuxr,800)0,i,itpos,ispos,ilen,mxsd(1),mxsc(1)
endif
c      xscon array will be used for temporary storage
c      flush 1st bin of array
do 42 j=1,icsz
42 xscon(lxsc+j)=0.0
c
c      calculate total p0 cross section and store in xscon
c      j1=start of flux data
c      j2=start of scat data
j1=lxsa+icsz*(i-1)
j2=j1+mxsd(1)*6
do 44 ifr=1,mxsd(1)
sum=0.0
do 43 ito=1,mxsd(1)
is=isp0(lisp+1,(ifr-1)*mxsd(1)+ito)
if(is.eq.0)go to 43
is=j2+is
sum=sum+xsarr(is)
43 continue
c      xsarr(j1+ifr) is the flux
if(xsarr(j1+ifr).ne.0.)xscon(lxsc+ifr)=sum/xsarr(j1+ifr)
44 continue
c      calc legendre components and store in xscon
c      - also store total p1 xs in xscon
c
if(mxsc(1).eq.0)go to 53
c
if(mxsc(5).eq.0)then
c      bin method
c      j1=start of flux data
c      j2=start of scat data array
c      j3=start of tot p1
c      j4=start of p1 data array

```

```

j1=lxsa+icsz*(i-1)
j2=j1+mxsd(1)*6
j3=lxsc+mxsd(1)
j4=j3+mxsd(1)
c      cycle through legendre bins
do 49 ip=1,mxsc(1)
c      cycle through groups
do 48 ifr=1,mxsd(1)
do 47 ito=1,mxsd(1)
is=isp0(lisp+1,(ifr-1)*mxsd(1)+ito)
if(is.eq.0)go to 47
it=isp0(lisp+2,(ifr-1)*mxsd(1)+ito)
if(xsarr(j2+is).eq.0.)go to 47
area=0.0
do 45 ic=1,msct
45 area=area+xsarr(j2+it+ic-1)
if (area.eq.0.) go to 47
area=1.0/area
tmpe=0.0
cdbg
cdbg      write(97,'(a4,i6,a5,i6,a4,i6)')'bin ',i,'from ',ifr,' to ',ito
cdbg      elr=-1.0d0
cdbg
do 46 ic=1,msct
cdbg
cdbg      write(97,'(1pe12.5,1x,1pe12.5)')elr,xsarr(j2+it+ic-1)
cdbg      1 *area*bdiv1
cdbg      elr=elr+bdiv
cdbg
c      xsarr(j2+is) is the P0 cross section
c      xsarr(j1+ifr) is the flux
46 tmpe=tmpe+pv(ip,ic)*xsarr(j2+it+ic-1)*area*bdiv1*xsarr(j2+is)/
1 xsarr(j1+ifr)
cdbg
cdbg      write(97,'(1pe12.5)')1.0
cdbg
xscon(j4+is)=tmpe
if(ip.eq.1)xscon(j3+ifr)=xscon(j3+ifr)+tmpe
47 continue
48 continue
j4=j4+mxsd(2)
49 continue
c
else
c
j1=start of flux data
c
j2=start of pl data array
c
j3=start of tot p0
c
j4=start of tot p1
c
j5=start of pl data array
j1=lxsa+icsz*(i-1)
j2=j1+mxsd(1)*6+mxsd(2)
j3=lxsc
j4=j3+mxsd(1)
j5=j4+mxsd(1)
c      cycle through legendre bins
do 52 ip=1,mxsc(1)
c      cycle through groups
do 51 ifr=1,mxsd(1)
do 50 ito=1,mxsd(1)
is=isp0(lisp+1,(ifr-1)*mxsd(1)+ito)
if(is.eq.0)go to 50
c
pl-xs=pl value / flux
if(xsarr(j1+ifr).ne.0.) xscon(j5+is)=xsarr(j2+is)/xsarr(j1+ifr)
if(ip.eq.1)xscon(j4+ifr)=xscon(j4+ifr)+xscon(j5+is)
50 continue
51 continue
j2=j2+mxsd(2)
j5=j5+mxsd(2)
52 continue
c
endif
c
sum up chis for norm
53 sum=0.0
is=lxsa+mxsd(1)+icsz*(i-1)
do 54 j=1,ixsn
54 sum=sum+xsarr(is+j)
if(sum.eq.0.)sum=1.
sum=1.0/sum

```

```

k1=0
c
c   cycle through all groups and store data
c
do 90 j=1,mxsd(1)
c
j1=position of flux
c   j2=position of data
c   j3=position within square data
c   j4=position within correlation data
c   j5=position of sqaure flux
j1=lxsa+icsz*(i-1)+j
j2=j1
j3=lxsa+mxsd(3)+mxsd(5)*(i-1)+j
j4=lxsa+mxsd(3)+mxsd(5)*nbinsx+mxsd(4)*(i-1)+j-mxsd(1)
j5=j3
k1=k1+1
if(k1.eq.7)call xsfl(k1,e,y)
c
flux value
c
volume normalization is not performed
if(xsarr(j1).eq.0.)go to 55
e(k1)=xsarr(j1)*fp
y(k1)=1.0
if(mxsc(6).ne.0.)go to 56
tmpe=xsarr(j5)/(xsarr(j1)**2)-fp
if(tmpe.gt.0.)y(k1)=sqrt(tmpe)
go to 56
55 e(k1)=0.0
y(k1)=0.0
c
chi value
c
no need to multiply by fp since values range betwen 0-1
56 j2=j2+mxsd(1)
j3=j3+mxsd(1)
k1=k1+1
if(k1.eq.7)call xsfl(k1,e,y)
if(xsarr(j2).eq.0.)go to 57
e(k1)=xsarr(j2)*sum
y(k1)=1.0
if(mxsc(6).ne.0.)go to 58
tmpe=xsarr(j3)/(xsarr(j2)**2)-fp
if(tmpe.gt.0.)y(k1)=sqrt(tmpe)
go to 58
57 e(k1)=0.0
y(k1)=0.0
c
tot-p0, tot-p1, diffusion coef
c
58 k1=k1+1
if(k1.eq.7)call xsfl(k1,e,y)
e(k1)=xscon(lxsc+j)
y(k1)=0.0
k1=k1+1
if(k1.eq.7)call xsfl(k1,e,y)
e(k1)=xscon(lxsc+j+mxsd(1))
y(k1)=0.0
k1=k1+1
if(k1.eq.7)call xsfl(k1,e,y)
sigtr=0.
if(xsarr(j1).ne.0.0)sigtr=xsarr(j2+mxsd(1)*4)/xsarr(j1)-
$      xscon(lxsc+j+mxsd(1))
if(sigtr.ne.0.0)sigtr=1./sigtr
e(k1)=third*sigtr
y(k1')=0.0
c
fiss,abs,nu*fiss,tot
c
do 60 i1=1,4
j2=j2+mxsd(1)
j3=j3+mxsd(1)
j4=j4+mxsd(1)
k1=k1+1
if(k1.eq.7)call xsfl(k1,e,y)
if(xsarr(j2).eq.0.)go to 59
e(k1)=xsarr(j2)/xsarr(j1)
y(k1)=1.0
if(mxsc(6).ne.0.)go to 60

```

```

tmpe=xsarr(j3)/(xsarr(j2)**2)
tmpe=tmpe+xsarr(j5)/(xsarr(j1)**2)
tmpe=tmpe-2.*xsarr(j4)/(xsarr(j2)*xsarr(j1))
if(tmpe.gt.0.)y(k1)=sqrt(tmpe)
go to 60
59 e(k1)=0.0
y(k1)=0.0
60 continue
c
c scattering array
c
c the scattering data in elrarr appears in the following manner
c the group to group numbers are listed
c 1->1 1->2 1->3 2->1 2->2 2->3 3->1 3->2 3->3
c
c the scattering data on the dtf file has to appear in the
c following order for three group example assuming full upscatter
c grp 1) 3->1 2->1 1->1 0.0 0.0
c grp 2) 0.0 3->2 2->2 1->2 0.0
c grp 3) 0.0 0.0 3->3 2->3 1->3
c
c j2=start of scat data
c j3=start of scat square data
c j4=start of scat correlation data
c
j2=lxsa+icsz*(i-1)+mxsd(1)*6
j3=lxsa+mxsd(3)+mxsd(5)*(i-1)+mxsd(1)*6
j4=lxsa+mxsd(3)+mxsd(5)*nbinxs+mxsd(4)*(i-1)+mxsd(1)*4
c do upscatter first
if(iupscat.eq.0)go to 75
do 70 k=iupscat,1,-1
ifr=k+j
if(ifr.gt.mxsd(1))go to 65
ito=j
ipos=ispo(lisp+1,(ifr-1)*mxsd(1)+ito)
k1=k1+1
if(k1.eq.7)call xsfl(k1,e,y)
if(ipos.eq.0)go to 61
ipol=ipos+j3
ipo2=ipos+j4
ipos=ipos+j2
iflx=j1+k
ifll=j5+k
if(xsarr(ipos).eq.0.)go to 61
e(k1)=xsarr(ipos)/xsarr(iflx)
if(e(k1).lt.xscon(lxsc+ifr)*sfrac)then
je=lxsc+mxsd(1)*2
do 62 ie=1,mxsc(1)
xscon(je+ipos-j2)=0.0
je=je+mxsd(2)
go to 61
endif
y(k1)=1.0
if(mxsc(6).ne.0)go to 70
tmpe=xsarr(ipol)/(xsarr(ipos)**2)
tmpe=tmpe+xsarr(ifll)/(xsarr(iflx)**2)
tmpe=tmpe-2.*xsarr(ipo2)/(xsarr(ipos)*xsarr(iflx))
if(tmpe.gt.0.)y(k1)=sqrt(tmpe)
go to 70
61 e(k1)=0.0
y(k1)=0.0
go to 70
65 k1=k1+1
if(k1.eq.7)call xsfl(k1,e,y)
e(k1)=0.0
y(k1)=0.0
70 continue
c do self scatter
75 ifr=j
ito=j
ipos=ispo(lisp+1,(ifr-1)*mxsd(1)+ito)
ipol=ipos+j3
ipo2=ipos+j4
ipos=ipos+j2
iflx=j1
ifll=j5
k1=k1+1
if(k1.eq.7)call xsfl(k1,e,y)
if(xsarr(ipos).eq.0.)go to 77
e(k1)=xsarr(ipos)/xsarr(iflx)

```

```

        if(e(k1).lt.xscon(lxsc+ifr)*sfrac)then
          je=lxsc+mxsd(1)*2
          do 63 ie=1,mxsc(1)
            xscon(je+ipos-j2)=0.0
63      je=je+mxsd(2)
          go to 77
        endif
        y(k1)=1.0
        if(mxsc(6).ne.0)go to 78
        tmpe=xsarr(ipol)/(xsarr(ipos)**2)
        tmpe=tmpe+xsarr(ifl1)/(xsarr(iflx)**2)
        tmpe=tmpe-2.*xsarr(ipo2)/(xsarr(ipos)*xsarr(iflx))
        if(tmpe.gt.0.)y(k1)=sqrt(tmpe)
        go to 78
77    e(k1)=0.0
        y(k1)=0.0
c      do down scatter
78    do 85 k=1,idnscat
        if(ifr.lt.1)go to 80
        ito=j
        ipos=ipso(lisp+1,(ifr-1)*mxsd(1)+ito)
        ipo1=ipos+j3
        ipo2=ipos+j4
        ipos=ipos+j2
        iflx=j1-k
        ifll=j5-k
        k1=k1+1
        if(k1.eq.7)call xsfl(k1,e,y)
        if(xsarr(ipos).eq.0.)go to 79
        e(k1)=xsarr(ipos)/xsarr(iflx)
        if(e(k1).lt.xscon(lxsc+ifr)*sfrac)then
          je=lxsc+mxsd(1)*2
          do 64 ie=1,mxsc(1)
            xscon(je+ipos-j2)=0.0
64      je=je+mxsd(2)
          go to 79
        endif
        y(k1)=1.0
        if(mxsc(6).ne.0)go to 85
        tmpe=xsarr(ipol)/(xsarr(ipos)**2)
        tmpe=tmpe+xsarr(ifl1)/(xsarr(iflx)**2)
        tmpe=tmpe-2.*xsarr(ipo2)/(xsarr(ipos)*xsarr(iflx))
        if(tmpe.gt.0.)y(k1)=sqrt(tmpe)
        go to 85
79    e(k1)=0.0
        y(k1)=0.0
        go to 85
80    k1=k1+1
        if(k1.eq.7)call xsfl(k1,e,y)
        e(k1)=0.0
        y(k1)=0.0
85    continue
90    continue
        call xsfl(k1,e,y)
c      do the p1 components
c
        if(mxsc(1).eq.0)go to 180
c      j1=start of data within xscon array
        j1=lxsc+mxsd(1)*2
c
        do 170 i1=1,mxsc(1)
          write(iuxo,800)i1,i,itpos,ispos,ilen,mxsd(1),mxsc(1)
          if(mxsc(6).eq.0)then
            write(iuxr,800)i1,i,itpos,ispos,ilen,mxsd(1),mxsc(1)
          endif
c      cycle through all groups
        k1=0
c
        do 160 j=1,mxsd(1)
          do 95 k=1,itpos
            k1=k1+1
            if(k1.eq.7)call xsfl(k1,e,y)
            e(k1)=0.
            y(k1)=0.
95      continue
c      scattering array

```

```

c      do upscatter first
      if(iupscat.eq.0)go to 115
      do 110 k=iupscat,1,-1
      ifr=k+j
      if(ifr.gt.mxsd(1))go to 105
      ito=j
      ipos=isp0(lisp+1,(ifr-1)*mxsd(1)+ito)
      k1=k1+1
      if(k1.eq.7)call xsfl(k1,e,y)
      if(ipos.eq.0)go to 101
      ipos=j1+ipos
      if(xscon(ipos).eq.0.)go to 101
      e(k1)=xscon(ipos)
      go to 102
101  e(k1)=0.0
102  y(k1)=0.0
      go to 110
105  k1=k1+1
      if(k1.eq.7)call xsfl(k1,e,y)
      e(k1)=0.0
      y(k1)=0.0
110  continue
c      do self scatter
115  ifr=j
      ito=j
      ipos=j1+isp0(lisp+1,(ifr-1)*mxsd(1)+ito)
      k1=k1+1
      if(k1.eq.7)call xsfl(k1,e,y)
      if(xscon(ipos).eq.0.)go to 121
      e(k1)=xscon(ipos)
      go to 122
121  e(k1)=0.0
122  y(k1)=0.0
c      do down scatter
      do 135 k=1,idnscat
      ifr=j-k
      if(ifr.lt.1)go to 130
      ito=j
      ipos=j1+isp0(lisp+1,(ifr-1)*mxsd(1)+ito)
      k1=k1+1
      if(k1.eq.7)call xsfl(k1,e,y)
      if(xscon(ipos).eq.0.)go to 126
      e(k1)=xscon(ipos)
      go to 127
126  e(k1)=0.0
127  y(k1)=0.0
      go to 135
130  k1=k1+1
      if(k1.eq.7)call xsfl(k1,e,y)
      e(k1)=0.0
      y(k1)=0.0
135  continue
c      160 continue
      call xsfl(k1,e,y)
c      j1=j1+mxsd(2)
170  continue
c      180 continue
      close(iuxo)
      close(iuxr)
c      flush container array again as a precaution
      do 181 i=1,icsz
181  xscon(lxsc+i)=0.0
c      do current output
      if(ixsf.ne.12)go to 220
      hm=xscur
      call unique(hm,jtty)
      open(iuxc,file=hm,status='new')
      write(iuxc,'(a18,i6)')'number of groups =',mxsd(1)
      open(iul,form='formatted',status='scratch')
      open(iu2,form='formatted',status='scratch')
      open(iu3,form='formatted',status='scratch')
      k1=0
      do 190 i=1,mxa
      if(lcrxs(llcr+i).eq.0)go to 190
      ip=lcrxs(llcr+i)

```

```

ifr=nc1(lncl+idacr(iacr+ip))
id=idacr(lacr+ip+1)
do 185 j=1,id
ito=nc1(lncl+idacr(lacr+ip+1+j))
k1=k1+1
if(k1.eq.7)call xsflc(1,k1,e,y,a)
write(a(k1),'(i5,a1,i5)')ifr,'>',ito
185 continue
190 continue
call xsflc(1,k1,e,y,a)
195 continue
k1=0
do 210 i=1,mxa
if(lcrxs(l1cr+i).eq.0)go to 210
ip=lcrxs(l1cr+i)
id=idacr(lacr+ip+1)
do 205 j=1,id
iq=idacr(lacr+ip+1+j+id)
do 200 k=1,mxsd(1)
k1=k1+1
if(k1.eq.7)call xsflc(2,k1,e,y,a)
if(crarr(lcra+iq+k-1).eq.0.0)go to 196
e(k1)=crarr(lcra+iq+k-1)*fp
y(k1)=1.0
tmpe=crarr(lcra+iq+k-1+mxsd(6))/(crarr(lcra+iq+k-1)**2)-fp
if(tmpe.gt.0.)y(k1)=sqrt(tmpe)
go to 200
196 e(k1)=0.0
y(k1)=0.0
200 continue
205 continue
210 continue
call xsflc(2,k1,e,y,a)
c
call xsflc(3,k1,e,y,a)
c
c      write out type 1 file if requested
c      can only do if binning method requested
220 if(mxsc(12).eq.1.or.mxsc(5).ne.0.or.mxsc(1).eq.0)return
c      open type 1 output files
hm=xst1m
call unique(hm,jtty)
open(iuxt,file=hm,status='new')
hml=xst1m
call unique(hml,jtty)
open(iuxx,file=hml,status='new')
c
open(iu1,form='formatted',status='scratch')
open(iu2,form='formatted',status='scratch')
c
nrec=1
do 286 i=1,nbinxs
do 222 j=1,16
222 nx1(j)=0
do 224 j=1,32
224 jx1(j)=0
c      will use xscon for temp storage of xss array
do 226 j=1,mxsd(3)
226 xscon(lxsc+j)=0.0
c
j1=lxsa+icsz*(i-1)
c
nx1(2)=100000+i
nx1(3)=33
nx1(4)=4
c      assuming no photon groups in calc
nx1(5)=mxsd(1)
nx1(6)=iupscat
nx1(7)=idnscat
nx1(8)=0
nx1(9)=0
nx1(10)=1
nx1(11)=0
nx1(12)=1
jx1(1)=1
c      energy center and widths
do 228 j=1,mxsd(1)
div=xsen(lxsn+mxsd(1)+2-j)-xsen(lxsn+mxsd(1)+1-j)
xscon(lxsc+jx1(1)+j-1)=xsen(lxsn+mxsd(1)+1-j)+div/2.0
xscon(lxsc+jx1(1)+mxsd(1)+j-1)=div

```

```

228 continue
c      tot
jx1(2)=jx1(1)+mxsd(1)*2
j2=j1+mxsd(1)*5
do 230 j=1,mxsd(1)
230 if(xsarr(j1+j).ne.0.)xscon(lxsc+jx1(2)+j-1)=xsarr(j2+j)/
1 xsarr(j1+j)
c      fission xs
c      check for existence
j2=j1+mxsd(1)*2
do 232 j=1,mxsd(1)
232 if(xsarr(j2+j).ne.0.)go to 234
jx1(3)=0
jx1(4)=0
jx1(5)=0
jx1(6)=jx1(2)+mxsd(1)
go to 244
c
234 jx1(3)=jx1(2)+mxsd(1)
do 236 j=1,mxsd(1)
236 if(xsarr(j1+j).ne.0.)xscon(lxsc+jx1(3)+j-1)=xsarr(j2+j)/
1 xsarr(j1+j)
c      nu-bar
jx1(4)=jx1(3)+mxsd(1)
j3=j1+mxsd(1)*4
do 238 j=1,mxsd(1)
238 if(xsarr(j2+j).ne.0.)xscon(lxsc+jx1(4)+j-1)=xsarr(j3+j)/
1 xsarr(j2+j)
c      chi
jx1(5)=jx1(4)+mxsd(1)
j2=j1+mxsd(1)
sum=0.0
do 240 j=1,mxsd(1)
240 sum=sum+xsarr(j2+j)
sum=1.0/sum
do 242 j=1,mxsd(1)
242 xscon(lxsc+jx1(5)+j-1)=xsarr(j2+j)*sum
c      absorption
jx1(6)=jx1(5)+mxsd(1)
244 j2=j1+mxsd(1)*3
do 246 j=1,mxsd(1)
246 if(xsarr(j1+j).ne.0.)xscon(lxsc+jx1(6)+j-1)=xsarr(j2+j)/
1 xsarr(j1+j)
c
jx1(7)=0
jx1(8)=0
jx1(9)=jx1(6)+mxsd(1)
c      edits are sum-P0, fissa, abs, tot
xscon(lxsc+jx1(9))=2
xscon(lxsc+jx1(9)+1)=18
xscon(lxsc+jx1(9)+2)=102
xscon(lxsc+jx1(9)+3)=1
jx1(10)=jx1(9)+4
c      sum P0
do 250 j=1,mxsd(1)
sum=0.0
do 248 k=1,mxsd(1)
is=isp0(lisp+1,(j-1)*mxsd(1)+k)
if(is.eq.0)go to 248
j2=j1+mxsd(1)*6+is
sum=sum+xsarr(j2)
248 continue
if(xsarr(j1+j).ne.0.)xscon(lxsc+jx1(10)+j-1)=sum/xsarr(j1+j)
250 continue
if(jx1(3).eq.0) go to 254
do 252 j=1,mxsd(1)
252 xscon(lxsc+jx1(10)+mxsd(1)+j-1)=xscon(lxsc+jx1(3)+j-1)
254 do 256 j=1,mxsd(1)
256 xscon(lxsc+jx1(10)+mxsd(1)*2+j-1)=xscon(lxsc+jx1(6)+j-1)
do 258 j=1,mxsd(1)
258 xscon(lxsc+jx1(10)+mxsd(1)*3+j-1)=xscon(lxsc+jx1(2)+j-1)
c
jx1(11)=0
jx1(12)=0
jx1(13)=jx1(10)+mxsd(1)*4
xscon(lxsc+jx1(13))=jx1(13)+1
j2=jx1(13)
c      j3 is used for position of scattering data
j3=j1+mxsd(1)*6
do 262 ifr=1,mxsd(1)

```

```

i1=max(1,ifr-nx1(6))
i2=min(nx1(5),ifr+nx1(7))
do 260 ito=i1,i2
j2=j2+1
i3=isp0(lisp+1,mxsd(1)*(ifr-1)+ito)
if(is.eq.0)go to 260
if(xsarr(j1+ifr).ne.0.)xscon(lxsc+j2)=xsarr(j3+is)/
1 xsarr(j1+ifr)
260 continue
262 continue
lg=j2-jx1(13)
jx1(14)=0
jx1(15)=0
jx1(16)=j2+1
xscon(lxsc+jx1(16))=jx1(16)+1
jx1(17)=jx1(16)+lg+1
xscon(lxsc+jx1(17))=jx1(17)+1
c j2 is used for pos of scat data
c j3 is the loc within xpn block
c j4 is the loc within pn block
c j5=lpnd variable
c j6=final count of length of xss array
c
j2=j1+mxsd(1)*6
j3=jx1(16)+1
j4=jx1(17)+1
j5=1
j6=jx1(17)
c
do 282 ifr=1,mxsd(1)
i1=max(1,ifr-nx1(6))
i2=min(nx1(5),ifr+nx1(7))
do 280 ito=i1,i2
is=isp0(lisp+1,mxsd(1)*(ifr-1)+ito)
c
if(is.eq.0)then
xscon(lxsc+j3)=-1.0
j3=j3+1
go to 280
endif
if (xsarr(j2+is).eq.0.)then
xscon(lxsc+j3)=-1.0
j3=j3+1
go to 280
endif
c
store equi-probable cosines
c
calculate normalization area first
is=isp0(lisp+2,mxsd(1)*(ifr-1)+ito)+j2
area=0.0
do 264 j=1,msct
264 area=area+xsarr(is+j-1)
c area=0.0 means isotropic scattering
if(area.eq.0.0)then
xscon(lxsc+j3)=0.0
j3=j3+1
go to 280
endif
area=1.0d0/area
xscon(lxsc+j3)=j5
j3=j3+1
c
find first bin with data
sa = start cosine of equi-prob bin
so = end cosine of equi-prob bin
ib = current bin to start with
ic = count of equi-probable bin
ic=1
do 266 j=1,msct
xscon(lxsc+j4+j5-1)=func(j)
ib=j
if (xsarr(is+j-1).ne.0.)go to 268
266 continue
268 av=0.03125d0
avt1=0.0d0
do 278 j=1,32
if(j.eq.32)go to 274
if(avt1.eq.av)go to 272
if(avt1.gt.av) go to 273

```

```

      do 269 k=ib,msct
         avt=xsarr(is+k-1)*area
         if((avt1+avt).eq.av)go to 270
         if((avt1+avt).gt.av)go to 271
         avt1=avt1+avt
269   continue
270   xscon(lxsc+j4+j5-1+j)=func(k+1)
      ib=k+1
      avt1=0.0
      go to 278
271   r0=av-avt1
      r1=xsarr(is+k-1)*area
      xscon(lxsc+j4+j5-1+j)=(r0/r1)*bdiv+func(k)
      avt1=avt-r0
      ib=k+1
      go to 278
272   xscon(lxsc+j4+j5-1+j)=func(ib)
      avt1=0.0
      go to 278
273   r1=xsarr(is+ib-1-1)*area
      xscon(lxsc+j4+j5-1+j)=(av/r1)*bdiv+xscon(lxsc+j4+j5-1+j-1)
      avt1=avt1-av
      go to 278
274   if(ib.gt.msct)xscon(lxsc+j4+j5-1+j)=func(msct+1)
      if(ib.le.msct)then
         xscon(lxsc+j4+j5-1+j)=func(ib)
         do 275 k=ib,msct
            if(xsarr(is+k-1).ne.0.)xscon(lxsc+j4+j5-1+j)=func(k+1)
275   continue
      endif
278   continue
      j5=j5+33
      j6=j6+33
280   continue
282   continue
      nx1(1)=j6
      zp=nx1(2)+0.22
      write(hzaid(1:9),'(f9.2)')zp
      hzaid(10:10)='m'
      atwr=1.0d0/1.008665d0
      write(iuxt,400)hzaid,atwr,idtm(2:11)
      write(iuxt,401)(nx1(j),j=1,16)
      write(iuxt,401)(jx1(j),j=1,32)
      write(iuxt,402)(xscon(lxsc+j),j=1,nx1(1))
      do 284 j=1,80
284   dir(j:j)=' '
      dir(1:10)=hzaid
      write(dir(12:21),'(f10.6)')atwr
      dir(23:31)=hm
      dir(33:33)='0'
      dir(35:35)='1'
      write(dir(37:43),'(i7)')nrec
      write(dir(46:51),'(i6)')nx1(1)
      dir(54:56)='0 0'
      write(dir(58:67),'(1pe10.4)')zero
      write(iu1,403)dir
      write(iu2,404)hzaid(1:6),atwr
      nrec=nrec+12+(nx1(1)+3)/4
286   continue
      rewind(iu1)
      rewind(iu2)
      write(iuxx,'(a20)')'atomic weight ratios'
288   do 290 i=1,3
290   a21(i)=' '
      read(iu2,406,end=292)a21(1)
      read(iu2,406,end=292)a21(2)
      read(iu2,406,end=292)a21(3)
      write(iuxx,405)(a21(i),i=1,3)
      go to 288
292   write(iuxx,405)(a21(i),i=1,3)
      close(iu2)
      write(iuxx,'(a10)')idtm(2:11)
      write(iuxx,'(a9)')'directory'
294   read(iu1,403,end=296)dir
      write(iuxx,403)dir
      go to 294
296   close(iu1)
      close(iuxx)
      close(iuxt)
      return

```

```

        end
*deck xsf
      subroutine xsfl(k1,e,y)
c
c      write out data to cross section files
c
*call cm
      dimension e(6),y(6)
c
      k1=1
      write(iuxo,'(1p6e12.5)')(e(i),i=1,6)
      if(mxsc(6).eq.0)write(iuxr,'(1p6e12.5)')(y(i),i=1,6)
      do 10 i=1,6
      e(i)=0.
      y(i)=0.
10   continue
      return
      end
*deck xsf
      subroutine xsflc(ip,k1,e,y,a)
c
c      write out data to cross section files
c
*call cm
      dimension e(6),y(6)
      character a(6)*12,a1*72,a12*12
      data a12/'          '/
c
      100 format(a72)
      101 format(a5)
      102 format(a12)
      go to (5,10,20)ip
c      write cells
      5 k1=1
      write(iu3,'(6a12)')(a(i),i=1,6)
      do 7 i=1,6
      7 a(i)=a12
      return
c      write data
      10 k1=1
      write(iu1,'(1p6e12.5)')(e(i),i=1,6)
      write(iu2,'(1p6e12.5)')(y(i),i=1,6)
      do 15 i=1,6
      e(i)=0.
      y(i)=0.
15   continue
      return
c      write output
      20 rewind(iu1)
      rewind(iu2)
      rewind(iu3)
      write(iuxc,101)'cells'
      write(iuxc,102)a12
      25 read(iu3,100,end=30,err=55)a1
      write(iuxc,100)a1
      go to 25
      30 close(iu3)
      write(iuxc,102)a12
      write(iuxc,101)'vals'
      write(iuxc,102)a12
      35 read(iu1,100,end=40,err=55)a1
      write(iuxc,100)a1
      go to 35
      40 close(iu1)
      write(iuxc,102)a12
      write(iuxc,101)'rel'
      write(iuxc,102)a12
      45 read(iu2,100,end=50,err=55)a1
      write(iuxc,100)a1
      go to 45
      50 close(iu2)
      close(iuxc)
      return
      55 write(jtty,100)'ERROR DURING READ OF SCRATCH FILES'
      return
      end
*deck acx
      subroutine acecolx(jq,iflag)
c
c      sample an elastic or inelastic neutron collision.
c      jq=0 normally.

```

```

c      jq=2 for kcode sampling of fission energy only.
*call cm PARAMETER(THIRTY=0.03125D0)
*call cmx
*call cmx1
      character ht*10
      DIMENSION COSV(33),COSM(32)
c
      nt=0
10 if(jq.eq.2)go to 80
c
c      sample elastic vs. inelastic collision.
if(nxs(lnxs+5,iex).eq.0)go to 60
ic=ktc(kktc+2,iex)+jxs(ljxs+1,iex)-1
n=nxs(lnxs+3,iex)
el=xss(ic+3*n)+rtc(krtc+2,iex)*(xss(ic+3*n+1)-xss(ic+3*n))
if(erg.ne.eg0)go to 20
sr=rtc(krtc+4,iex)-rtc(krtc+3,iex)-rtc(krtc+8,iex)-el
go to 30
20 sr=xss(ic+n)-xss(ic+2*n)+rtc(krtc+2,iex)*
1 (xss(ic+n+1)-xss(ic+n)-xss(ic+2*n+1)+xss(ic+2*n))-el
if(lfc1(llfc+i1).eq.0)go to 30
l=jxs(ljxs+21,iex)
if(l.eq.0)go to 30
j=l+2+ktc(kktc+2,iex)-xss(l)
sr=sr-xss(j)-rtc(krtc+2,iex)*(xss(j+1)-xss(j))
30 if(awn(lawn+iex)*erg.gt.500.*tbt(ltbt+iex))go to 50
a2=awn(lawn+iex)*erg/tbt(ltbt+iex)
if(a2.lt.4.)go to 40
el=el*a2/(a2+.5)
go to 50
40 a=sqrt(a2)
b=25.*a
i=b
el=el*a/(thgf(i)+(b-i)*(thgf(i+1)-thgf(i)))
50 r=rang()*(sr+el)-el
if(r.ge.0.)go to 100
c
***** elastic case *****
*
60 ixre=0
ntyn=-99
cmult=1.
c
IF(IFLAG.NE.2)GO TO 65
*****XSMODS*****
*
A2=AWN(LAWN+IEX)
A3=2.*A2
A4=A2**2+1.
A5=(A2+1.)**2
c
JS=FLAG FOR ISOTROPIC OF NOT
CALL ACECOSX(COSV,COSM,JS)
c
USING ERGOLDE FOR KINEMATICS SINCE EQUATIONS START WITH LAB ERG
IFLAG=-2
IF(IFLTG.EQ.1)THEN
CAZM=COS(RANG())*2.*PIE)
B1=2.*SQRT(A2*ERGOLDE*ETAR)*CTAR
B=ERGOLDE+A2*ETAR+B1
C=A2*ERGOLDE+ETAR-B1
D=SQRT(ERGOLDE)+SQRT(A2*ETAR)*CTAR
D1=SQRT(A2*B*C)*2.0
D2=B+A2*C
D3=SQRT((A2*C)/B)
D4=SQRT(A2*C)
D5=SQRT(1.-(D**2)/B)
ENDIF
IG=GROUP FROM
IG1=GROUP TO
IGT=GROUP THAT TOP ENERGY BELONGS TO
IGB=GROUP THAT BOTTOM ENERGY BELONGS TO
BOTTOM ENERGY WILL GET REASSIGNED AS INTEGRATION PROGRESSES
IGFN=IG GROUP STARTING POSITION WITHIN ENERGY GRID
IP=POINTER IN ENERGY GRID OF TOP ENERGY OF BOTTOM GROUP
TOP=TOP ENERGY THAT CAN SCATTER TO
TMUC= MUC CORRESPONDING TO TOP
BOT= BOTTOM ENERGY THAT CAN SCATTER TO
BMUC=MUC CORRESPONDING TO BOT
c
ICHK=0

```

```

      IGFN=IXSN+1-IG
C
      IF(IFLTG.EQ.0)THEN
        TOP=ECM(ERGOLDE,COSV(33))
        BOT=ECM(ERGOLDE,COSV(1))
      ELSE
        TOP=ECMM(COSV(33))
        BOT=ECMM(COSV(1))
      ENDIF
      TMUC=COSV(33)
      BMUC=COSV(1)

C     CHECK TO MAKE SURE TOP ENERGY IS NOT OUT OF RANGE
      IF(TOP.GT.XSEN(LXSN+IXSN+1))THEN
        ICHK=ICHK+1
        TOP=XSEN(LXSN+IXSN+1)
        IF(IFLTG.EQ.0)THEN
          TMUC=CCM(ERGOLDE, TOP)
        ELSE
          TMUC=CCMM(TOP)
        ENDIF
        IGT=1
        GO TO 520
      ENDIF
C     DETERMINE WHICH GROUP TOP BELONGS TO
C     USING ERGOLDE BECAUSE ERG MAY BE RELATIVE VELOCITY
      IF(TOP.EQ.ERGOLDE)THEN
        IGT=IG
        GO TO 520
      ENDIF
      IF(TOP.LT.ERGOLDE)THEN
        DO 500 I=IGFN,1,-1
          IF(TOP.GT.XSEN(LXSN+I))GO TO 505
        CONTINUE
      500 SCATTERING NOT POSSIBLE IN USER DEFINED GROUP STRUCTURE
      RETURN
      505 IGT=IXSN+1-I
      GO TO 520
      ENDIF
      IF(TOP.GT.ERGOLDE)THEN
        DO 510 I=IGFN+1,IXSN+1
          IF(TOP.LT.XSEN(LXSN+I))GO TO 515
        CONTINUE
      510 IGT=IXSN+1-(I-1)
      515
      ENDIF
C     IF BOT OUT OF RANGE THEN RESET
      520 IF(BOT.LT.XSEN(IXSN+1))THEN
        ICHK=ICHK+1
        BOT=XSEN(IXSN+1)
        IF(IFLTG.EQ.0)THEN
          BMUC=CCM(ERGOLDE, BOT)
        ELSE
          BMUC=CCMM(BOT)
        ENDIF
        IGB=IXSN
        IP=2
        GO TO 535
      ENDIF
C     FIND WHICH GROUP BOT BELONGS TO
      DO 525 I=2,IXSN+1
        IF(BOT.LT.XSEN(IXSN+I))GO TO 530
      525 CONTINUE
      530 IGB=IXSN+2-I
      IP=I
C
C     BEGIN CALCULATION OF GROUP FRACTIONS AND LEGENDRE INFORMATION
C
      535 LY=0
      IF(IGT.EQ.IGB.AND.ICHK.EQ.0)THEN
        CALL AREAFULL(LY,COSV,IGB,BMUC,TMUC)
        RETURN
      ENDIF
      CALL AREAINT(LY,COSV,COSM,BMUC,TMUC,IP,IGB,IGT)
      RETURN
      *
      *****XSMODS*****
C
C       sample the neutron output direction and calculate its energy.
      65   c=acecos(i)
      ixcos=i

```

```

if(awn(lawn+iex).lt.1.)go to 70
t1=1.+awn(lawn+iex)*(awn(lawn+iex)+2.*c)
colout(1,1)=erg*t1/(1.+awn(lawn+iex))**2
colout(2,1)=(1.+awn(lawn+iex)*c)/sqrt(t1)
return
c      special calculation for hydrogen.
70 colout(1,1)=.5*erg*(1.+c)
colout(2,1)=sqrt(.5+.5*c)
return
c **** sample the reaction, using cumulative partial cross sections.
c
c      CAN'T HANDLE INELASTIC WITH TARGET IN MOTION, MUST DO OLD WAY
80 IF(IFLAG.EQ.2.AND.IFLTG.EQ.1)IFLAG=0
if(erg.ne.eg0)go to 90
r=rang()*rtc(krtc+8,iex)
go to 100
90 l=jxs(ljxs+21,iex)
j=l+2+ktc(kktc+2,iex)-xss(1)
r=rang()*(xss(j)+rtc(krtc+2,iex)*(xss(j+1)-xss(j)))
100 tt=0.
kx=0
do 120 ixre=1,nxs(lnxs+5,iex)
if(lfc1(lfc+ic1).eq.0)go to 110
if(jq.ne.2.eqv.xss(jxs(ljxs+5,iex)+ixre-1).ge.18)go to 120
c      calculate the cross section for reaction whose index is ixre.
c      data: ie,ne,(xs(i),i=1,ne)   xs(1) corresponds to es(ie).
110 is=jxs(ljxs+7,iex)+xss(jxs(ljxs+6,iex)+ixre-1)
ic=ktc(kktc+2,iex)+1-xss(is-1)
if(ic.lt.1.or.ic.gt.xss(is).or.ic.eq.xss(is).and.
1 rtc(krtc+2,iex).ne.0.)go to 120
t=xss(ic+is)
if(rtc(krtc+2,iex).ne.0.)t=t+rtc(krtc+2,iex) *
1 (xss(ic+is+1)-xss(ic+is))
if(t.eq.0.)go to 120
if(t.gt.tt)kx=ixre
tt=max(t,tt)
r=r-t
if(r.lt.0.)go to 150
120 continue
c      handle failure of the reaction cross sections to add up.
if(kx.eq.0)go to 60
call zaid(2,ht,ixl(lixl+1,iex))
call errprn(1,npnm,-1,4,erg,zero,'erg',' ')
1 'no reaction mt found. collision resampled. zaid = '//ht)
ixre=kx
c      get the reaction type number.
150 ntyn=xss(jxs(ljxs+5,iex)+ixre-1)
c      check that the energy is within the band for this reaction.
if(nty(lnty+iex).ne.2)go to 160
l=jxs(ljxs+11,iex)+2+xss(jxs(ljxs+10,iex)+ixre-1)
ie=2*xss(1)+1
if(erg.le.xss(ie+2).or.erg.ge.xss(ie+1+int(xss(ie+1))))go to 180
c      sample the energy and scattering angle of emerging neutrons.
160 ns=1
if(abs(ntyn).ge.100)go to 190
cmult=abs(ntyn)
if(ntyn.eq.19)cmult=aint(acenu()+rang())
C THIS LINE REMOVED BECAUSE ACECOLX ONLY USED FOR XS GENERATION AND
C WHEN SAMPLING NU ONE WANTS TO SAMPLE NU TIMES NOT JUST ONCE
C IF(JQ.EQ.0)NS=CMULT
NS=CMULT
do 170 i=1,ns
CALL ACECAXS(I,IFLAG)
if(iflag.lt.0)return
170 if(kdb.ne.0)return
return
c      try again, up to 100 times, if the energy is out of the
c      reaction band.
180 nt=nt+1
ndr(kndr+iex)=ndr(kndr+iex)+1
if(nt.le.100)go to 10

```

```

call expirx(1,'acecol',
1 'energy was not within the band for the reaction 100 times.')
return
c
c      high energy reaction other than fission with energy-dependent
c      multiplicity.  kalbach-87 (law 44) endf/b-vi.
c
c      the location of the multiplicity table relative to dlw is
c      abs(ntyn)-100.  acefcn gets non-fission reaction multiplicity.
190 l=jxs(ljxs+11,iex)+abs(ntyn)-101
WRITE(IUO,191)
WRITE(IUO,192)
WRITE(IUO,193)
191 FORMAT('WARNING: LAW 44 USED - THIS LAW HAS NOT BEEN CHECKED')
192 FORMAT('TO VERIFY THAT IT WORKS WITH XS GENERATION')
193 FORMAT('USE AT OWN RISK')
cmult=acefcn(l,erg,ln)
if(cmult.lt.1.)go to 210
cmult=aint(cmult+rang())
do 200 i=1,int(cmult)
CALL ACECASK(I,IFLAG)
200 if(kdb.ne.0)return
return
c
c      multiplicity < 1.0 ; reduce weight by "absorption."
c
c      analog capture case.
210 if(wcl(1).ne.0..and.erg.gt.emcf(1))go to 220
if(cmult+rang().le.1.)go to 280
go to 270
c
c      otherwise simulate capture by weight reduction.
220 if(cmult.eq.0.)go to 280
w1=(1.-cmult)*wgt
wgt=wgt*cmult
if(nsrl.ne.71)go to 240
do 230 i=1,2
rltcc(2,i,1)=rlttc(2,i,1)+w1*tme
230 rlttc(2,i,2)=rlttc(2,i,2)+w1
240 if(igww.ne.0)call wgtwg(1,w1)
tmavtc(1,2)=tmavtc(1,2)+tme*w1
tmavtc(1,3)=tmavtc(1,3)+tme*w1
paxtc(5,12,1)=paxtc(5,12,1)+w1
paxtc(6,12,1)=paxtc(6,12,1)+w1*eg0
pwb(kpwb+1,15,ic1)=pwb(kpwb+1,15,ic1)-w1
pan(kpan+1,3,mpan)=pan(kpan+1,3,mpan)+w1
c
c      do russian roulette on weight cutoff as required.
if(abs(wwp(1,4)).eq.1..and.idx.eq.0)go to 270
if(for(lfor+1,ic1).ne.0.)go to 270
if(idx.eq.0)go to 250
if(wgt*fiml(1).gt.fiml9(1,1)*dxw(1,2))go to 270
t1=dxw(1,1)*fiml9(1,1)/fiml(1)
go to 260
250 if(wgt*fiml(1).gt.fiml9(1,1)*wcs2tc(1))go to 270
t1=wcs1tc(1)*fiml9(1,1)/fiml(1)
260 if(wgt.lt.t1*rang())go to 300
paxtc(2,6,1)=paxtc(2,6,1)+(t1-wgt)
paxtc(3,6,1)=paxtc(3,6,1)+(t1-wgt)*eg0
pwb(kpwb+1,8,ic1)=pwb(kpwb+1,8,ic1)+t1-wgt
wgt=t1
c
c      sample the energy and scattering angle of emerging neutrons.
270 cmult=1.
CALL ACECASK(1,IFLAG)
return
c
c      kill particles of zero multiplicity or by analog capture.
280 nter=12
pan(kpan+1,3,mpan)=pan(kpan+1,3,mpan)+wgt
if(krflg.ne.0)call eventp(4)
if(nsrl.ne.71)return
do 290 i=1,2
rltcc(2,i,1)=rlttc(2,i,1)+wgt*tme
290 rlttc(2,i,2)=rlttc(2,i,2)+wgt
return
c
c      kill particles below weight cutoff.
300 nter=6
if(krflg.ne.0)call eventp(4)

```

```

        return
    end
*deck asx
    subroutine acecasx(ls,iflag)
c           sample the energy of an output neutron or photon from law data.
c           also sample the scattering cosine.
c           convert to the nucleus-at-rest system if necessary.
*call cm
    PARAMETER (THIRTY=0.03125d0)
*call cmx
*call cmx1
    DIMENSION COSV(33),COSM(32),PV(0:ILEG),BIS(NBIN),B1(2,50)
    character ht*10
    dimension ml(13),nl(13)
    data ml/1,0,0,1,-1,-1,-1,1,1,1,1,0,1/
    data nl/1,2,3,4,5,7,9,11,22,24,44,66,67/
c
131  FORMAT('XSERR: SCATTERING OCCURRED FROM GRP ',I5,' TO ',I5,
     1 ' BUT NOT STORED')
c
c       select the law.
nx=0
10 t1=rang()
n=jxs(ljxs+8*ipt+2,iex)+ixre
20 n=jxs(ljxs+8*ipt+3,iex)+xss(n-1)
    if(xss(n-1).eq.0)go to 30
    t1=t1-acefcn(n+2,erg,1)
    if(t1.ge.0.)go to 20
c
c       use the selected law to sample the energy.
30 colout(1,ls)=-huge
lw=xss(n)
celr
celr      lawelr=lw
celr
    iw=jxs(ljxs+8*ipt+3,iex)-1+xss(n+1)
    do 32 i=1,13
        lawelr=i
32 if(nl(i).eq.lw)go to 34
    go to 300
34 if(ml(i).gt.0)call acetbl(iw,ic,r,1)
    if(ml(i).lt.0)t1=acefcn(iw,erg,1)
    go to(40,60,70,80,160,170,190,210,230,242,80,245,255)i
c
c >>>> law 1 -- tabular neutron energies out.
40 IF(IFLAG.EQ.2)THEN
    IFLAG=-2
    NT=XSS(IW+L)
    IF(NT.GT.50)THEN
        WRITE(IUO,35)NT
35     FORMAT('VARIABLE B1 TOO SMALL - NT=',I6)
        CALL EXPIRX(1,'ACECASX','VARIABLE B1 IS DIM TOO SMALL')
    ENDIF
    IW=IW+NT*(IC-1)+L
    AREA=1.0D0/(NT-1)
    DO 41 I=1,IXSN
        XSNT(LXNT+I)=0.0
41    CONTINUE
    IF(R.EQ.0.)THEN
        B1(1,1)=XSS(IW+1)
        DO 42 I=2,NT
            B1(1,I)=XSS(IW+I)
            B1(2,I-1)=AREA/(B1(1,I)-B1(1,I-1))
42    CONTINUE
    ELSE
        T1=XSS(IW+1)+R*(XSS(IW+1+NT)-XSS(IW+1))
        T2=XSS(IW+NT)+R*(XSS(IW+2*NT)-XSS(IW+NT))
        T3=T2-T1
        I1=IW
        IF(RANG().LE.R)I1=I1+NT
        T4=(XSS(I1+NT)-XSS(I1+1))
        T5=T3/T4
        B1(1,1)=T1
        DO 43 I=2,NT-1
            B1(1,I)=T1+T5*(XSS(I1+I)-XSS(I1+1))
            B1(2,I-1)=AREA/(B1(1,I)-B1(1,I-1))
43    CONTINUE
        B1(1,NT)=T2
        B1(2,NT-1)=AREA/(B1(1,NT)-B1(1,NT-1))
    ENDIF

```

```

C      CHECK THAT FIRST ENERGY IS NOT GREATER THAN TOP ENERGY
      IF(B1(1,1).GE.XSEN(LXSN+IXSN+1))RETURN
C      CHECK THAT LAST ENERGY IS NOT LESS THAN FIRST ENERGY
      IF(B1(1,NT).LE.XSEN(LXSN+1))RETURN
C      FIND START VALUE OF INTEGRATION
C      IB=START OF NEXT USER GROUP
C      IA=START OF NEXT OUTGOING ENERGY
      IF(B1(1,1).LT.XSEN(LXSN+1))THEN
          ST=XSEN(LXSN+1)
          IB=2
          DO 44 I=2,NT
              IF(B1(1,I).GT.ST)THEN
                  IA=I
                  GO TO 46
              ENDIF
        CONTINUE
44    ELSE IF((B1(1,1).GT.XSEN(LXSN+1)))THEN
          ST=B1(1,1)
          IA=2
          DO 45 I=1,IXSN
              IF(XSEN(LXSN+I).GT.ST)THEN
                  IB=I
                  GO TO 46
              ENDIF
45    CONTINUE
        ELSE
          ST=B1(1,1)
          IA=2
          IB=2
        ENDIF
C      PERFORM INTEGRATION
46    IF(XSEN(LXSN+IB).LT.B1(1,IA))THEN
          XSNT(LXNT+IB-1)=XSNT(LXNT+IB-1)+  

$          (XSEN(LXSN+IB)-ST)*B1(2,IA-1)
          ST=XSEN(LXSN+IB)
          IB=IB+1
          IF(IB.GT.IXSN+1)GO TO 47
          GO TO 46
        ENDIF
        IF(XSEN(LXSN+IB).GT.B1(1,IA))THEN
          XSNT(LXNT+IB-1)=XSNT(LXNT+IB-1)+  

$          (B1(1,IA)-ST)*B1(2,IA-1)
          ST=B1(1,IA)
          IA=IA+1
          IF(IA.GT.NT)GO TO 47
          GO TO 46
        ENDIF
        IF(XSEN(LXSN+IB).EQ.B1(1,IA))THEN
          XSNT(LXNT+IB-1)=XSNT(LXNT+IB-1)+  

$          (B1(1,IA)-ST)*B1(2,IA-1)
          ST=B1(1,IA)
          IA=IA+1
          IB=IB+1
          IF(IA.GT.NT)GO TO 47
          IF(IB.GT.IXSN+1)GO TO 47
          GO TO 46
        ENDIF
47    CALL ACECOSX(COSV,COSM,JS)
C      NO PL IS THE SAME AS ISOTROPIC IN LAB
      IF(MXSC(1).EQ.0)JS=1
      IF(JS.EQ.1)GO TO 48
      IF(MXSC(5).EQ.1)CALL PLDIRREC(PV,COSV,COSM)
      IF(MXSC(5).EQ.0)CALL BINOVER(BIS,COSV,COSM)
48    W7=W4
      W4=W4*CMULT
      DO 53 I=1,IXSN
          IF(XSNT(LXNT+I).EQ.0.)GO TO 53
          IG1=IXSN+1-I
          I1=ISPO(LISP+1,(IG-1)*MXSD(1)+IG1)
          IF(I1.EQ.0)THEN
              WRITE(IUO,131)IG,IGB
              WRITE(JTTY,131)IG,IGB
              GO TO 53
          ENDIF
          I1=I1+MXSD(1)*6
          I2=MXSD(1)*6+ISPO(LISP+2,(IG-1)*MXSD(1)+IG1)
          VA=XSNT(LXNT+I)*W4
          DO 52 J=1,1DATR(LIDR+IM1)
              II=KXSC+1DATR(LIDR+IM1+J)
              JP=II+I1-1

```

```

XSCON(JP)=XSCON(JP)+VA
IF(JS.EQ.1)GO TO 52
IF(MXSC(5).EQ.1)GO TO 50
JP=II+I2-1
DO 49 K=1,MSCT
    XSCON(JP)=XSCON(JP)+VA*BIS(K)
    JP=JP+1
49    CONTINUE
    GO TO 52
50    DO 51 K=1,MXSC(1)
        JP=JP+MXSD(2)
        XSCON(JP)=XSCON(JP)+VA*PV(K)
51    CONTINUE
52    CONTINUE
53    CONTINUE
    W4=W7
    RETURN
ENDIF
IFLAG=0
nt=xss(iw+1)
iw=iw+nt*(ic-1)+1
k=rang()*(nt-1)+1
fr=rang()
if(r.ne.0.)go to 55
c      sample from single table.
colout(1,ls)=xss(iw+k)+fr*(xss(iw+k+1)-xss(iw+k))
go to 260
c      sample by scaled interpolation between tables.
55    t1=xss(iw+1)+r*(xss(iw+1+nt)-xss(iw+1))
    i=iw
    if(rang().le.r)i=i+nt
    colout(1,ls)=t1+(xss(i+k)+fr*(xss(i+k+1)-xss(i+k))-xss(i+1))*1
    (xss(iw+nt)+r*(xss(iw+2*nt)-xss(iw+nt))-t1)/(xss(i+nt)-xss(i+1))
    go to 260
c
c >>>> law 2 -- discrete photon lines.
60 IFLAG=0
    t=xss(iw+1)
    if(xss(iw).eq.2.)t=t+erg*awn(lawn+iex)/(awn(lawn+iex)+1.)
    go to 320
c
c >>>> law 3 -- neutron level scattering.
70 IF(IFLAG.EQ.2)THEN
    IFLAG=-2
    A2=AWN(LAWN+IEX)+1.0D0
    A3=A2**2
    A2=2.0D0*A2
    A4=XSS(IW+1)*(ERG-XSS(IW))
    A5=SQRT(ERG*A4)
    CALL ACECOSK(COSV,COSM,JS)
c
    ICHK=0
    IGFN=IXSN+1-IG
c
    TOP=E3C(ERG,COSV(33))
    TMUC=COSV(33)
c
    CHECK TO MAKE SURE TOP ENERGY IS NOT OUT OF RANGE
    IF(TOP.GT.XSEN(LXSN+IXSN+1))THEN
        ICHK=ICHK+1
        TOP=XSEN(LXSN+IXSN+1)
        TMUC=C3M(ERG,TOP)
        IGT=1
        GO TO 76
    ENDIF
c
    DETERMINE WHICH GROUP TOP BELONGS TO
    IF(TOP.EQ.ERG)THEN
        IGT=IG
        GO TO 76
    ENDIF
    IF(TOP.LT.ERG)THEN
        DO 72 I=IGFN,1,-1
            IF(TOP.GT.XSEN(LXSN+I))GO TO 73
72    CONTINUE
c
    SCATTERING NOT POSSIBLE IN USER DEFINED GROUP STRUCTURE
    RETURN
73    IGT=IXSN+1-I
    GO TO 76
    ENDIF
    IF(TOP.GT.ERG)THEN
        DO 74 I=IGFN+1,IXSN+1

```

```

        IF(TOP.LT.XSEN(LXSN+I))GO TO 75
74      CONTINUE
75      IGT=IXSN+1-(I-1)
ENDIF
C      DETERMINE BOTTOM ENERGY
76      BOT=E3C(ERG,COSV(1))
BMUC=COSV(1)
C      IF BOT OUT OF RANGE THEN RESET
IF(BOT.LT.XSEN(LXSN+1))THEN
ICHK=ICHK+1
BOT=XSEN(LXSN+1)
BMUC=C3M(ERG,BOT)
IGB=IXSN
IP=2
GO TO 79
ENDIF
C      FIND WHICH GROUP BOT BELONGS TO
DO 77 I=2,IXSN+1
IF(BOT.LT.XSEN(LXSN+I))GO TO 78
77      CONTINUE
78      IGB=IXSN+2-I
IP=I
C
C      BEGIN CALCULATION OF GROUP FRACTIONS AND LEGENDRE INFORMATION
C
79      LY=3
C      SAVE W4 TO BE ABLE TO RESET IT
W7=W4
W4=W4*CMULT
IF(IGT.EQ.IGB.AND.ICHK.EQ.0)THEN
CALL AREAFULL(LY,COSV,IGB)
W4=W7
RETURN
ENDIF
CALL AREAINT(LY,COSV,COSM,BMUC,TMUC,IP,IGB,IGT)
W4=W7
RETURN
ENDIF
IFLAG=0
colout(1,ls)=xss(iw+1)*(erg-xss(iw))
go to 260
C
C >>>> law 4 -- sample energy distribution tables.
C >>>> law 44 -- kalbach-87 endf/b-vi
80 IFLAG=0
nr=xss(iw)
lb=iw+1+2*nr+xss(iw+1+2*nr)+ic
lc=jxs(ljxs+3+8*ipt,iex)+xss(lb)
ld=lc
lf=lc
np=xss(lc)
mp=np
jj=xss(lc-1)
nd=0
if(jj.lt.9999)nd=jj/10
ra=2.
if(r.eq.0.)go to 90
ld=jxs(ljxs+3+8*ipt,iex)+xss(lb+1)
mp=xss(ld)
t1=xss(lc+nd+1)+r*(xss(ld+nd+1)-xss(lc+nd+1))
t2=xss(lc+np)+r*(xss(ld+mp)-xss(lc+np))
ra=rang()
if(ra.ge.r)go to 90
lf=ld
jj=xss(ld-1)
if(jj.lt.9999.and.jj/10.ne.nd)call expirx(1,'acecas',
1 'wrong number of discrete lines for law 4/44')
90 if(jj.eq.9999)return
jj=jj-nd*10
r1=rang()
C
C      check for a hit in the discrete part.
C      use histogram or corresponding-point interpolation.
if(nd.eq.0)go to 97
do 95 ih=1,nd
cc=xss(lc+2*np+ih)+r*(xss(ld+2*mp+ih)-xss(lc+2*np+ih))
95 if(cc.ge.r1)go to 96
if(np.eq.nd.and.mp.eq.nd)go to 152
go to 97
96 t=xss(lc+ih)+r*(xss(ld+ih)-xss(lc+ih))

```

```

if(t.lt.0.)t=erg*awn(lawn+iel) / (awn(lawn+iel)+1.)-t
if(lw.ne.44)go to 152
tpd(1)=xss(lc+ih+3*np)+r*(xss(ld+ih+3*mp)-xss(lc+ih+3*np))
tpd(2)=xss(lc+ih+4*np)+r*(xss(ld+ih+4*mp)-xss(lc+ih+4*np))
go to 152
c
c      handle a hit in the continuous part.
c      use histogram or unit-base interpolation inside table.
c      use scaled interpolation between tables.
97 np=xss(lf)
ns=nd+1
if(nd.ne.0)r1=1.-(1.-r1)*(1.-xss(lf+2*np+nd))/(1.-cc)
c
c      adjust for the energy cutoff if necessary.
c      see subroutine expung for law 4/44 distribution. 0 < wc < 1
if(jj.lt.9999)go to 100
jj=jj/10000
ns=xss(lf-1)-jj*10000
wc=2.*(xss(lf)-np)
r1=1.-wc*(1.-r1)
wgt=wgt*wc
c
100 ic=lf+2*np+ns
ib=lf+3*np
c      find the energy bin in the continuous part.
110 if(ib-ic.eq.1)go to 130
ih=(ic+ib)/2
if(r1.lt.xss(ih))go to 120
ic=ih
go to 110
120 ib=ih
go to 110
130 l=ic-2*np
fa=xss(l+np)
ea=xss(l)
c
c      sample from linear-linear interpolated bin.
if(jj.eq.1)go to 140
bb=(xss(l+np+1)-fa)/(xss(l+1)-ea)
if(bb.eq.0)go to 140
t=ea+(sqrt(max(zero,fa**2+2.*bb*(r1-xss(ic)))-fa)/bb
if(lw.ne.44)go to 148
fb=(t-xss(l))/(xss(l+1)-xss(l))
tpd(1)=xss(ic+np)+fb*(xss(ic+1+np)-xss(ic+np))
tpd(2)=xss(ic+2*np)+fb*(xss(ic+1+2*np)-xss(ic+2*np))
go to 150
c
c      sample from histogram bin.
140 t=ea+(r1-xss(ic))/fa
if(lw.ne.44)go to 148
tpd(1)=xss(ic+np)
tpd(2)=xss(ic+2*np)
go to 150
c
c      for law=4 photons, do not interpolate.
148 if(ip.t.eq.2)go to 320
c
c      use scaled interpolation between energies.
150 if(r.ne.0.)t=t1+(t-xss(lf+nd+1))*(t2-t1)/(xss(lf+np)-xss(lf+nd+1))
152 if(ip.t.eq.2)go to 320
colout(1,ls)=t
if(lw.ne.44)go to 260
c
c      sample law 44 -- kalbach-87 endf/b-vi (r,a both given)
c      tpd(1)=r, tpd(2)=a
if(nint(xss(jxs(ljxs+8,iex)+ixre)).ne.-1.or.ntyn.ge.0)go to 295
if(colout(1,ls).lt.0..or.colout(1,ls).gt.erg)go to 300
ipsc=14
if(rang().le.tpd(1))go to 155
t1=(2.*rang()-1.)*sinh(tpd(2))
colout(2,ls)=log(t1+sqrt(t1**2+1.))/tpd(2)
go to 280
155 r2=rang()
colout(2,ls)=log(r2*exp(tpd(2))+(1.-r2)*exp(-tpd(2)))/tpd(2)
go to 280
c
c >>>> law 5 -- general neutron evaporation spectrum.
160 IFLAG=0
i=iw+1+rang()*(xss(iw)-1)+1
colout(1,ls)=t1*(xss(i)+rang()*(xss(i+1)-xss(i)))

```

```

    go to 260
c
c >>> law 7 -- simple fission spectrum.
170 IFLAG=0
    t3=erg-xss(iw+1)
    if(t3.le.0.)go to 260
180 t4=rang()**2
    t2=t4+rang()**2
    if(t2.gt.1.)go to 180
    t2=log(rang())*t4/t2
    colout(1,ls)=-t1*(t2+log(rang()))
c        reject if outside range 0 ... e-u
    if(colout(1,ls).gt.t3)go to 180
    go to 260
c
c >>> law 9 -- neutron evaporation spectrum.
190 t2=erg-xss(iw+1)
    if(t2.le.0.)go to 260
    IF(IFLAG.GT.0)THEN
        XT1=0.0
        IE=0
        IF(IFLAG.EQ.1)IFLAG=-1
        C1=T1**2
        C2=C1-EXP(-T2/T1)*(C1+T1*T2)
        C3=1./C2
        C4=1./T1
        IF(IFLAG.EQ.-1)THEN
            CU=1.0E-3
        ELSE
            CU=0.0
        ENDIF
    C     CALCULATE CDF AT EACH ENERGY
    DO 191 I=1,IXSN
        XSNT(LXNT+I)=0.0
        X1=XSEN(LXSN+I)
        IF(X1.GE.T2)GO TO 191
        X2=XSEN(LXSN+I+1)
        IF(X2.LT.CU)GO TO 191
        IF(X2.GT.T2)X2=T2
        C6=EXP(-X2*C4)
        IF(IE.EQ.0)THEN
            IE=1
            C5=EXP(-X1*C4)
            XT1=C3*(-1.0*(T1*X1*C5)-C1*C5)
        ENDIF
        XSNT(LXNT+I)=C3*(-1.0*(T1*X2*C6)-C1*C6)
    191    CONTINUE
    C     CALCULATE GROUP FRACTION
    DO 192 I=1,IXSN
        IF(XSNT(LXNT+I).EQ.0.0)GO TO 192
        XT2=XSNT(LXNT+I)
        XSNT(LXNT+I)=XSNT(LXNT+I)-XT1
        XT1=XT2
    192    CONTINUE
        IF(IFLAG.EQ.-1)RETURN
        IFLAG=-2
        CALL ACECOSX(COSV,COSM,JS)
    C     NO PL IS THE SAME AS ISOTROPIC IN LAB
        IF(MXSC(1).EQ.0)JS=1
        IF(JS.EQ.1)GO TO 194
        IF(MXSC(5).EQ.1)CALL PLDIREC(PV,COSV,COSM)
        IF(MXSC(5).EQ.0)CALL BINOVER(BIS,COSV,COSM)
    194    W7=W4
        W4=W4*CMULT
        DO 200 I=1,IXSN
            IF(XSNT(LXNT+I).EQ.0.)GO TO 200
            IG1=IXSN+1-I
            I1=ISPO(LISP+1,(IG-1)*MXSD(1)+IG1)
            IF(I1.EQ.0)THEN
                WRITE(IUO,131)IG,IGB
                WRITE(JTTY,131)IG,IGB
                GO TO 200
            ENDIF
            I1=I1+MXSD(1)*6
            I2=MXSD(1)*6+ISPO(LISP+2,(IG-1)*MXSD(1)+IG1)
            VA=XSNT(LXNT+I)*W4
            DO 198 J=1,1DATR(LIDR+IM1)
                II=KXSC+1DATR(LIDR+IM1+J)
                JP=II+I1-1
                XSCON(JP)=XSCON(JP)+VA

```

```

        IF(JS.EQ.1)GO TO 198
        IF(MXSC(5).EQ.1)GO TO 196
        JP=II+I2-1
        DO 195 K=1,MSCT
          XSCON(JP)=XSCON(JP)+VA*BIS(K)
          JP=JP+1
195      CONTINUE
          GO TO 198
196      DO 197 K=1,MXSC(1)
          JP=JP+MXSD(2)
          XSCON(JP)=XSCON(JP)+VA*PV(K)
197      CONTINUE
198      CONTINUE
200      CONTINUE
          W4=W7
          RETURN
        ENDIF
209      fr=rang()
        IFLAG=0
        colout(1,ls)=-t1*log(fr*rang())
        if(colout(1,ls).gt.t2)go to 209
        go to 260
C
C >>>> law 11 -- energy-dependent watt spectrum.
210      t2=acefcn(iw+1,erg,11)
        if(erg.le.xss(iw+1+11))go to 260
        IF(IFLAG.EQ.1)THEN
          XT1=0.0
          IE=0
          IFLAG=-1
          C1=ERG-XSS(IW+L+L1)
          C2=T1*T2*0.25
          C3=EXP(C2)
          C4=SQRT(C2)
          C5=SQRT(C1/T1)
          C6=0.5*SQRT(PIE*(T1**3)*T2*0.25)
          C7=1./T1
          C8=C6*C3*(ERF2(C5-C4)+ERF2(C5+C4))-1
          T1*EXP(-C1*C7)*SINH(SQRT(T2*C1))
          C9=1./C8
C       CALCULATE CDF AT EACH ENERGY
        DO 211 I=1,IXSN
          XSNT(LXNT+I)=0.0
          X1=XSEN(LXSN+I)
          IF(X1.GE.C1)GO TO 211
          X2=XSEN(LXSN+I+1)
          IF(X2.LT.1.0E-3)GO TO 211
          IF(X2.GT.C1)X2=C1
          C10=SQRT(X2*C7)
          IF(IE.EQ.0)THEN
            IE=1
            C11=SQRT(X1*C7)
            XT1=C9*(T1*(-EXP(-X1*C7)*SINH(SQRT(T2*X1)))+
            +C3*C6*(ERF2(C11-C4)+ERF2(C11+C4)))
          ENDIF
          XSNT(LXNT+I)=C9*(T1*(-EXP(-X2*C7)*SINH(SQRT(T2*X2)))+
          +C3*C6*(ERF2(C10-C4)+ERF2(C10+C4)))
1           CONTINUE
211      CALCULATE GROUP FRACTIONS
        DO 212 I=1,IXSN
          IF(XSNT(LXNT+I).EQ.0.0)GO TO 212
          XT2=XSNT(LXNT+I)
          XSNT(LXNT+I)=XSNT(LXNT+I)-XT1
          XT1=XT2
212      CONTINUE
          RETURN
        ENDIF
        IFLAG=0
        t5=sqrt((1.+125*t1*t2)**2-1.)*1.+125*t1*t2
220      t=-log(rang())
        colout(1,ls)=t1*t5*t
        if(((1.-t5)*(1.+t)-log(rang()))**2.gt.t2*colout(1,ls))go to 220
        go to 260
C
C >>>> neutron law 22 (uk law 2)
230      IFLAG=0
        ie=jxs(ljxs+11,iex)-1+xss(iw+1+ic-1)
        nf=xss(ie)
235      iw=ie
        fr=rang()

```

```

240 iw=iw+1
fr=fr-xss(iw)
if(fr.ge.0.)go to 240
if(iw.gt.ie+nf)go to 235
colout(1,ls)=xss(iw+2*nf)*(erg-xss(iw+nf))
go to 260
c
c >>>> neutron law 24 (uk law 6)
242 IFLAG=0
i=iw+1+xss(iw+1)*(ic-1)+rang()*(xss(iw+1)-1)
colout(1,ls)=erg*(xss(i)+rang()*(xss(i+1)-xss(i)))
go to 260
c
c >>>> law 66 (endf/b-vi law 6 file 6) -- phase space.
245 IFLAG=0
nb=xss(iw)
ap=xss(iw+1)
246 r=rang()**2
s=r+rang()**2
if(s.gt.1.)go to 246
x=-r*log(s)/s-log(rang())
248 r=rang()**2
s=r+rang()**2
if(s.gt.1.)go to 248
p=rang()
go to(252,251,250)nb-2
250 p=p*rang()*rang()
251 p=p*rang()
252 y=-r*log(s)/s-log(p)
t=x/(x+y)
aw=aawn(lawn+ixex)
colout(1,ls)=t*((ap-1.)/ap)*(erg*aw/(aw+1.)*
1 xss(jxs(1jxs+4,ixex)+ixre-1))
colout(2,ls)=2.*rang()-1.
ixcos=0
if(ntyn.ge.0)go to 295
if(colout(1,ls).lt.0..or.colout(1,ls).gt.erg)go to 300
go to 280
c
c >>>> law 67 (endf/b-vi law 7) -- correleted energy-angle scatter.
255 IFLAG=0
cs=acecos(i)
ixcos=i
ipsc=15
colout(2,ls)=cs
colout(1,ls)=acecs6(0,iw,ic,r,cs)
if(ntyn.le.0)go to 295
if(colout(1,ls).lt.0..or.colout(1,ls).gt.erg)go to 300
go to 290
c
      for neutrons, check the energy and calculate the cosine.
260 if(colout(1,ls).lt.0..or.ntyn.le.17.and.colout(1,ls).gt.erg.and.
1 nxn(lnxs+2,ixex).ne.95242)go to 300
colout(2,ls)=acecos(i)
ixcos=i
c
c      adjust if energy and cosine are given in center-of-mass system.
280 ergace=colout(1,ls)
if(ntyn.ge.0)go to 290
t1=aawn(lawn+ixex)+1.
t2=colout(1,ls)
t3=colout(2,ls)
colout(1,ls)=t2+(erg+2.*t3*t1*sqrt(erg*t2))/t1**2
colout(2,ls)=t3*sqrt(t2/colout(1,ls))+sqrt(erg/colout(1,ls))/t1
c
      resample energy up to 100 times if > emx(1).
290 if(colout(1,ls).le.emx(1))return
nx=nx+1
call errprn(1,netb(2),-1,4,erg,zero,'erg',' ',
1 'energy of neutron from inelastic collision > emx')
if(nx.lt.100)go to 10
call expirx(1,'acecas','erg>emx 100 times in one collision.')
return
c
      print debug information for cross-section table errors.
295 colout(1,ls)=-huge
300 call zaid(2,ht,ixl(1ixl+1,ixex))
*call lkon
      write(iuo,310)ht,erg,ixre,
1 int(xss(jxs(1jxs+3,ixex)+ixre-1)),ntyn,lw,colout(1,ls)

```

```

310 format(/30h error in cross-section table ,a10/12h energy in =,
1 lpe12.4,5x,16hreaction index =,i3,5x,4hmt =,i4,5x,4hty =,i4,
2 5x,5hlaw =,i3,5x,12henergy out =,lpe12.4)
if(colout(1,ls).eq.-huge)call expirx(1,'acecas',
1 'an inappropriate or non-existent law was selected.')
if(colout(1,ls).lt.0.)call expirx(1,'acecas',
1 'energy of neutron from inelastic collision is negative.')
if(colout(1,ls).gt.erg)call expirx(1,'acecas',
1 'energy of emerging neutron exceeds energy of incident neutron')
return
c
c      for photons, calculate the cosine.
320 if(t.ge.elc(2))colout(1,ls)=t
if(jxs(ljxs+16,iex).ne.0)go to 330
colout(2,ls)=2.*rang()-1.
return
330 colout(2,ls)=acecos(i)
ixcos=i
if(i.ne.0)ipsc=8
return
end
*deck tvx
subroutine tgtvelx(ip,e2,r2,ar,vr,k2)
c      sample the velocity of the target nucleus for the neutron free
c      gas thermal treatment.
c      should be used for elastic scattering only.
c      e2=relative energy in rest frame of target motion.
c      r2, k2 are interpolation fraction, index of energy in xss.
*call cm
*call cmx
dimension vr(3)
save c,z
c
c      sample the velocity of the target nucleus.
GO TO (1,80)IP
1 IFLTG=1
ar=awn(lawn+iex)/ttn
ycn=sqrt(erg*ar)
10 if(rang()*(ycn+1.12837917).le.ycn)go to 20
r1=rang()
z2=-log(r1*rang())
go to 30
20 r1=rang()**2
s=r1+rang()**2
if(s.gt.1.)go to 20
z2=-r1*log(s)/s-log(rang())
30 z=sqrt(z2)
c=2.*rang()-1.
x2=ycn**2+z2-2.*ycn*z*c
if((rang()*(ycn+z))**2.gt.x2)go to 10
if(x2.lt.ar*elc(1))go to 10
E2=X2/AR
ETAR=Z2*TTN
CTAR=C
IC=KTC(KKTC+1,IEX)+JXS(LJXS+1,IEX)-1
IF(E2.LE.XSS(IC+1).AND.E2.GE.XSS(IC))GO TO 70
IF(E2.GT.XSS(IC))GO TO 50
40 IF(IC.EQ.JXS(LJXS+1,IEX))GO TO 60
IC=IC-1
IF(E2.LT.XSS(IC))GO TO 40
GO TO 60
50 IF(IC.EQ.JXS(LJXS+1,IEX)+NXS(LNXS+3,IEX)-2)GO TO 60
IC=IC+1
IF(E2.GT.XSS(IC+1))GO TO 50
60 K2=IC-JXS(LJXS+1,IEX)+1
70 IF(NTY(LNTY+IEX).NE.2)R2=
1 MAX(ZERO,MIN(ONE,(E2-XSS(IC))/(XSS(IC+1)-XSS(IC))))
RETURN
c
80 call rotasz(c,uuu,vtr,irt)
vtr(1)=z*vtr(1)
vtr(2)=z*vtr(2)
vtr(3)=z*vtr(3)
c
c      calculate functions of the target velocity.
vr(1)=ycn*uuu-vtr(1)
vr(2)=ycn*vvv-vtr(2)
vr(3)=ycn*www-vtr(3)
ssr=sqrt(vr(1)**2+vr(2)**2+vr(3)**2)
return

```

```

    end
*deck aox
    subroutine acecosx(cosv,cosm,js)
c      sample a cosine from the distribution indicated by the reaction
c      number ixre and the energy erg.  return also k, the index in
c      xss of the first word of the cosine table used.
c      data: ne,(e(i),i=1,ne),(lmu(i),i=1,ne),some(mu(i),i=1,33)
*call cm
    DIMENSION COSV(33),COSM(32)
C
    JS=0
c      ixre = 0 for an elastic collision, .gt. 0 for inelastic.
    l=jxs(ljxs+8*ipt, iex)+ixre-ipt+1
    if(xss(l).eq.0)go to 40
c      find the cosine table by binary search on the energy table.
    ic=jxs(ljxs+8*ipt+1, iex)+xss(l)
    n=xss(ic-1)
    ib=ic-1+n
    10 if(ib-ic.eq.1)go to 30
    ih=(ic+ib)/2
    if(erg.lt.xss(ih))go to 20
    ic=ih
    go to 10
    20 ib=ih
    go to 10
c      sample between adjoining tables by interpolation fraction.
    30 if(rang()*(xss(ib)-xss(ic)).lt.erg-xss(ic))ic=ib
    if(xss(ic+n).eq.0)go to 40
c      sample from table of 32 equiprobable cosine groups.
    k=jxs(ljxs+8*ipt+1, iex)-1+xss(ic+n)
    COSV(1)=XSS(K)
    DO 35 I=2,33
      COSV(I)=XSS(K+I-1)
      COSM(I-1)=0.03125D0/(COSV(I)-COSV(I-1))
    35 CONTINUE
    return
c      isotropic case
    40 COSV(1)=-1.0D0
    DIV=0.0625D0
    DO 45 I=1,32
      COSV(I+1)=COSV(I)+DIV
      COSM(I)=0.5
    45 CONTINUE
    JS=1
    return
end
*deck pld
    subroutine pldirec(pv,cosv,cosm)
c      this subroutine calculates pl components directly for lab bins
c
*call cm
    dimension pv(0:ileg),cosv(33),cosm(32),cos1(ileg,33)
    common/plvs/v(ileg+1)
c      store pl moments directly
    do 20 i=1,33
      v(1)=cosv(i)
      do 10 j=2,mxsc(1)+1
        v(j)=v(j-1)*cosv(i)
        cos1(j-1,i)=pvint(j-1)
    10     continue
    20   continue
    do 30 i=1,mxsc(1)
    30   pv(i)=0.0
      do 50 i=1,32
        do 40 j=1,mxsc(1)
          pv(j)=pv(j)+(cos1(j,i+1)-cos1(j,i))*cosm(i)
    40     continue
    50   continue
    return
end
function pvint(j)
*call cm
    common/plvs/v(ileg+1)
    go to (10,20,30,40,50,60,70)j
    write(iuo,*)'error pv too far'
    stop

```

```

10  pvint=0.5d0*v(2)
    return
20  pvint=0.5d0*v(3)-0.5d0*v(1)
    return
30  pvint=0.625d0*v(4)-0.75d0*v(2)
    return
40  pvint=0.875d0*v(5)-1.25d0*v(3)+0.375d0*v(1)
    return
50  pvint=1.3125d0*v(6)-2.1875d0*v(4)+0.9375d0*v(2)
    return
60  pvint=2.0625d0*v(7)-3.9375d0*v(5)+2.1875d0*v(3)-0.3125d0*v(1)
    return
70  pvint=3.3515625d0*v(8)-7.21875d0*v(6)+4.921875d0*v(4)
$      -1.09375d0*v(2)
    return
end
*deck bin
subroutine binover(bis,cosv,cosm)
*call cm
dimension pv(0:ileg),cosv(33),cosm(32),cos1(ileg,33),bis(nbin)
dimension func(nbin+1)
c
c   this subroutine overlays lab 32 bins onto equi-probable bins
func(1)=-1.0d0
do 10 i=1,msct
    func(i+1)=func(i):bdiv
    bis(i)=0.0
10  continue
c   find initial equi bin that overlaps lab bins
lst=1
do 20 i=1,msct
    if(func(i+1).gt.cosv(i))go to 30
    lst=lst+1
20  continue
30  lpos=1
do 50 i=lst,msct
    area=0.0
40  if(func(i).lt.cosv(lpos))then
        start=cosv(lpos)
    else
        start=func(i)
    endif
    if(func(i+1).le.cosv(lpos+1))then
        finish=func(i+1)
        area=area+(finish-start)*cosm(lpos)
c   store area not height of new bin
        bis(i)=area
    else
        finish=cosv(lpos+1)
        area=area+(finish-start)*cosm(lpos)
        lpos=lpos+1
        if(lpos.eq.33)then
            bis(i)=area
            go to 55
        endif
        go to 40
    endif
50  continue
55  return
end
*deck agx
function e3c(erg,cosv)
*if def,cheap,1
    implicit double precision (a-h,o-z)
*call cmx1
c   calculates lab energy for law 3
e3c=a4+(erg*cosv*a2*a5)/a3
return
end
function ecm(erg,cosv)
*if def,cheap,1
    implicit double precision (a-h,o-z)
*call cmx1
c   calculates the lab energy given initial energy and cm angle
c   for target at rest
    ecm=erg*(a4+a3*cosv)/a5
    return
end
function c3m(erg,e)
*if def,cheap,1

```

```

    implicit double precision (a-h,o-z)
*call cmx1
c calculates the cm angle given final energy, initial energy
c for law 3
c c3m=((e-a4)*a3-erg)/(a2*a5)
    return
end
function ccm(erg,e)
*if def,cheap,1
    implicit double precision (a-h,o-z)
*call cmx1
c calculates the cm angle given final energy, initial energy
c for target at rest
ccm=((e/erg)*a5-a4)/a3
    return
end
function a3m(e,erg,cosv)
*if def,cheap,1
    implicit double precision (a-h,o-z)
*call cmx1
c calculates the lab angle given a cm angle
c for law 3
el=1.0d0/e
a3m=cosv*sqrt(a4*el)+2.0d0*sqrt(erg*el)/a2
    return
end
function acm(cosv)
*if def,cheap,1
    implicit double precision (a-h,o-z)
*call cmx1
c calculates the lab angle given a cm angle
c for target at rest
acm=(a2*cosv+1.)/sqrt(a4+a3*cosv)
    return
end
function ecmm(cosv)
*if def,cheap,1
    implicit double precision (a-h,o-z)
*call cmx1
c calculates the lab energy for a given cm angle for
c target in motion
ecmm=(d2+d1*cosv)/a5
    return
end
function ccmm(e)
*if def,cheap,1
    implicit double precision (a-h,o-z)
*call cmx1
c calculates the cm angle given final energy, initial energy
c for target in motion
ccmm=(a5*e-d2)/(d1)
    return
end
function acmm(cosv,caz)
*if def,cheap,1
    implicit double precision (a-h,o-z)
*call cmx1
c calculates the lab angle given a cm angle
c for target in motion
t1=sqrt(d2+d1*cosv)
t2=d*(1.+cosv*d3)
t3=d4*d5*sqrt(1.-cosv**2)*caz
acmm=(t2-t3)/t1
    return
end
*deck afx
subroutine areafull(lly,cosv,igb)
c this subroutine calculates and store group to group data
c for the case of full within group scattering
*call cm
parameter(thirty=0.03125d0)
*call cmx
*call cmx1
dimension cosv(33),cos1(ileg,32),pv(0:ileg),cosl(32)
c
c special case where all scattering goes into same group
c cycle through all tally bins
il=isp0(lisp+1,(ig-1)*mxsd(1)+igb)
if(il.eq.0)then
    write(iuo,131)ig,igb

```

```

        write(jtty,131)ig,igb
        return
    endif
c     i1=position within scattering matrix
c     i2=position of legendre data or bin data
c     i3=actual position within equi angle bins
c
c     i1=mxsd(1)*6+i1
c     do legendre stuff
c       if(mxsc(1).eq.0)go to 580
c       i2=mxsd(1)*6+ispo(lisp+2,(ig-1)*mxsd(1)+igb)
c       using only one random number for time sake
c       rn=rang()
c       do 540 k=1,i1
c         cos1(k,1)=0.0
540  continue
c       if(l1.ne.3)go to 543
c       do 542 j=1,32
c         cm=cosv(j)+rn*(cosv(j+1)-cosv(j))
c         em=e3c(ergolde,cm)
c         cosl(j)=a3m(em,ergolde,cm)
542  continue
c       go to 547
543  if(ifltg.ne.0)go to 545
c       do 544 j=1,32
c         cm=cosv(j)+rn*(cosv(j+1)-cosv(j))
c         cosl(j)=acm(cm)
544  continue
c       go to 547
545  do 546 j=1,32
c         cm=cosv(j)+rn*(cosv(j+1)-cosv(j))
c         cosl(j)=acmm(cm,cazm)
546  continue
547  if(mxsc(5).eq.1)go to 560
c     store lab angle in equi-angle bins
c     the lab angles go from -1 to 1
c     do 550 j=1,32
c       am=cosl(j)+1.0d0
c       k=am*bdiv1+1
c       if(k.gt.msct)then
c         write(iuo,'*)'WARNING - k > msct'
c       endif
c       i3=i2+k-1
c       cosl(1,j)=i3
c       cosl(2,j)=thirty*w4
550  continue
c     go to 580
560  pv(0)=1.0
c     do 575 j=1,32
c       pv(1)=cosl(j)
c       do 565 k=1,mxsc(1)-1
c         pv(k+1)=((2.*k+1.)*cosl(j)*pv(k)-k*pv(k-1))/(k+1.)
c         cosl(k,1)=cosl(k,1)+pv(k)
565  continue
c       cosl(k,1)=cosl(k,1)+pv(k)
575  continue
c     store scattering stuff
580  do 600 i=1,idatr(lidr+im1)
c       ii=kxsc+idatr(lidr+im1+i)
c       jp=ii+i1-1
c       xscon(jp)=xscon(jp)+w4
c       if(mxsc(1).eq.0)go to 600
c       if(mxsc(5).eq.1)go to 590
c       do 585 j=1,32
c         jp=ii+cosl(1,j)-1
c         xscon(jp)=xscon(jp)+cosl(2,j)
585  continue
c       go to 600
590  do 595 k=1,mxsc(1)
c         jp=jp+mxsd(2)
c         xscon(jp)=xscon(jp)+cosl(k,1)*w4*thirty
595  continue
600  continue
131  format('XSERR: scattering occurred from grp ',i5,' to ',i5,
1 ' but not stored')
        return
end
*deck anx
        subroutine areaint(l1,cosv,cosm,bmuc,tmuc,ip,igb,igt)
c     this subroutine calculates and store group to group data

```

```

c     for the case of group to group scattering
*call cm
parameter(thirty=0.03125d0)
*call cmx
*call cmx1
dimension cosv(33),cos1(ileg,32),pv(0:ileg),cosm(32)
dimension cosc(32),cos1(32),cdl(32)
c
c     cycle through all scattering bins
c
c     mark position of begin of integration
c
c     begin = first value of integration
c     top   = last value of integration
c     last=flag
c
c     do 400 i=1,32
c         if(bmuc.lt.cosv(i+1))then
c             begin=bmuc
c             go to 405
c         endif
400  continue
405  im=i
      if(igb.eq.igt)then
        tml=t muc
        go to 430
      endif
c     determine top cm angle for igb bin
c
425  tp1=xsen(lxsn+ip)
      if(lx.ne.3)go to 426
      tml=c3m(ergolde,tp1)
      go to 430
426  if(ifltg.eq.0)then
      tml=ccm(ergolde,tp1)
    else
      tml=ccmm(tp1)
    endif
c
c     i1=position within scattering matrix
c     i2=position of legendre data or bin data
c     i3=actual position within equi angle bins
c
430  il=ispo(lisp+1,(ig-1)*mxsd(1)+igb)
      if(il.eq.0)then
        write(iuo,131)ig,igb
        write(jtty,131)ig,igb
      endif
      if(il.ne.0)il=mxsd(1)*6+il
      i2=mxsd(1)*6+ispo(lisp+2,(ig-1)*mxsd(1)+igb)
c
c     initiate calculations for this group to group interaction
      sum=0.0
c     using only one random number for time sake
      rn=rang()
      do 431 k=1,i leg
        cos1(k,1)=0.0
431  continue
      ic=0
c
435  if(tml.le.cosv(im+1))then
      top=tml
      last=1
    else
      top=cosv(im+1)
      last=0
    endif
c     skip storage if this group to group not accounted for
      if(il.eq.0)go to 440
      del=top-begin
      dels=del*cosm(im)
      sum=sum+dels
c     do legendre stuff
      if(mxsc(1).eq.0)go to 440
      cm=begin+r n*(top-begin)
      ic=ic+1
      cosc(ic)=cm
      cdl(ic)=dels
440  begin=top
      if(last.eq.0)then

```

```

        im=im+1
        go to 435
    endif
    if(i1.eq.0)go to 605
    if(mxsc(1).eq.0)go to 580
c     calculate lab angles
    if(ly.ne.3)go to 450
    do 445 i=1,ic
        em=e3c(ergolde,cosc(i))
        cosl(i)=a3m(em,ergolde,cosc(i))
445  continue
        go to 455
450  if(ifltg.eq.0)then
        do 452 i=1,ic
            cosl(i)=acm(cosc(i))
452  continue
        else
            do 453 i=1,ic
                cosl(i)=acmm(cosc(i),cazm)
453  continue
        endif
455  if(mxsc(5).eq.1)go to 560
c     store lab angle in equi-angle bins
c     the lab angles go from -1 to 1
    do 456 i=1,ic
        am=cosl(i)+1.0d0
        k=am*bdiv1+1
        if(k.gt.msct)then
            write(iuo,*)'WARNING - k > msct'
        endif
        i3=i2+k-1
        cos1(1,i)=i3
        cos1(2,i)=cdl(i)*w4
456  continue
        go to 580
c     calculate legendre components using delta function
560  do 575 i=1,ic
        pv(0)=1.0
        pv(1)=cosl(i)
        do 565 k=1,mxsc(1)-1
            pv(k+1)=((2.*k+1.)*cosl(i)*pv(k)-k*pv(k-1))/(k+1.)
            cos1(k,1)=cos1(k,1)+pv(k)*cdl(i)
565  continue
        cos1(k,1)=cos1(k,1)+pv(k)*cdl(i)
575  continue
c
c     store scattering stuff
580  do 600 i=1,idatr(lidr+im1)
        ii=kxsc+idatr(lidr+im1+i)
        jp=ii+i1-1
        xscon(jp)=xscon(jp)+w4*sum
        if(mxsc(1).eq.0)go to 600
        if(mxsc(5).eq.1)go to 590
        do 585 j=1,ic
            jp=ii+cos1(1,j)-1
            xscon(jp)=xscon(jp)+cos1(2,j)
585  continue
        go to 600
590  do 595 k=1,mxsc(1)
        jp=jp+mxsd(2)
        xscon(jp)=xscon(jp)+cos1(k,1)*w4
595  continue
600  continue
c
605  if(igb.eq.igt)return
        igb=igb-1
        if(igb.eq.igt)then
            tml=tmu
            go to 430
        endif
        ip=ip+1
        go to 425
131  format('XSERR: scattering occurred from grp ',i5,' to ',i5,
     1 ' but not stored')
        end
*deck sb
        subroutine sabcolx(iflag)
c           calculate exit energy and direction for an s(a,b) collision.
*call cm
*call cmx

```

```

      dimension cos1(ileg,32),pv(0:ileg)
c
c      select elastic or inelastic scattering.
      ipsc=9
      nctyn=0
      ixcos=0
      if(jxs(ljxs+4,iet).eq.0)go to 10
      if(rang()*rtc(krtc+7,iet).gt rtc(krtc+8,iet))go to 60
c
c      get the sampling parameters and energy for the inelastic case.
10 IF(NXS(LNXS+2,IET).EQ.3.AND.IFLAG.EQ.3)GO TO 200
      if(nxs(lnxs+7,iet).ne.0)go to 20
      kr=rang()*nxs(lnxs+4,iet)
      go to 50
20 r=rang()*(nxs(lnxs+4,iet)-3)
      if(r.lt.1.)go to 30
      kr=r+1.
      go to 50
30 if(r.lt..5)go to 40
      kr=nxs(lnxs+4,iet)-2
      if(r.lt..6)kr=kr+1
      go to 50
40 kr=1
      if(r.lt..1)kr=0
50 ll=nxs(lnxs+3,iet)
      ni=nxs(lnxs+4,iet)*(ll+2)
      kx=(kktc(kktc+1,iet)-1)*ni+jxs(ljxs+3,iet)+kr*(ll+2)
      erg=xss(kx)+rtc(krtc+1,iet)*(xss(kx+ni)-xss(kx))
      if(erg.lt.elc(1))return
      if(erg.lt.1.e-11)erg=1.e-11
      kx=kx+1
      ri=rtc(krtc+1,iet)
      ntny=1
      go to(140,140,70,80)nxs(lnxs+2,iet)+1
c
c      set up the sampling parameters for the elastic case.
60 kx=jxs(ljxs+6,iet)+(kktc(kktc+2,iet)-1)*(abs(nxs(lnxs+6,iet))+1)
      ri=rtc(krtc+4,iet)
      ni=nxs(lnxs+6,iet)+1
      ll=nxs(lnxs+6,iet)
      ntny=2
      go to(140,140,70,80,90)nxs(lnxs+5,iet)+1
c
c >>>>  equally-probable angle bins.
70 if(rang().le.ri)kx=kx+ni
      f=ll*rang()+1.
      j=kx+f
      cs=xss(j)+(xss(j-1)-xss(j))*(f-aint(f))
      ixcos=-kx
      go to 130
c
c >>>>  equally-probable discrete angles.
80 j=kx+(ll+1)*rang()
      cs=xss(j)+ri*(xss(j+ni)-xss(j))
      ixcos=kx
      go to 130
c
c >>>>  exact treatment of coherent elastic scattering
90 ic=jxs(ljxs+5,iet)-1
      ib=jxs(ljxs+5,iet)+kktc(kktc+2,iet)-1
      if(ib.eq.ic+1)go to 120
      pr=xss(ib)*rang()
100 if(ib-ic.eq.1)go to 120
      ih=(ic+ib)/2
      if(pr.lt.xss(ih))go to 110
      ic=ih
      go to 100
110 ib=ih
      go to 100
120 cs=1.-2.*xss(ib-int(xss(jxs(ljxs+4,iet))))/erg
c
c      sample the azimuthal angle.
130 call rotas(cs)
      return
140 call expirx(1,'sabcol','inappropriate distribution required.')
      return
c
c      do storage of data for multigroup cross sections
200 IFLAG=-3
      LL=NXS(LNXS+3,IET)
      NI=NXS(LNXS+4,IET)*(LL+2)

```

```

      NJ=NXS(LNXS+4,IET)
      AJ=NXS(LNXS+4,IET)
C     LOCATE STARTING POSITION OF DATA
      KX=(KTC(KKTC+1,IET)-1)*NI+JXS(LJXS+3,IET)
C     CYCLE THROUGH ENERGIES AND ANGLES TO STORE DATA
      IC=0
      FM=1.0D0/AJ
      IF(NXS(LNXS+7,IET).NE.0)FM1=1.0D0/(AJ-3.0D0)
      DO 250 I=1,NJ
         IF(NXS(LNXS+7,IET).NE.0)THEN
            IF(I.EQ.1.OR.I.EQ.NJ)THEN
               FM=0.1*FM1
            ELSE IF(I.EQ.2.OR.I.EQ.(NJ-1))THEN
               FM=0.4*FM1
            ELSE
               FM=1.0*FM1
            ENDIF
         ENDIF
         E1=XSS(KX+IC)+RTC(KRTC+1,IET)*(XSS(KX+NI+IC)-XSS(KX+IC))
         IF(E1.LT.XSEN(LXSN+1))THEN
            IC=IC+LL+2
            GO TO 250
         ENDIF
C     DETERMINE ENERGY GROUP
         DO 210 J=2,IXSN+1
            IF(E1.LT.XSEN(LXSN+J))GO TO 215
            IC=IC+LL+2
            GO TO 250
215      IGB=J-1
            IGB=IXSN+1-IGB
C     STORE CROSS SECTION DATA
            I1=ISPO(LISP+1,(IG-1)*MXSD(1)+IGB)
            IF(I1.EQ.0)THEN
               WRITE(IUO,131)IG,IGB
               WRITE(JTTY,131)IG,IGB
               IC=IC+LL+2
               GO TO 250
            ENDIF
            I1=MXSD(1)*6+I1
            DO 220 J=1,IDAIR(LIDR+IM1)
               II=KXSC+IDAIR(LIDR+IM1+J)
               JP=II+I1-1
               XSCON(JP)=XSCON(JP)+W4*FM
220      CONTINUE
            IC=IC+1
C     STORE PL IF REQUESTED
            IF(MXSC(1).EQ.0)THEN
               IC=IC+LL+1
               GO TO 250
            ENDIF
            AA=LL+1
            FM2=FM/AA
            I2=MXSD(1)*6+ISPO(LISP+2,(IG-1)*MXSD(1)+IGB)
            DO 222 J=1,MXSC(1)
               COS1(J,1)=0.0
222      DO 226 J=1,LL+1
               AM=XSS(KX+IC)+RTC(KRTC+1,IET)*(XSS(KX+NI+IC)-XSS(KX+IC))
               IF(MXSC(5).EQ.1)GO TO 223
C     STORE LAB ANGLE IN EQUI-ANGLE BINS
C     THE LAB ANGLES GO FROM -1 TO 1
               AM=AM+1.0
               K=AM*BDIV1+1
               IF(K.GT.MSCT)THEN
                  WRITE(IUO,'*)'WARNING - K > MSCT'
               ENDIF
               I3=I2+K-1
               COS1(1,J)=I3
               COS1(2,J)=FM2*W4
               GO TO 225
C     CALCULATE LEGENDRE COMPONENTS USING DELTA FUNCTION
223      PV(0)=ONE
               PV(1)=AM
               DO 224 K=1,MXSC(1)-1
                  PV(K+1)=((2.*K+1.)*AM*PV(K)-K*PV(K-1))/(K+1.)
                  COS1(K,1)=COS1(K,1)+PV(K)*FM2
224      CONTINUE
               COS1(K,1)=COS1(K,1)+PV(K)*FM2
C
225      IC=IC+1
226      CONTINUE

```

```

C      STORE PL DATA
      DO 230 J=1, IDATR(LIDR+IM1)
          II=KXSC+IDATR(LIDR+IM1+J)
          IF(MXSC(5).EQ.1)GO TO 228
          DO 227 K=1,LL+1
              JP=II+COS1(1,K)-1
              XSCON(JP)=XSCON(JP)+COS1(2,K)
227      CONTINUE
          GO TO 230
228      JP=II+I1-1
          DO 229 K=1,MXSC(1)
              JP=JP+MXSD(2)
              XSCON(JP)=XSCON(JP)+COS1(K,1)*W4
229      CONTINUE
230      CONTINUE
250      CONTINUE
      RETURN
131  FORMAT('XSERR: SCATTERING OCCURRED FROM GRP ',I5,' TO ',I5,
1   ' BUT NOT STORED')
      end

```