ANL-7112
RETURN TO IDAHO LIBRARY.

ANL-7112

Argonne National Laboratory

XLIBIT: ANL Cross-section Library Code

by

S. D. Sparck

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

Printed in USA. Price $7.00 . Available from the Clearinghouse for Federal Scientific and Technical Information, National Bureau of Standards, U. S. Department of Commerce, Springfield, Virginia 22151

ANL-7112
Mathematics and Computers
(TID-4500)
AEC Research and
Development Report

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

XLIBIT: ANL Cross-section Library Code

by

S. D. Sparck

Applied Mathematics Division

February 1966

Operated by The University of Chicago
under
Contract W-31-109-eng-38
with the
U. S. Atomic Energy Commission

TABLE OF CONTENTS

		Page
I.	INTRODUCTION	5
II.	CROSS-SECTION NOMENCLATURE	6
III.	DESCRIPTION OF CROSS-SECTION INPUT DECKS	10
IV.	SPECIFIC CARD TYPE DESCRIPTIONS	14
	A. Type-plus (+) Card	14
	B. Type-0 Cards	14
	C. Type-1 Cards	16
	D. Type-2 Cards	17
	E. Type-3 Cards (Elastic Scattering)	19
	F. Type-4 Cards (Elastic Scattering)	21
	G. Type-5 Cards (Inelastic Scattering)	24
	H. Type-6 Cards (Inelastic Scattering)	24
	I. Type-7 Cards [(n,2n) Scattering]	25
	J. Type-8 Cards [(n,2n) Scattering]	25
V.	PROGRAM OPERATION	26
	A. List of Control Cards	26
	B. General Rules and Restrictions	27
	C. Description of Individual Control Cards	28
	D. Structure of the Input Deck	33
	E. Input-Output Units Used by XLIBIT	37
VI.	LIST OF ERROR RETURNS	38
VII.	STRUCTURE OF THE CROSS-SECTION LIBRARY TAPE PREPARED BY THE XLIBIT PROGRAM	49
	A. Ordering of the Files Comprising the Library Tape	49
	B. Description of the Directory of Sets File	50
	C. Description of a Cross-section Set File	51
	D. Structure of a Material Block, by Material Identification,	53
	Chi, Group Constants, and Scattering-data Records	55
	E. Description of the Material Identification Record of a Block	56
	F. Description of the Chi Record of a Material Block	57
	G. Description of a Group-constants Record of a Block	59
	H. Description of a Scattering-data Record	61
APPENDIXES		
	A. Sample XLIBIT INPUT FORMS	61
	B. List of XLIBIT Subroutines and Common Block Assignments	69
	C. Flow Charts and Descriptions	75
ACKNOWLEDGMENTS		304

XLIBIT: ANL Cross-section Library Code

by

S. D. Sparck

I. INTRODUCTION

With the acquisition by the Laboratory of a Control Data 3600 computer in the fall of 1963, a joint reprogramming project was undertaken by the Laboratory's Applied Mathematics Division and the Control Data Corporation to develop one- and two-dimensional diffusion-theory and S_n transport-theory programs for the CDC-3600. The XLIBIT code described in this report was developed as a part of this project.

XLIBIT is a CDC-3600 program designed to:

1) Prepare a cross-section library tape.

2) Modify an existing cross-section library tape.

3) Duplicate an existing cross-section library tape.*

4) Print all or a portion of the cross-section data on the library tape.

5) Punch cross-section decks for all or a portion of the data on the library tape. The decks produced are identical to those used in preparing the tape.**

The cross-section library tape is composed of one or more cross-section sets. A cross-section set is defined as the collection of the relevant parameters for the materials assigned a particular energy-angle structure. One or more materials must be included in each set, and each may be considered an isotope or a mixture of isotopes or pure materials, as convenient.

---

*See Part V, Section D, paragraph 3 (p. 34).

**Following the last data card, a BCD card containing "RUN COMPLETED." in columns 5 through 18 is punched to mark termination of the punch output for a run.

II. CROSS-SECTION NOMENCLATURE

The following definitions have been employed in specifying the parameters included in the cross-section set. To avoid confusion with the summation sign, σ is used for cross sections throughout. This is not meant to imply any restriction on the type of cross section used. The user must, of course, designate these corresponding to his use of densities or volume fractions in the problem specifications.

For each material and each energy (lethargy) group j, the following are included:

D_j = diffusion coefficient. If required by code and not specified by user, this will be calculated as $1/3\ \sigma_{tr,j}$.

$\sigma_{c,j}$ = the capture cross section, which includes (n,α), (n,p), (n,γ) losses.

ν_j = number of neutrons emitted per fission.

$\sigma_{f,j}$ = fission cross section.

$\sigma_{tr,j}$ = transport cross section, defined as total cross section for transport codes. If required by code and not specified by user, this will be calculated as the total cross section.

$\chi(j,k)$ = fission-spectrum fractions for group k caused by neutrons in group j.

Three types of transfer may be included: elastic, inelastic, and (n,2n). Each of these transfer cross sections may have angular dependence expressed as a truncated Legendre series.

$\sigma_{el,j}(\mu)$ = the total elastic-transfer cross section from group j, as a function of the cosine of the scattering angle μ in the laboratory system

$$= \sum_{\ell=0}^{N} \frac{2\ell + 1}{4\pi} \sigma_{el,j}^{(\ell)} P_\ell(\mu) = \sum_{\ell=0}^{N} \sum_{k=1}^{J} \frac{2\ell + 1}{4\pi} \sigma_{el}^{(\ell)}(j \to k) P_\ell(\mu).$$

If

$T_{el}^{(\ell)}(j \to k)$ = the elastic-transfer vector for component ℓ representing transfer from group j to group k, i.e., the jth row in each of the N-plus-1 scattering matrices, each element in this row gives the fraction transferred to group k, and

$\sigma_{el,j}^{(\ell)} \cdot T_{el}^{(\ell)}(j \to k) = \sigma_{el}^{(\ell)}(j \to k)$, the elastic-transfer cross section for component ℓ representing transfer from group j to group k.

The same form is used if the inelastic and (n,2n) transfer cross sections are given.

$\sigma_{in,j}(\mu)$ = the total inelastic-scattering cross section

$$= \sum_{\ell=0}^{N'} \frac{2\ell + 1}{4\pi} \sigma_{in,j}^{(\ell)} P_\ell(\mu) = \sum_{\ell=0}^{N'} \sum_{k=1}^{J} \frac{2\ell + 1}{4\pi} \sigma_{in}^{(\ell)}(j \to k) P_\ell(\mu).$$

$T_{in}^{(\ell)}(j \to k)$ = the inelastic-scattering vector for component ℓ representing transfer from group j to group k,

and $\sigma_{in,j}^{(\ell)} \cdot T_{in}^{(\ell)}(j \to k) = \sigma_{in}^{(\ell)}(j \to k)$, the inelastic-transfer cross section.

$\sigma_{(n,2n),j}(\mu)$ = the (n,2n) cross section

$$= \sum_{\ell=0}^{N''} \frac{2\ell + 1}{4\pi} \sigma_{(n,2n),j}^{(\ell)} P_\ell(\mu) = \sum_{\ell=0}^{N''} \sum_{k=1}^{J} \frac{2\ell + 1}{4\pi} \sigma_{(n,2n)}^{(\ell)} (j \to k) P_\ell(\mu).$$

$T_{(n,2n)}^{(\ell)}(j \to k)$ = the (n,2n)-reaction transfer vector representing transfer from group j to group k, and

$\sigma_{(n,2n),j}^{(\ell)} \cdot T_{(n,2n)}^{(\ell)}(j \to k) = \sigma_{(n,2n)}^{(\ell)}(j \to k)$,

the (n,2n) transfer cross section.

A distinct pair of upper and lower bounds (i.e., number of groups into which up- and downtransfer is to be considered) must be specified for each of the three types of transfer used.

In addition, the following data are required (these data are material-independent but as a convenience may be given with each material):

v_j = the mean velocity of group j,

E_j or u_j = the minimum energy of group j, or maximum lethargy of group j. J plus 1 entries for j = 0, 1, 2, ..., J are required.

For applications to S_n-type transport-theory codes, the various scattering-type cross sections can be combined into two classes: one, scattering that removes the neutron from the group $\sigma_{j \neq k}(j \to k, \mu)$; and two, scattering that leaves the neutron in the group $\sigma(j \to j, \mu)$. In terms of the above nomenclature,

$$\sigma_{j\neq k}(j \to k,\mu) = \left\{ \sum_{\substack{k=1 \\ k\neq j}}^{J} \frac{1}{2} \sum_{\ell=0}^{N''} \frac{2\ell+1}{4\pi} \sigma^{(\ell)}_{(n,2n),j} \cdot T^{(\ell)}_{(n,2n)}(j \to k) P_\ell(\mu) \right.$$

$$+ \sum_{\ell=0}^{N'} \frac{2\ell+1}{4\pi} \sigma^{(\ell)}_{in,j} \cdot T^{(\ell)}_{in}(j \to k) P_\ell(\mu)$$

$$\left. + \sum_{\ell=0}^{N} \frac{2\ell+1}{4\pi} \sigma^{(\ell)}_{el,j} \cdot T^{(\ell)}_{el}(j \to k) P_\ell(\mu) \right\},$$

and

$$\sigma(j \to j,\mu) = \frac{1}{2} \sum_{\ell=0}^{N''} \frac{2\ell+1}{4\pi} \sigma^{(\ell)}_{(n,2n),j} \cdot T^{(\ell)}_{(n,2n)}(j \to j) P_\ell(\mu)$$

$$+ \sum_{\ell=0}^{N'} \frac{2\ell+1}{4\pi} \sigma^{(\ell)}_{in,j} \cdot T^{(\ell)}_{in}(j \to j) P_\ell(\mu)$$

$$+ \sum_{\ell=0}^{N} \frac{2\ell+1}{4\pi} \sigma^{(\ell)}_{el,j} \cdot T^{(\ell)}_{el}(j \to j) P_\ell(\mu).$$

For diffusion-theory codes, the above expressions become

$$\sigma_{j\neq k}(j \to k) = \sum_{\substack{k=1 \\ k\neq j}}^{J} \left\{ \frac{1}{2} \sigma^{(0)}_{(n,2n),j} \cdot T^{(0)}_{(n,2n)}(j \to k) + \sigma^{(0)}_{in,j} T^{(0)}_{in}(j \to k) \right.$$

$$\left. + \sigma^{(0)}_{el,j} T^{(0)}_{el}(j \to k) \right\},$$

and

$$\sigma(j \to j) = \frac{1}{2} \sigma^{(0)}_{(n,2n),j} \cdot T^{(0)}_{(n,2n)}(j \to j) + \sigma^{(0)}_{in,j} T^{(0)}_{in}(j \to j)$$

$$+ \sigma^{(0)}_{el,j} T^{(0)}_{el}(j \to j).$$

The total cross section, $\sigma_{t,j}$, is then

$$\sigma_{t,j} = \sigma_{c,j} + \sigma_{f,j} + \sum_{k=1}^{J} \sigma(j \to k).$$

If the scattering matrices specified are normalized such that

$$\sum_{k=1}^{J} T_{el}^{(0)}(j \to k) = 1,$$

$$\sum_{k=1}^{J} T_{in}^{(0)}(j \to k) = 1,$$

and

$$\sum_{k=1}^{J} T_{(n,2n)}^{(0)}(j \to k) = 2;$$

then

$$\sigma_{t,j} = \sigma_{c,j} + \sigma_{f,j} + \sigma_{(n,2n),j}^{(0)} + \sigma_{in,j}^{(0)} + \sigma_{el,j}^{(0)}.$$

III. DESCRIPTION OF CROSS-SECTION INPUT DECKS

The cross-section input deck, which is the basic unit of XLIBIT input, is the series of cards used to specify the relevant parameters for a single material (see Fig. 1).

The cross-section input decks for all materials included in a cross-section set have an identical energy-group structure.

Provision is made for ten card types in each input deck.* These types are designated as +, 0, 1, 2, ..., 8, to identify their respective contents.

Type (+), an optional card, is provided to accommodate the user who wishes to personally identify a material.

Type (0) defines the group structure of the cross-section set.

Type (1) contains the group constants D_j, $\sigma_{tr,j}$, $\sigma_{c,j}$, $\sigma_{f,j}$, and ν_j.

Type (2) specifies the fission-spectrum fractions.

Types (3) through (8) contain the transfer cross-section data.

The values on the ANL cross-section data cards are either numeric quantities or Hollerith (BCD) quantities. The general rules given below should be followed in filling out the data cards:

1. Hollerith quantities (i.e., set and material identification and options) may contain any valid Hollerith character except the period (.) and the comma (,).

2. Numeric fields of 12 columns are considered to contain floating-point quantities. If these quantities are written in the FORTRAN exponential format with the scientific notation of fraction and exponent (for example, 5.63E01), they must be right-justified.

3. Numeric fields of one or three columns are considered to contain integer quantities. These quantities must be right-justified.

4. Numeric fields left blank are treated as zero, unless specifically stated otherwise.

Columns 1 through 20 of all types of cross-section data cards, except the + card, have the same format. The following rules apply to the quantities appearing in columns 1 through 20:

1. Set Name (columns 1-5)

   The set name must be given on all data cards. All cards for an individual set must be placed together. No two sets on the same library tape may have the same name.

---

*A sample set of forms available for use in keypunching cross-section data is included as Appendix A.

SET NAME				MAT. NAME				SP. CH.	TYPE	GROUP NO.		CARD NO.			ℓ-NO.		
1			5	6				11	12	13	14	16	17		19	20	21 . . 25 26 . . .
A	1	3	C	U	2	3	4		$	+	0	4	8				ANL-5800
A	1	3	C	U	2	3	4		$	0		0			1		
A	1	3	C	U	2	3	4		$	0		1			2		
A	1	3	C	U	2	3	4		$	0		2			3		
A	1	3	C	U	2	3	4		$	0		3			4		
A	1	3	C	U	2	3	4		$	1		1			5		
A	1	3	C	U	2	3	4		$	1		2			6		
A	1	3	C	U	2	3	4		$	1		3			7		
A	1	3	C	U	2	3	4		$	2		1			8		
A	1	3	C	U	2	3	4		$	2		1			9		
A	1	3	C	U	2	3	4		$	2		2		1	0		.
A	1	3	C	U	2	3	4		$	2		2		1	1		
A	1	3	C	U	2	3	4		$	2		3		1	2		.
A	1	3	C	U	2	3	4		$	2		3		1	3		
A	1	3	C	U	2	3	4		$	3		1		1	4	0	.
A	1	3	C	U	2	3	4		$	3		1		1	5	1	
A	1	3	C	U	2	3	4		$	3		1		1	6	2	D
A	1	3	C	U	2	3	4		$	3		2		1	7	0	
A	1	3	C	U	2	3	4		$	3		2		1	8	1	A
A	1	3	C	U	2	3	4		$	3		2		1	9	2	
A	1	3	C	U	2	3	4		$	3		3		2	0	0	T
A	1	3	C	U	2	3	4		$	3		3		2	1	1	
A	1	3	C	U	2	3	4		$	3		3		2	2	2	A
A	1	3	C	U	2	3	4		$	4		1		2	3	0	
A	1	3	C	U	2	3	4		$	4		1		2	4	1	.
A	1	3	C	U	2	3	4		$	4		1		2	5	2	
A	1	3	C	U	2	3	4		$	4		2		2	6	0	.
A	1	3	C	U	2	3	4		$	4		2		2	7	1	
A	1	3	C	U	2	3	4		$	4		2		2	8	2	.
A	1	3	C	U	2	3	4		$	4		3		2	9	0	
A	1	3	C	U	2	3	4		$	4		3		3	0	1	
A	1	3	C	U	2	3	4		$	4		3		3	1	2	
A	1	3	C	U	2	3	4		$	7		1		3	2	0	
A	1	3	C	U	2	3	4		$	7		2		3	3	0	
A	1	3	C	U	2	3	4		$	7		3		3	4	0	
A	1	3	C	U	2	3	4		$	8		1		3	5	0	
A	1	3	C	U	2	3	4		$	8		2		3	6	0	
A	1	3	C	U	2	3	4		$	8		3		3	7	0	

NOTE: The card-numbering system shown above (columns 17-19) is not the only one possible. Numbering may, for example, start with each card type, or with each group for the same card type, or with each group and ℓ-number combination for the same card type, etc.

Fig. 1. Cross-section Input Deck for a Material with Three Energy Groups

2. Material Name (columns 6-11)

   The material name must be given on all data cards. All cards --- individual material within a set must be placed together. No two materials in the same set may have the same name.

   The code does not consider the special character (see paragraph 3 below) to be part of the material name.

3. Special Character (column 12)

   The special character is intended for use in special applications to convey additional information about a material (e.g., to denote hydrogenous or heavy materials).

   The program reads the special character from the type-0 card, and disregards column 12 on cards of all other types. This quantity may be a blank or any other Hollerith character except a period or a comma.

4. Card Type (column 13)

   The card type must be given on all data cards used. The card type must be a plus (+) or one of the integers 0 through 8; the arrangement of data cards for an individual material within a set is by increasing card-type number following the initial + card.

5. Group Number (columns 14-16)

   The group number must be given on all data cards used. It must be an integer, and must be right-justified in columns 14-16.

   A group number of zero appears on the type-0 card to allow specification of a maximum energy for the highest energy group.

   In addition, XLIBIT uses the convention of a zero group number on the type-2 cards to indicate a vector (rather than a matrix) fission spectrum.

   The arrangement of data cards of a single card type for a material is by increasing group number.

   NOTE: Group 1 is the group of highest energy.

6. Card Number (columns 17-19)

   The card number is an integer taking on the values n, n + 1, ..., and must be right-justified in columns 17-19. It is used as a sequence number.

   The program ignores columns 17-19 on cards of types other than 2, 4, 6, and 8.

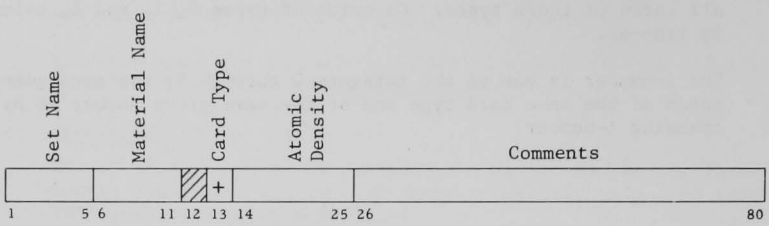
7. <u>ℓ-number</u> (column 20)

The ℓ-number is used only on cards of types 3 through 8 to designate the order of the anisotropy. This is an integer and must be given on all cards of these types. On cards of types 0, 1, and 2, column 20 is ignored.

The ℓ-number is one of the integers 0 through 9; the arrangement of cards of the same card type and of the same group number is by increasing ℓ-number.

IV. SPECIFIC CARD TYPE DESCRIPTIONS

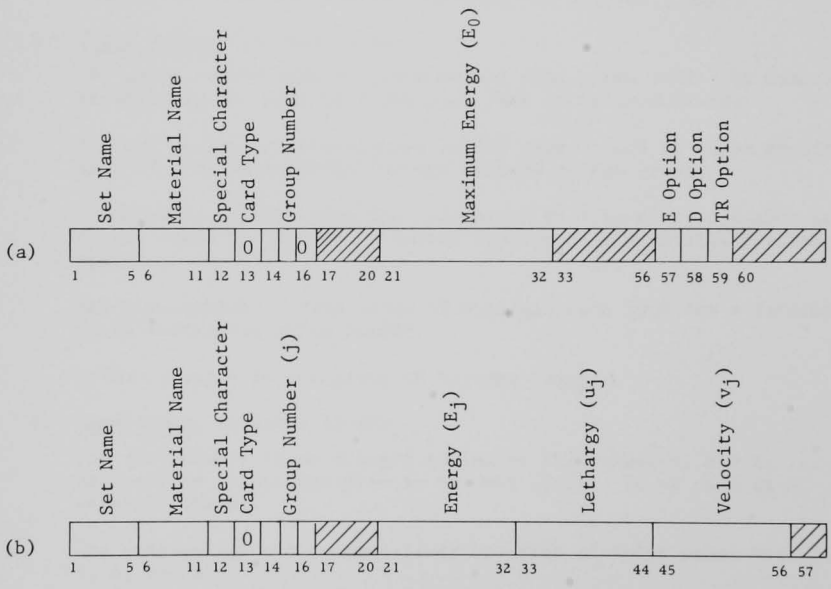
A. Type-plus (+) Card



At most one type-plus card may be used for a material. This card is optional and need not be used; if it is used, however, it must be the first data card for a material (i.e., precede the first type-0 card).

Columns 14 through 25 have been set aside for the atomic density of the material. Columns 26-80 may contain free-field information using any character in the Hollerith character set.

B. Type-0 Cards



1. There must be exactly J + 1 type-0 cards in the data deck for one material, where J is the number of energy groups, arranged in order of increasing group number.

2. On the card having group number 0 [diagram (a) above], a nonzero, nonblank energy value and the E, D, and TR options must be given.

3. On the cards having group numbers 1, 2, ..., J [diagram (b) above] XLIBIT requires that the velocity values either all be blank (or zero), or all be given nonzero values. If the velocities are blank or zero, XLIBIT will assign for all velocities the value 1.0.

4. If the E option is blank, the energy values must be given on the type-0 cards for groups 1, 2, ..., J. XLIBIT will use these values to compute the corresponding lethargies. If a zero energy value is given, XLIBIT assigns 10^{50} as the corresponding lethargy. If the E option is nonblank, the lethargy values must be given on the type-0 cards for groups 1, 2, ..., J. XLIBIT will use these values to compute the corresponding energies.

5. If the D option is blank, the diffusion coefficients must be given on the type-1 cards.

   If the D option is nonblank, the code will calculate the diffusion coefficients as $1/(3\sigma_{tr})$ (where the σ_{tr} used is the value given on each type-1 card).

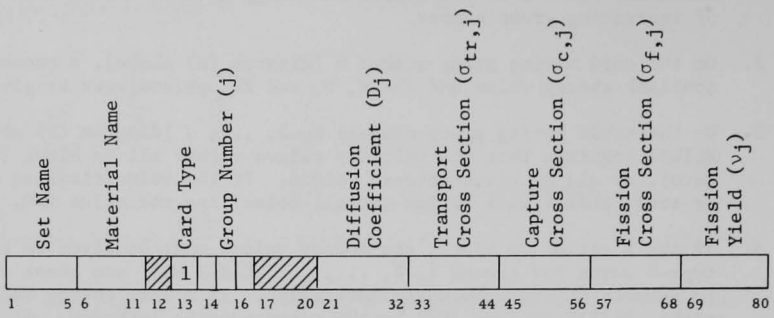
   NOTE: If the D-option is nonblank and σ_{tr} is zero or blank, then a value of 10^{50} will be assigned the corresponding diffusion coefficient.

6. If the TR option is blank, the transport cross sections must be given on each type-1 card.

   If the TR option is nonblank, XLIBIT will calculate the transport cross sections from the other given cross sections.

7. If both the D and TR options are nonblank, XLIBIT will first calculate σ_{tr} (as in 6 above); and will then calculate the diffusion coefficients (as in 5 above) by using the σ_{tr} just calculated.

C. Type-1 Cards



1. There must be exactly J type-1 cards in the data deck for one material; the cards are to be arranged in order of increasing group number.

2. If the TR option is nonblank, any transport cross section given will be ignored by the program; the transport cross sections will be calculated as $\sigma_{t,j}$, the total cross section (see page 8).

3. If the D option is nonblank, any diffusion coefficient given will be ignored by the program; the diffusion coefficients will be calculated as $D_j = 1/(3\sigma_{tr,j})$.

   If the transport cross section is left zero or blank on the type-1 cards, the diffusion coefficients will be given the value 10^{50}.

4. XLIBIT will assume that a material is fissionable if there is at least one nonzero, nonblank entry for a fission cross section ($\sigma_{f,j}$) or fission yield (ν_j).

   Conversely, XLIBIT will assume a material to be nonfissionable if <u>all</u> entries for $\sigma_{f,j}$ and ν_j are blank or zero.

IDA

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

ADDENDA for ANL-7112

XLIBIT: ANL Cross-section Library Code

by

S. D. Sparck

The following are changes to the XLIBIT program as it is described in the report. The changes are listed in chronological order. All references to page and paragraph numbers refer to the report.

1. Pages 86-87, paragraphs 9 and 10.
   Subroutine ADDAMAT.

   If a discrepancy leading to an ERFLAG value of 105 or 106 is encountered processing continues as shown, but in addition the following messages are printed:

   ERFLAG = 105/106, ENERGY/LETHARGY/VELOCITY VALUES FOR SET (name) MATERIAL (name) DO NOT AGREE WITH PREVIOUS VALUES GIVEN

   WITH THESE VALUES HAVING BEEN NOTED, PROCESSING OF THE SET WILL CONTINUE.

2. Pages 226-227, paragraph 1.
   Subroutine RDSTOR2.

   Between the call to GREAD and the three-way branch test on BFLAG the following test is inserted: If COL1 = 1H* or if TYPE = 1H+ or 1H0, then a return is made immediately from RDSTOR2; else the three-way test on BFLAG is made as shown on page 227. The new test was inserted to allow materials without cards of types 3 through 8 (i.e., without scattering) to be included and inserted in any position among the materials of a set.

Alvin Glassner

Alvin Glassner
Technical Publications Department

February 21, 1967

IDA

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

ADDENDA for ANL-7112

XLIBIT: ANL Cross-section Library Code

by

S. D. Sparck

The following are changes to the XLIBIT program as it is described in the report. The changes are listed in chronological order. All references to page and paragraph numbers refer to the report.

1. Pages 86-87, paragraphs 9 and 10.
   Subroutine ADDAMAT.

   If a discrepancy leading to an ERFLAG value of 105 or 106 is encountered processing continues as shown, but in addition the following messages are printed:

   ERFLAG = 105/106, ENERGY/LETHARGY/VELOCITY VALUES FOR SET (name) MATERIAL (name) DO NOT AGREE WITH PREVIOUS VALUES GIVEN

   WITH THESE VALUES HAVING BEEN NOTED, PROCESSING OF THE SET WILL CONTINUE.

2. Pages 226-227, paragraph 1.
   Subroutine RDSTOR2.

   Between the call to GREAD and the three-way branch test on BFLAG the following test is inserted: If COL1 = 1H* or if TYPE = 1H+ or 1H0, then a return is made immediately from RDSTOR2; else the three-way test on BFLAG is made as shown on page 227. The new test was inserted to allow materials without cards of types 3 through 8 (i.e., without scattering) to be included and inserted in any position among the materials of a set.

Alvin Glassner
Technical Publications Department

February 21, 1967

IDA

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

ADDENDA for ANL-7112

XLIBIT: ANL Cross-section Library Code

by

S. D. Sparck

The following are changes to the XLIBIT program as it is described in the report. The changes are listed in chronological order. All references to page and paragraph numbers refer to the report.

1. Pages 86-87, paragraphs 9 and 10.
   Subroutine ADDAMAT.

   If a discrepancy leading to an ERFLAG value of 105 or 106 is encountered processing continues as shown, but in addition the following messages are printed:

   ERFLAG = 105/106, ENERGY/LETHARGY/VELOCITY VALUES FOR SET (name) MATERIAL (name) DO NOT AGREE WITH PREVIOUS VALUES GIVEN

   WITH THESE VALUES HAVING BEEN NOTED, PROCESSING OF THE SET WILL CONTINUE.

2. Pages 226-227, paragraph 1.
   Subroutine RDSTOR2.

   Between the call to GREAD and the three-way branch test on BFLAG the following test is inserted: If COL1 = 1H* or if TYPE = 1H+ or 1H0, then a return is made immediately from RDSTOR2; else the three-way test on BFLAG is made as shown on page 227. The new test was inserted to allow materials without cards of types 3 through 8 (i.e., without scattering) to be included and inserted in any position among the materials of a set.

Alvin Glassner
Technical Publications Department

February 21, 1967

IDA

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

ADDENDA for ANL-7112

XLIBIT: ANL Cross-section Library Code

by

S. D. Sparck

The following are changes to the XLIBIT program as it is described in the report. The changes are listed in chronological order. All references to page and paragraph numbers refer to the report.

1. Pages 86-87, paragraphs 9 and 10.
   Subroutine ADDAMAT.

   If a discrepancy leading to an ERFLAG value of 105 or 106 is encountered processing continues as shown, but in addition the following messages are printed:

   ERFLAG = 105/106, ENERGY/LETHARGY/VELOCITY VALUES FOR SET (name) MATERIAL (name) DO NOT AGREE WITH PREVIOUS VALUES GIVEN

   WITH THESE VALUES HAVING BEEN NOTED, PROCESSING OF THE SET WILL CONTINUE.

2. Pages 226-227, paragraph 1.
   Subroutine RDSTOR2.

   Between the call to GREAD and the three-way branch test on BFLAG the following test is inserted: If COL1 = 1H* or if TYPE = 1H+ or 1H0, then a return is made immediately from RDSTOR2; else the three-way test on BFLAG is made as shown on page 227. The new test was inserted to allow materials without cards of types 3 through 8 (i.e., without scattering) to be included and inserted in any position among the materials of a set.

Alvin Glassner
Alvin Glassner
Technical Publications Department

February 21, 1967

MAR 15 1967
ASSOC. DIRECTOR'S OFFICE
ID-ANL

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

ADDENDA for ANL-7112

XLIBIT: ANL Cross-section Library Code

by

S. D. Sparck

The following are changes to the XLIBIT program as it is described in the report. The changes are listed in chronological order. All references to page and paragraph numbers refer to the report.

1. Pages 86-87, paragraphs 9 and 10.
   Subroutine ADDAMAT.

   If a discrepancy leading to an ERFLAG value of 105 or 106 is encountered processing continues as shown, but in addition the following messages are printed:

   ERFLAG = 105/106, ENERGY/LETHARGY/VELOCITY VALUES FOR SET (name) MATERIAL (name) DO NOT AGREE WITH PREVIOUS VALUES GIVEN

   WITH THESE VALUES HAVING BEEN NOTED, PROCESSING OF THE SET WILL CONTINUE.

2. Pages 226-227, paragraph 1.
   Subroutine RDSTOR2.

   Between the call to GREAD and the three-way branch test on BFLAG the following test is inserted: If COL1 = 1H* or if TYPE = 1H+ or 1H0, then a return is made immediately from RDSTOR2; else the three-way test on BFLAG is made as shown on page 227. The new test was inserted to allow materials without cards of types 3 through 8 (i.e., without scattering) to be included and inserted in any position among the materials of a set.

Alvin Glassner
Technical Publications Department

February 21, 1967

MAR 1 5 1967
ASSOC. DIRECTOR'S
OFFICE
ID-ANL

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

ADDENDA for ANL-7112

XLIBIT: ANL Cross-section Library Code

by

S. D. Sparck

The following are changes to the XLIBIT program as it is described in the report. The changes are listed in chronological order. All references to page and paragraph numbers refer to the report.

1. Pages 86-87, paragraphs 9 and 10.
   Subroutine ADDAMAT.

   If a discrepancy leading to an ERFLAG value of 105 or 106 is encountered processing continues as shown, but in addition the following messages are printed:

   ERFLAG = 105/106, ENERGY/LETHARGY/VELOCITY VALUES FOR SET (name) MATERIAL (name) DO NOT AGREE WITH PREVIOUS VALUES GIVEN

   WITH THESE VALUES HAVING BEEN NOTED, PROCESSING OF THE SET WILL CONTINUE.

2. Pages 226-227, paragraph 1.
   Subroutine RDSTOR2.

   Between the call to GREAD and the three-way branch test on BFLAG the following test is inserted: If COL1 = 1H* or if TYPE = 1H+ or 1H0, then a return is made immediately from RDSTOR2; else the three-way test on BFLAG is made as shown on page 227. The new test was inserted to allow materials without cards of types 3 through 8 (i.e., without scattering) to be included and inserted in any position among the materials of a set.

Alvin Glassner

Alvin Glassner
Technical Publications Department

February 21, 1967

MAR 1 3 1967
ASSOC. DIRECTOR'S
OFFICE
ID-ANL

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

ADDENDA for ANL-7112

XLIBIT: ANL Cross-section Library Code

by

S. D. Sparck

The following are changes to the XLIBIT program as it is described in the report. The changes are listed in chronological order. All references to page and paragraph numbers refer to the report.

1. Pages 86-87, paragraphs 9 and 10.
   Subroutine ADDAMAT.

   If a discrepancy leading to an ERFLAG value of 105 or 106 is encountered processing continues as shown, but in addition the following messages are printed:

   ERFLAG = 105/106, ENERGY/LETHARGY/VELOCITY VALUES FOR SET (name) MATERIAL (name) DO NOT AGREE WITH PREVIOUS VALUES GIVEN

   WITH THESE VALUES HAVING BEEN NOTED, PROCESSING OF THE SET WILL CONTINUE.

2. Pages 226-227, paragraph 1.
   Subroutine RDSTOR2.

   Between the call to GREAD and the three-way branch test on BFLAG the following test is inserted: If COL1 = 1H* or if TYPE = 1H+ or 1H0, then a return is made immediately from RDSTOR2; else the three-way test on BFLAG is made as shown on page 227. The new test was inserted to allow materials without cards of types 3 through 8 (i.e., without scattering) to be included and inserted in any position among the materials of a set.

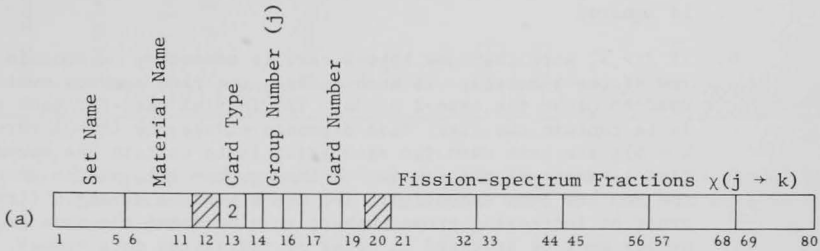
Alvin Glassner

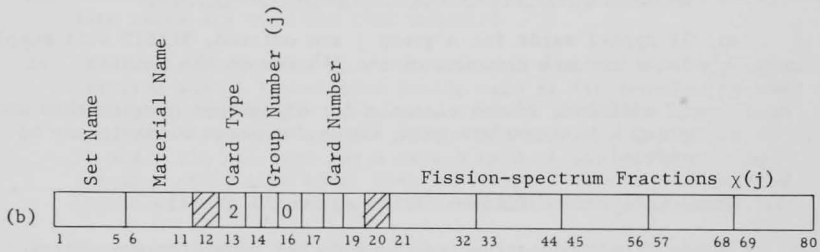
Alvin Glassner
Technical Publications Department

February 21, 1967

D. Type-2 Cards

1. The type-2 cards for a material give the fission-spectrum fraction matrix $[\chi(j,k)]$. An arbitrary element of the matrix, $\chi(j,k)$, gives the probability that a neutron with incident energy in group j will yield a neutron whose energy lies in energy group k.





2. Type-2 cards are required for a fissionable material.

   If a material is nonfissionable, any type-2 cards present are ignored by the program.

   The program distinguishes between a fissionable and a nonfissionable material by examining the values given to the fission cross section and the fission yield (ν) on the type-1 cards.

3. If type-2 cards are used, they must be arranged in order of increasing group number. If more than one card is required for a group number, the cards for the group number must be arranged by increasing card number. (See also paragraphs 4 and 5 below.)

4. Group-dependent fission [see diagram (a) above]. The χ-matrix is arranged as follows:

	k = 1	k = 2	k = 3. . .	. k = J
j = 1	$X_{1\to1}$	$X_{1\to2}$	$X_{1\to3}$	$X_{1\to J}$
j = 2	$X_{2\to1}$	$X_{2\to2}$	$X_{2\to3}$	$X_{2\to J}$
j = 3	$X_{3\to1}$	$X_{3\to2}$	$X_{3\to3}$	$X_{3\to J}$
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
j = J	$X_{J\to1}$	$X_{J\to2}$	$X_{J\to3}$	$X_{J\to J}$

A single type-2 card may contain up to five consecutive eleme one row, i.e., five consecutive values of k for one value of group number j.

a. If $J \leq 5$, one row of the matrix (all k-values for a single j-value) will fit on one card. Thus, only one type-2 card for each group number is required, and in this case the card number is ignored.

b. If $J > 5$, more than one type-2 card is necessary to contain one row of the χ-matrix. In such a case, the card numbers must be used to order the type-2 cards. The initial card for each group is to contain the first five elements of the row ($k = 1$ through $k = 5$); the next card for each group is to contain the second five elements of the row ($k = 6$ through $k = 10$), and so on until the row has been exhausted. The cards must be arranged first in order of increasing group number; then cards of the same group number must be arranged in order of increasing card number. The group number must be repeated on all such cards.

c. If type-2 cards for a group j are omitted, XLIBIT will supply zeros for all elements of the jth row of the matrix.

   In addition, if the elements for all groups greater than some group k in a row are zero, cards for these elements may be omitted.

5. Group-independent fission [see diagram (b) above].

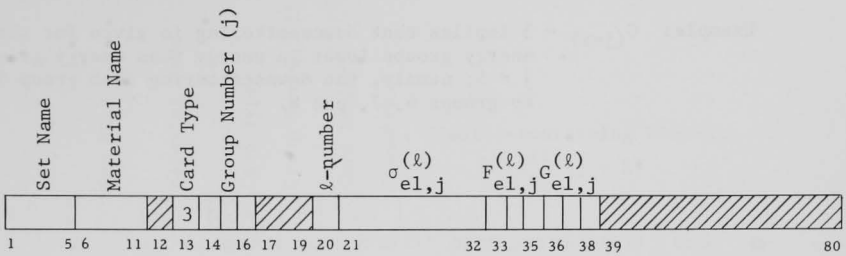
If the fission-spectrum fractions do not depend on the energy j of the incident neutron, instead of a matrix, a χ-vector [$\chi(k)$] of J elements is given as shown below:

	$k = 1$	$k = 2$	$k = 3$	$k = 4 \ldots$	$\ldots k = J$
(for all j)	χ_1	χ_2	χ_3	χ_4	χ_J

If this group-independent option is used, the type-2 cards have the following form: The group number must be set to zero on all type-2 cards for the material. The elements χ_k of the χ-vector are contained five/card in the same manner as are the elements in a single row of a χ-matrix: i.e., $k = 1$ through $k = 5$ are contained on the first card with group 0; $k = 6$ through $k = 10$ are contained on the next card with group 0, etc.

If the elements for all groups greater than some group k are zero, the cards for these elements may be omitted.

E. Type-3 Cards (Elastic Scattering)



1. The group number must be given on each type-3 card. If any of the quantities ℓ, σ, F, or G on a type-3 card is blank, it will be interpreted as zero.

2a. If elastic scattering is not considered for a material, no type-3 data cards are used for that material.

2b. If elastic scattering is considered for a material, the program requires that a type-3 card having zero as its ℓ-number be used for each group number j = 1, 2, ..., J, even if the σ, F, and G on some of these cards are all zero. If, however, the values of σ, F, and G are all zero for a type-3 card of any group with an ℓ-number other than zero, then that type-3 card may be omitted (zeros will be assumed for the σ, F, and G in such a case).

2c. The type-3 cards for a material must be arranged in order of increasing group number. Type-3 cards having the same group number must be arranged in order of increasing ℓ-number.

3. $\sigma^{(\ell)}_{el,j}$ = the coefficient used in the ℓth term of the Legendre polynomial expansion for the elastic-scattering cross section for group j.

$F^{(\ell)}_{el,j}$ = the maximum number of groups into which elastic-upscattering cross sections from group j are given.

$G^{(\ell)}_{el,j}$ = the maximum number of groups into which elastic-downscattering cross sections from group j are given.

The superscript ℓ used with F and G specifies the "order" of the scattering. An ℓ-value of zero indicates isotropic scattering; an ℓ-value greater than zero indicates anisotropic scattering components. Values of ℓ through ℓ = 9 are possible. For each ℓ-value, the following three transfer matrices may exist:

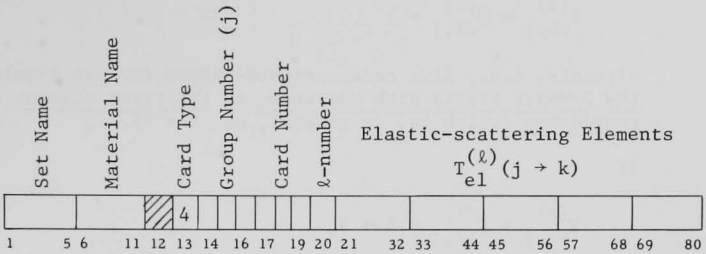
$$\left[T^{(\ell)}_{elastic}\right], \quad \left[T^{(\ell)}_{inelastic}\right], \quad \text{and} \quad \left[T^{(\ell)}_{(n,2n)}\right].$$

(See the description of the data cards of type-4, type-6 and type-8 in paragraphs F, H, and J below.)

4. The value of F or G must be used relative to the value of j that is associated with the F or G.

   Example: $G_{(j=5)} = 3$ implies that downscattering is given for three energy groups lower in energy than energy group j = 5; namely, the downscattering from group 5 to groups 6, 7, and 8.

F. Type-4 Cards (Elastic Scattering)



1. The type-4 cards for a material are arranged first in order of increasing group number. Cards having the same group number are arranged in order of increasing ℓ-number. Cards having the same group number and ℓ-number are arranged in order of increasing card number.

2. The program requires that the number of type-4 cards depend on the values of F and G on the various type-3 cards, in the following ways:

   a. If no type-3 card is used for a given combination of group number (j) and ℓ-number (ℓ), then the type-4 card for this combination of j and ℓ may be omitted. If included, it will be ignored.

   b. If

   $$F_{el,j}^{(\ell)} = G_{el,j}^{(\ell)} = 0$$

   for a type-3 card bearing group number (j) and ℓ-number (ℓ), then the type-4 card for this combination of j and ℓ may be omitted. If included, it will be ignored.

   c. If either

   $$F_{el,j}^{(\ell)} \text{ or } G_{el,j}^{(\ell)}$$

   is nonzero for a type-3 card having a given pair of j and ℓ values, then at least one type-4 card for this combination of j and ℓ <u>must</u> be used.

   In this case, the type-4 cards for the given combination of j and ℓ values are to contain

   $$F_{el,j}^{(\ell)} + G_{el,j}^{(\ell)} + 1$$

   elements of the jth row of the matrix

   $$\left[T_{el}^{(\ell)}(j \rightarrow k)\right].$$

d. Each type-4 card may contain up to five of the string of

$$F^{(\ell)}_{el,j} + G^{(\ell)}_{el,j} + 1$$

elements, i.e., five consecutive k-values for one j-value, where the k-value starts with the index of the first element to be considered (which is, in general, not k = 1).

If

$$F^{(\ell)}_{el,j} + G^{(\ell)}_{el,j} + 1 \leq 5,$$

then the elements under consideration for row j of this matrix are contained on one type-4 card. In this case, the card numbers are ignored.

If

$$F^{(\ell)}_{el,j} + G^{(\ell)}_{el,j} + 1 > 5,$$

then card numbers are used to order the cards. The first card will contain the first five elements, the next card the next five, and so on, until the string of elements of row j of this matrix is exhausted. The group number and ℓ-number must be repeated on all such cards.

3. The type-4 cards for a material contain certain elements of the elastic-scattering matrix

$$\left[T^{(\ell)}_{el}(j \rightarrow k)\right].$$

There may be as many as 10 such matrices, one each for ℓ = 0(1)9. Each matrix may have as many as J*J elements, arranged as follows:

	k = 1	k = 2	k = 3 . . .	k = J
j = 1	$T^{\ell}_{1\rightarrow 1}$	$T^{\ell}_{1\rightarrow 2}$	$T^{\ell}_{1\rightarrow 3}$ . . .	$T^{\ell}_{1\rightarrow J}$
j = 2	$T^{\ell}_{2\rightarrow 1}$	$T^{\ell}_{2\rightarrow 2}$	$T^{\ell}_{2\rightarrow 3}$	$T^{\ell}_{2\rightarrow J}$
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
j = J	$T^{\ell}_{J\rightarrow 1}$	$T^{\ell}_{J\rightarrow 2}$	$T^{\ell}_{J\rightarrow 3}$	$T^{\ell}_{J\rightarrow J}$

The diagonal elements in the matrix represent self-scattering terms. Elements to the left of the diagonal represent scattering to groups of lower group numbers; these are the upscattering terms. Similarly, elements to the right of the diagonal represent scattering to groups of higher group number; these are the downscattering terms.

Corresponding to each of the rows of each elastic-scattering matrix

$$\left[T_{el}^{(\ell)}(j \to k)\right]$$

are the two quantities

$$F_{el,j}^{(\ell)}$$

and

$$G_{el,j}^{(\ell)},$$

giving the maximum number of groups into which up- and downscattering, respectively, will be considered in this (j^{th}) row of the $T_{el}^{(\ell)}$ matrix.

4. For example, suppose that the total number of groups, J, is 6. Suppose further that

$$F_{el,4}^{(\ell)} = 1,\ G_{el,4}^{(\ell)} = 2;\quad F_{el,5}^{(\ell)} = G_{el,5}^{(\ell)} = 0;\quad F_{el,6}^{(\ell)} = 1,\ G_{el,6}^{(\ell)} = 0.$$

This implies that in the fourth row of the matrix $F_4 + G_4 + 1 = 4$; and thus four elements will be considered. The fifth row, since only self-scattering is being considered, is ignored. In this case columns 21-32 of the type-3 card for group 5 and this ℓ-number contain the self-scattering cross section, $\sigma_{el}^{(\ell)}(5 \to 5)$, and the corresponding type-4 card may be omitted. In the sixth row,

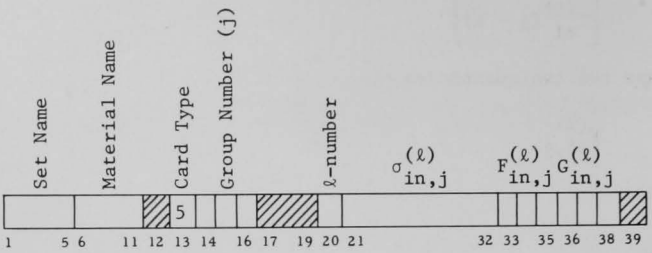
$$F_6 + G_6 + 1 = 2,$$

and two elements are considered.

The elements of these rows of the matrix given on type-4 cards are as follows:

	k = 1	k = 2	k = 3	k = 4	k = 5	k = 6
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
j = 4	.	.	$T_{4\to3}^{\ell}$	$T_{4\to4}^{\ell}$	$T_{4\to5}^{\ell}$	$T_{4\to6}^{\ell}$
j = 5	.	.	.	.	.	.
j = 6	.	.	.	.	$T_{6\to5}^{\ell}$	$T_{6\to6}^{\ell}$

G. Type-5 Cards (Inelastic Scattering)



1. $\sigma^{(\ell)}_{in,j}$ = the coefficient used in the ℓth term of the Legendre polynomial expansion for the inelastic-scattering cross section for group j.

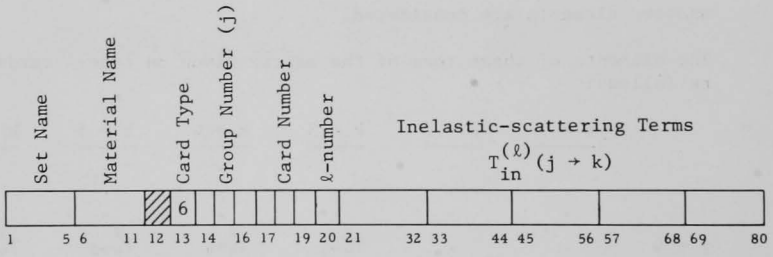
   $F^{(\ell)}_{in,j}$ = the maximum number of groups into which inelastic-upscattering cross sections from group j are given.

   $G^{(\ell)}_{in,j}$ = the maximum number of groups into which inelastic-downscattering cross sections from group j are given.

2. The function and description of the type-5 cards are identical to those of the type-3 cards, except that the type-3 cards refer to <u>elastic</u> scattering; the type-5 cards refer to <u>inelastic</u> scattering.

   The type-5 cards bear the same relationship to the type-6 cards, for inelastic scattering, as do the type-3 cards to the type-4 cards for elastic scattering.

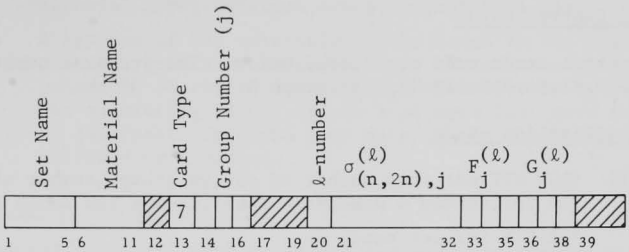
H. Type-6 Cards (Inelastic Scattering)



1. The type-6 cards for a material are arranged first in order of increasing group number. Cards having the same group number are arranged in order of increasing ℓ-number. Cards having the same group number and ℓ-number are arranged in order of increasing card number.

2. The type-6 cards bear the same relationship to the type-5 cards for inelastic scattering, as do the type-4 cards to the type-3 cards for elastic scattering.

I. Type-7 Cards [(n,2n) Scattering]



1. $\sigma^{(\ell)}_{(n,2n),j}$ = a component used in the ℓth term of the Legendre polynomial expansion for the (n,2n) scattering cross section for group j.

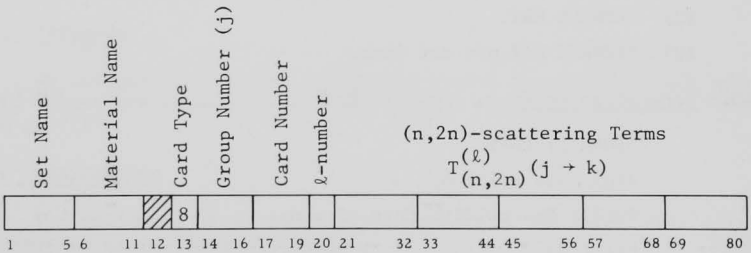
   $F^{(\ell)}_{(n,2n),j}$ = the maximum number of groups into which (n,2n) upscattering cross sections from group j are given.

   $G^{(\ell)}_{(n,2n),j}$ = the maximum number of groups into which (n,2n) downscattering cross sections from group j are given.

2. The function and description of the type-7 cards are identical to those of the type-3 cards, except where the type-3 cards refer to the elastic scattering, the type-7 cards refer to (n,2n) scattering.

   The type-7 cards bear the same relationship to the type-8 cards for (n,2n) scattering, as do the type-3 cards to the type-4 cards for elastic scattering.

J. Type-8 Cards [(n,2n) Scattering]



1. The type-8 cards for a material are arranged first in order of increasing group number. Cards having the same group number are arranged in order of increasing ℓ-number. Cards having the same group number and ℓ-number are arranged in order of increasing card number.

2. The type-8 cards bear the same relationship to the type-7 cards for (n,2n) scattering, as do the type-4 cards to the type-3 cards for elastic scattering.

V. PROGRAM OPERATION

A. List of Control Cards

The control cards used are listed below. The prefixed numbers (1) and (2) are explained following paragraph 3.

1. Modification cards

(1) *ADD SET,set name,number of energy groups,number of materials in set,author's name,date submitted.

*DELETE SET,set name.

*MOD SET,set name,submitter's name,date submitted.

(1) *ADD MAT,number of materials to be added to set.

(2) *DELETE MAT,number of materials to be deleted from set.

*REORDER,set name,new position (e.g., 5 for fifth) of set in library.

2. Extraction Cards

*PRINT SET DIRECTORY.

*PRINT SET,set name.

*PRINT MAT DIRECTORY,set name.

*PRINT MAT DIRECTORY,ALL SETS.

(2) *PRINT MAT,set name.

*PUNCH SET,set name.

*RENAME SET,new set name.

(2) *PUNCH MAT,set name.

(2) *RENAME MAT.

(2) *RENAME MAT,new set name.

3. Separation Cards

*FORM LIBRARY.

*FORM NEW EDITION.

*WRITE NEW EDITION,date of run.

*FINISH.

(1) Cross-section data cards for one or more materials must follow these cards.

(2) A list card or cards (see 4 below) must follow these cards.

4. List Cards

*material $name_1$,material $name_2$,...,material $name_n$.

A maximum of 100 materials may be named in the list. If more than one list card is required to contain the list, then column 80 of one card is followed immediately by column 2 of the succeeding card. Column 1 of each list card must contain an asterisk. Only the last material in the entire list is followed by a period.

B. General Rules and Restrictions

The control cards and list cards are divided into fields; a field consists of those columns bounded by an asterisk (*) and a comma, an asterisk and a period, two commas, or a comma and a period. The fields may contain quantities of three types: a command (e.g., ADD SET,PRINT MAT), which is always in the first field, starting in column 2; a Hollerith quantity (set name,material name,author,date); or an integer (e.g., number of energy groups, new position of set).

1. Hollerith Quantities

Suppose that a maximum of N characters are allowed for a Hollerith quantity (e.g., five for a set name, eight for the author's name). In this case, the N leftmost characters in the appropriate field will be used for this quantity. If the field is less than N characters in length, Hollerith blanks will be appended to make a total of N characters; if the field is greater than N characters in length, only the leftmost N characters will be used.

NOTE: The five-character set name and/or the six-character material name given on a control card must be identical, character by character, to the set name and material name used on the cross-section data cards for the material.

2. Integers

An integer need not be positioned within its field in any way. However, blanks are deleted (e.g., the integer field "...,b1b2bb,..." will be be read as "12").

3. Limitations

The following is a list of limitations on the quantities appearing on the control cards and list cards:

a. The characters period (.) and comma (,) may not be used in a Hollerith quantity.

A set name may consist of from one through five Hollerith characters.

A material name may consist of from one through six Hollerith characters.

The author's name or submitter's name may consist of fr through eight Hollerith characters.

A date may consist of from one through eight Hollerith characters.

b. The maximum allowable number of energy groups for a material is 101.

c. A limit of 300 has been placed on the number of sets that may be contained on one library tape.

Therefore the new position of a set in the library may be no greater than 300.

d. A limit of 500 has been placed on the number of materials in a set.

A list card, or the series of list cards, containing a single material list, may contain no more than 100 material names.

Therefore the number of materials to be deleted from a set may be no greater than 100. Also, the number of materials to be added to a set may be (a) no greater than 100, and (b) such that the total number of materials in the set is not greater than 500.

NOTE: The limitations of paragraph d apply only when using the *ADD MAT or *DELETE MAT cards. Use of the *DELETE SET card will delete an entire set, even if it contains more than 100 materials. Likewise, the *ADD SET card may be followed by the cross-section data cards for more than 100 (but not more than 500) materials.

C. Description of Individual Control Cards

This section describes the functions of each type of control card.

1. *ADD SET,set name,number of energy groups,number of materials, author,date.

This card must be followed by the cross-section data cards for one cross-section set. The set name on the data cards, the number of materials for which data cards are supplied, and the number of energy groups specified on the data cards must all conform to the information given on the ADD SET card. Any discrepancies will result in the run being aborted.

Use of the ADD SET card will place the single cross-section set for which data cards are supplied on the cross-section library tape. The set will be placed behind those sets that are already on the library tape at the time the ADD SET card is encountered. The set name must differ from all set names already in use on the cross-section tape at the time the ADD SET card is encountered; otherwise the run is aborted.

In both ADD SET and DELETE SET cards are used for the same set name, the ADD SET card must follow the DELETE SET card.

2. *DELETE SET,set name.

   Use of the DELETE SET card will cause the set whose name is given to be deleted from the library tape. If there is no set on the library tape corresponding to the set name given on the name DELETE SET card, the run is aborted; no new tape is formed.

   If both DELETE SET and ADD SET cards are used for the same set name, the DELETE SET card must precede the ADD SET card.

3. *MOD SET,set name,submitter,date.

   This card indicates that the set whose name is given is to be modified by the addition of new materials to the set, the deletion of existing materials in the set, or both. Note that to change the values of individual quantities within a material in a set, it is necessary first to delete the material from the set and then to add the material to the set, with the addition to contain the altered values.

   If there is no set name on the library tape like the set name given on the MOD SET card, the run is aborted.

   The MOD SET card must be followed by one or both of the cards DELETE MAT and ADD MAT. If both DELETE MAT and ADD MAT are used, the DELETE MAT card must precede the ADD MAT card.

   The order of appearance of a set on the library tape is not altered if the set is modified via the MOD SET operation.

4. *ADD MAT,number of materials to be added.

   This card must be preceded by a MOD SET card and followed by the cross-section data cards for a number of materials equal to the number given on the ADD MAT card. The set name on all data cards must be the same as the set name given on the preceding MOD SET card. If both a DELETE MAT and an ADD MAT card are used with a MOD SET card, the ADD MAT card must follow the DELETE MAT card. If there are any discrepancies between the data cards and either the ADD MAT or MOD SET control cards, then the entire run is aborted; no new version of the cross-section library tape is formed.

   Use of the ADD MAT card results in the materials for which data cards are supplied being added to the set whose name is given on the MOD SET card. The names of the materials to be added must all be different, and none of them may duplicate the name of a material already present in the set. (Note that different sets may contain the same material name.) Any discrepancies will result in the run being aborted. No new tape is formed.

5. *DELETE MAT,number of materials to be deleted.

   This card must be preceded by a MOD SET card.

   The DELETE MAT card must be followed by one or more list cards on which are given the names of the materials to be deleted from the set. The number of material names in the list must equal the number given on the DELETE MAT card. The names in the list must all be different, and each must be present in the set to be modified. In case of any discrepancies, the run is aborted; no new library tape will be formed.

   Use of the DELETE MAT card causes the materials whose names are given on the following list cards to be deleted from the set whose name is given on the preceding MOD SET card.

6. *REORDER,set name,new position on library tape.

   The set name used on this card must be present on the library tape at the time this card is encountered. (This includes the possibility of reordering a set that has just been added to the tape.)

   Use of this card will place a set in a specified position (e.g., fifth) among the cross-section sets on the tape.

   If the number given as the "new position" is greater than the number of sets on the library tapes, the set named on the REORDER card will be repositioned as the last set in the library.

   Note that subsequent REORDER cards repositioning other sets can partially undo the effect of earlier REORDER cards.

   Example: Suppose that after the command *REORDER,A,3. is carried out, set B is added to the library (automatically placing it last) and is then repositioned by *REORDER,B,2. This will have the effect of making B the second set on the tape. The "old" second set becomes the third, set A now becomes the fourth set on the tape, and so on. Note that the work of *REORDER,A,3. has been partially undone.

   Note also that two or more REORDER cards may specify the same position. In the example, the card *REORDER,B,3. could have been used.

   It is allowable to use two or more REORDER cards specifying the same set name. This will result in two or more distinct repositionings of the set.

7. *PRINT SET DIRECTORY.

   Use of this card causes a list of the set names of those sets on the library tape to be printed, in the order in which the sets appear on the cross-section library tape.

8. *PRINT SET,set name.

Use of this card causes the cross-section data for all materials in the specified set to be printed.

9a. *PRINT MAT DIRECTORY,set name.

Use of this card causes a list of the materials contained in the set whose name is given on the card to be printed, in the order in which the materials appear within the set.

9b. *PRINT MAT DIRECTORY,ALL SETS.

Use of this card will produce the same result as that of paragraph 9a, but for all cross-section sets on the library tape, in the order of the occurrence of the sets on the tape.

10. *PRINT MAT,set name.

Use of this card will cause the cross-section data for a list of specified materials in the specified set to be printed.

This card must be followed by one or more list cards on which are given the material names of the materials to be printed.

11. *PUNCH SET,set name.

Use of this card causes the cross-section data for all materials in this specified set to be punched on 80-column cards.

12. *RENAME SET,new set name.

This card must be used immediately following a PUNCH SET card (but note that a PUNCH SET card may be used without being followed by a RENAME SET card).

Use of this card causes the 80-column data cards produced by the PUNCH SET card to contain the new set name given on the RENAME card (and not the set name given on the PUNCH card).

13. *PUNCH MAT,set name.

Use of this card causes the cross-section data for a list of specified materials in the specified set to be punched on 80-column cards.

This card must be followed by one or more list cards on which are given the material names of the materials to be punched. The names in the list must all be different and must all be present among the material names of the specified set.

14a. *RENAME MAT.

This card must be used immediately following a PUNCH MAT card and its accompanying list card(s) (note that a PUNCH MAT card may be used without being followed by a RENAME MAT card).

This card must be followed by one or more list cards on which are given a number of new material names. The number of new names on the list card(s) must be the same as the number of names given on the list card(s) following the PUNCH MAT card.

The new material names must all be different among themselves. They may be the same as, or may differ from, the names on the list card(s) following the PUNCH MAT card.

Use of this card causes the 80-column data cards produced by the PUNCH MAT card to contain the new material names (and not the material names given on the list card following the PUNCH MAT card).

14b. *RENAME MAT,new set name.

The description, use, and effect of this card are the same as the RENAME MAT card of paragraph 14a, with the following addition: the data cards produced will bear the new set name given on this card instead of the set name given on the PUNCH MAT card.

15a. *FORM LIBRARY.

Either the FORM LIBRARY card or the FORM NEW EDITION card is used--never both. Whichever card is used may appear only once in a control-card deck.

The use of the FORM LIBRARY card implies that there is no existing "old" library tape to be modified, and that a library tape is to be constructed only from the data cards that appear later in this input deck. (It follows that the DELETE SET, MOD SET, DELETE MAT, and ADD MAT modification cards, which make reference to an existing library, have no meaning when used in a control-card deck that includes the FORM LIBRARY card. If found, they cause the run to be aborted.)

Only ADD SET and REORDER modifications cards should follow the FORM LIBRARY card.

15b. *FORM NEW EDITION.

The use of this card implies the presence of an existing "old" library tape which is to be updated by the following modification cards (of all kinds) and cross-section data cards. Contrast this to the description of the FORM LIBRARY card.

16. *WRITE NEW EDITION,date of run.

This card, if used, may not appear more than once in an input deck. The card is used to mark the end of that section of the control-card deck containing the modification cards (and their attendant list cards and cross-section data cards). It must be used if either the FORM LIBRARY or FORM NEW EDITION card is used; if neither of these cards is used (and if this is the case, the input deck consists only of extraction cards), then the WRITE NEW EDITION card must not be used.

17. *FINISH.

This card marks the end of the input deck. It must appear as the last control card of the deck, and must be followed by two blank cards.

D. Structure of the Input Deck

The various commands to XLIBIT as defined by input control cards are processed individually. The execution of these commands is independent of the order in which the commands occur, subject only to the rules of formation of the input deck itself.

The commands processed first are those that edit (i.e., extract information from) an existing library tape. Next are processed commands that form a new library tape. Last, any commands that edit the newly formed tape are processed.

The "edit" commands refer to some or all materials within a particular set on the tape. For each such command the tape being edited is positioned to the set in question; a record is kept of the current position of the tape to facilitate subsequent repositionings. The first file on the library tape is a directory of the sets on the tape. Within each cross-section set is a directory of the materials in the set.

In the formation of a new tape, the following procedure is used: new sets, and sets existing on an "old" library tape that are to be modified, are written as one file each on two scratch tapes. One tape contains the first three records of such sets (the specifications, group structure, and material directory records); the other tape contains the material blocks of these sets (i.e., all other records of the sets). Sets to be carried over unchanged from an "old" library are left on the old library tape. A table (named SETLOC) is used to keep track of the locations of the sets that are to form a new tape: if SETLOC(I) is zero, the Ith set is to be found on the old library tape (and the set directory of the old tape is used to locate the set), if SETLOC(I) has a nonzero value N, then the Ith set exists as the Nth file on each of the two scratch tapes. (The index I refers to the sets as they are to appear on the new tape.) The sets are retrieved in order of increasing I and written on a single tape; this is the new library tape.

1. An input deck is divided into sections according to one of lowing two configurations:

```
                                        (1) { extraction card
                                              .
                                              .
                                              .
                                              extraction card

    (2) { *FORM LIBRARY.                  (2) { *FORM NEW EDITION.
          modification card                     modification card
                .                                     .
                .                                     .
                .                                     .
          modification card                     modification card
        *WRITE NEW EDITION,date.              *WRITE NEW EDITION,date.

    (3) { extraction card                 (3) { extraction card
                .                                     .
                .                                     .
                .                                     .
          extraction card                       extraction card

        *FINISH.                              *FINISH.
         (two blank cards)                     (two blank cards)
```

All extraction cards must be placed either before the FORM... card or after the WRITE... card, i.e., in area (1) or (3). Any extraction cards placed in area (2) will cause the entire run to be aborted. No new tape will be formed.

Similarly, all modification cards must be placed in area (2) between the FORM... and WRITE... cards. Any modification card placed in section (1) will cause the entire run to be aborted. Any modification card placed in section (3) will cause the run to stop at that point; subsequent extraction cards in section (3) will not be processed.

2. The section (1) extraction cards refer to the "old," unmodified library tape; the printed output or the cards produced will contain data from the "old" library. An input deck may contain a section (1) only if the FORM NEW EDITION card is used. If the FORM LIBRARY card is used, it must be the first card in the input deck; no section (1) extraction cards may precede it (see also Section C above, paragraphs 15a and 15b).

3. The section (2) modification cards contain either the data to form a new library tape or changes to be made to an existing library. In the latter case a "new," modified library tape will be formed from the "old" existing library, the new library incorporating the changes indicated by the modification cards. If a FORM NEW EDITION card (and not FORM LIBRARY) is the first card of section (2), then the modification cards may be omitted. Section (2) in this case will consist merely of the two cards FORM NEW EDITION and WRITE NEW EDITION. The effect will be to prepare a duplicate copy of the existing library tape.

Note that the formation of an updated version of (or a duplicate copy of) an existing library tape does not imply the destruction of the existing library. At the conclusion of such a procedure, two libraries exist on two separate tapes: the "old" library and the "new," modified library (or a copy of the "old" library).

4. The section (3) extraction cards refer to the "new," modified library tape; the printed output or the cards produced will contain data from this "new" tape. An input deck may have a section (3) only if it also contains a section (2) to generate a new version of the library tape.

5. An input deck may consist of the following combinations of sections:

   a. (1) only — (This deck prints and/or punches portions of the old library tape, only.)

   b. (2) only — (This deck forms a new tape, or a copy of the old one, only.)

   c. (1) and (2) only — (This deck extracts portions of the old library and forms a new library tape.)

   d. (2) and (3) only — (This deck forms a new library tape and then extracts portions of it.)

   e. (1), (2), and (3) — (This deck extracts portions of an old library, forms a new library tape or a copy of the old one, and extracts portions of the new tape.)

   Note that section (2) includes not only modification cards but also the two separation cards *FORM... and *WRITE..., which must be the first and last cards of section (2).

6. The following rules apply to the modification control cards.

   a. For a given set name, only one of each kind of modification card except REORDER may be used.

   b. If a MOD SET card is used with a set name, neither an ADD SET nor a DELETE SET card can be used for that set name.

      Conversely, if an ADD SET card or a DELETE SET card is used with a set name, the MOD SET card cannot be used for that set name.

   c. Both an ADD SET card and a DELETE SET card may be used with the same set name. If both are used, the DELETE SET card must precede the ADD SET card.

   d. If a MOD SET card is used, it must be followed by at least one of the cards DELETE MAT or ADD MAT.

Conversely, the DELETE MAT and ADD MAT cards can be us they are preceded by a MOD SET card.

At most one DELETE MAT card and one ADD MAT card can be used with a MOD SET card.

If both DELETE MAT and ADD MAT cards are used following a MOD SET card, the DELETE MAT card must precede the ADD MAT card.

e. If the number of materials in a set is zero after the DELETE MAT and ADD MAT cards are processed for the set, an error is flagged and the run is aborted.

f. If a FORM NEW EDITION card is used to initiate the modification section, all types of modification cards may be used following it, subject to the rules above. If a FORM LIBRARY card is used, the only kinds of modification cards that may follow it are ADD SET and REORDER.

g. Any discrepancy in the placement or format of a modification card, or a list card or cross-section data card that accompanies a modification card, will result in the entire run being aborted. No new library tape will be formed.

7. The following rules apply to the extraction control cards.

a. No extraction card may be placed before the FORM LIBRARY card.

b. Two or more extraction cards of the same kind having the same set name may be used. Extraction cards of different kinds having the same set name may be used.

c. The RENAME SET card may be used only if it is preceded by the PUNCH SET card. However, a PUNCH SET card may be used either by itself or it may be followed by a RENAME SET card.

d. The RENAME MAT card may be used only if it is preceded by the PUNCH MAT card and list card(s); however, a PUNCH MAT card and its accompanying list card(s) may be used either by themselves or may be followed by a RENAME MAT card [and its accompanying list card(s)].

e. Any discrepancies in an extraction card, or the list card accompanying an extraction card, that appear in section (1) will cause the entire run to be aborted.

f. Any discrepancies in an extraction card or accompanying list card, that occur in section (3), will cause the run to be halted at that point. No subsequent control cards will be processed.

E. Input-Output Units Used by XLIBIT

Logical Unit Number	Use of Function
01	Scratch tape; new library tape (if any)
02*	Scratch tape
03*	Scratch tape
04	Scratch tape
06	Scratch tape
20	Contains input to XLIBIT
21	Contains printed output from XLIBIT
44	Old library tape (if any)
PUNCH	Standard punch unit; contains punch output from XLIBIT

*Note: Tapes 2 and 3 may be bypassed, at the cost of being unable to include materials of a very large "size" in the cross-section library. No precise formula for the maximum "size" of a material that may be processed without tapes 2 and 3 can be derived; however, if for a single material,

$$J[15 + (J + 3)(L + L' + L'') + 3J + J^{\dagger} + 1^{\dagger\dagger}] < 20000^{\S},$$

then the material will not use tapes 2 and 3. Materials for which this relationship does not hold may still not use tapes 2 and 3; this is, however, contingent on a number of other factors. In subroutines TESTSTOR and PRINT1, it is determined whether or not tapes 2 and 3 are required by a material.

J is the total number of energy groups in the material. The quantities L, L', and L" are defined in Part VII, Section E (p. 55) of this report.

---

†Include only if the Fission flag is positive.

††Include only if the Fission flag is negative.

§This is the value given to the quantity MAX (common block 65), which is the size of the container array RCDS (common block 55). The figure of 20000 may be increased according to the amount of core, drum, or disc storage available; this may be done by setting MAX to the new value in subroutine XLIBIT and altering common block 55 in all subroutines in which it appears.

VI. LIST OF ERROR RETURNS

An error message has the following form:

ERFLAG = N

followed by the image of the last input card processed.

In the list that follows, the number given is the value N of ERFLAG corresponding to some error; and, in parentheses, the subroutine name or names in which the particular error occurred.

1. (XLIBIT,DELMATS)

   The program is processing a card that should be a control card; however, this card does not have an asterisk (*) in column one.

2. (MODSET)

   The card following a *MOD SET control card does not have an asterisk (*) in column one.

3. (PUNCHSET)

   The card following a *PUNCH SET control card does not have an asterisk (*) in column one.

4. (READ468)

   A card containing an asterisk (*) in column one was encountered where the program expected a type-4 or type-6 or type-8 cross-section data card.

5. (SCAN1)

   A card without an asterisk (*) in column one has been encountered in the input deck, where the program expects to find a modification control card.

6. (SCAN2)

   A card having the characters "PRINT SE" in columns 2 through 9 does not have either the characters "T·" or "T," in columns 10 and 11.

7. (MODSET)

   The set name as given on a *MOD SET card does not correspond to any of the set names in the set directory of the new library tape being formed. This indicates either that the name on the *MOD SET card does not correspond to the name of any of the sets in the existing library, or that a *DELETE SET card for this set name has been previously encountered in the input deck.

10. (ADDSET)

The new library tape being formed contains more than 300 sets.

11. (ADDSET)

A set being added contains more than 500 materials.

12. (ADDAMAT)

All the set names as given on the type-0 cards for a material are not the same as the "master" set name as obtained either from the *ADD SET card for a set being added, or from the specifications record of an existing set.

13. (ADDAMAT)

All the material names as given on the type-0 cards for a material are not the same as the "master" material name as obtained from the first type-0 card of the material.

14. (ADDAMAT)

The special characters as given on the type-0 cards for a material are not all the same as the special character on the first type-0 card of the material.

15. (ADDSET)

A set being added contains materials with more than 100 energy groups.

16. (ADDSET)

The set name as given on an *ADD SET card duplicates the name of a set already in the existing library or the name of a set on another *ADD SET card encountered earlier in this input deck.

17. (ADDSET)

Same as error 12.

18. (ADDSET)

Same as error 13.

19. (ADDSET)

Same as error 14.

20. (SCAN1)

A control card of a type other than *ADD SET or *REORDER was used following the *FORM LIBRARY card.

21. (SCAN1)

A nonmodification control card was encountered in the modification section [section (2)] of the input deck.

22. (MODSET)

The set name on a *MOD SET control card has been used on a previously encountered *MOD SET card in the input deck.

23. (MODSET)

The set name on a *MOD SET card does not correspond to the name of any set on the old library tape.

24. (MODSET)

The program has tried twice unsuccessfully to position the old library tape to the set name specified on a *MOD SET card. Suspect a tape-drive or data-channel failure.

25. (MODSET)

The card following a *MOD SET control card is neither an *ADD MAT card nor a *DELETE MAT card.

26. (DELMATS)

The number of material names on the list-card(s) following a *DELETE MAT card is not equal to the number of materials to be deleted, as given on the *DELETE MAT card.

27. (DELMATS)

There are one or more duplications of names among the material names as given on the list card(s) following a *DELETE MAT card.

28. (DELMATS)

One of the material names given on the list-card(s) following a *DELETE MAT card does not correspond to any of the names of the materials in the set specified on the *DELETE MAT card.

29. (DELMATS)

The card following the list card(s) that follow a *DELETE MAT card does not have an asterisk (*) in column one. (This card should be a control card of some kind.)

30. (DELMATS)

After a set has been modified [i.e., after a *MOD SET card and either or both of the *DELETE MAT and *ADD MAT cards and their accompanying list card(s) which follow the *MOD SET card have been processed], it is found that there are no materials left in the set; i.e., all have been dele

31. (READZERO)

The group number on a type-0 card for a material is out of order.

32. (READZERO)

The program expected a type-0 cross-section data card, but the card encountered did not have a 0 (zero) in column 13.

33. (READZERO)

The program expected a type-0 cross-section data card, but the card encountered had an asterisk (*) in column one.

34. (DELETSET)

The set name given on a *DELETE SET card was also used on an *ADD SET or a *MOD SET card encountered earlier in the input deck.

35. (DELETSET)

The set name given on a *DELETE SET card does not correspond to any of the names of the sets on the old library tape.

36. (DELETSET)

The set name given on a *DELETE SET card does not correspond to the name of any of the sets on the new library being formed. This implies that the set name was used on another *DELETE SET card encountered earlier in the input deck.

37. (DELETSET)

After a *DELETE SET card has been processed, it is found that no cross-section sets are left on the new library tape being formed; i.e., all have been deleted. (*ADD SET cards included in later portions of the input deck would eliminate this situation, but such usage is inefficient since the net effect of the input deck in which this situation occurs can be accomplished by discarding the old library tape and starting afresh. The user is penalized for inefficient usage of the program via this error stop.)

38. (REORDER)

The set name given on a *REORDER card does not correspond to the name of any of the sets on the new library tape being formed. This implies either that the set is not on the old library tape, or that it has been deleted via a *DELETE SET card bearing this set name that was included earlier in the input deck.

39. (SCANCARD)

A control card in the input deck has faulty punctuation. A period or decimal point occurs in the first field on the card or is used in place of a comma to terminate the first field.

40. (SCANCARD)

The last field on a control card is not terminated by a period, or a control card that is not a list card occupies more than one card.

41. (SCANCARD)

More than eight digits have been assigned to a numeric field on a control card.

42. (SCANCARD)

More than 100 fields have been used on a set of list cards, or the last field on the last list card is not terminated by a period.

43. (DELMATS,MODSET)

A set being modified contains more than 500 materials, as a result of addition of materials to the set.

48. (WRITEMAT)

The number of records written in copying a material block does not equal the value of MORERECS as given in the material-identification record of the material block. Suspect a hardware or tape error.

49. (WRITELIB)

The name of the set located on tape unit 4 does not match the name of the set supposedly in that position, as obtained from the set directory. Suspect a hardware or tape error.

50. (WRITELIB)

The name of a set supposedly on the old library tape cannot be located in the set directory of the old library. Suspect a hardware error.

51. (WRITELIB)

The name of a set on the old library tape does not match the name of the set supposedly in that position, as obtained from the set directory. Suspect a hardware or tape error.

55. (TESTSTOR)

This material cannot be placed on the library tape because there is insufficient room in core to store all the group-constants records. Either reduce the number of groups and/or the size of a single group-constants record, or increase the size of both the quantity MAX (in numbered common, block /65/) and the array RCDS (in numbered common, block /55/).

74. (WRITSCRH)

The switch WFLAG is zero. Suspect a hardware error.

76. (PUTAWAY)

The switch STORFLG(KK) is zero. Suspect a hardware error.

90. (XLIBIT)

No end-of-file mark was found following the directory of sets on the old library tape. Suspect a hardware or tape error.

91. (XLIBIT)

No end-of-file mark was found following the directory of sets on the newly-formed library tape. Suspect a hardware or tape error.

105. (ADDAMAT)

The lethargy or energy value given on a type-0 card differs from the "master" value obtained either from the corresponding type-0 card of the first material of a set being added, or from the group-structure record of a set already in the "old" library. NOTE: This is no longer an error halt; processing continues, using the "master" value.

106. (ADDAMAT)

Same as error 105, but for a velocity value given on a type-0 card.

107. (ADDAMAT)

All the set names on the type-1 cards are not the same as the master set name as obtained either from the *ADD SET card for a set being added, or from the specifications record of an existing set.

108. (ADDAMAT)

The material names as given on the type-1 cards for a material are not all the same as the "master" material name as obtained from the first type-0 card for the material.

109. (ADDAMAT)

The quantities UPSUM(KK) and DWNSUM(KK) are both negative, but LMAX(KK) is nonnegative. Suspect a hardware error.

110. (ADDAMAT)

Cards of type-3 through -8 are not arranged by increasing type; or type-4, -6, or -8 cards were supplied where none were needed; or a card should have begun a new material (type + or 0) or control function (* in col. 1) and did not.

119. (RDSTOR1)

The program expected a type-1 cross-section data card, but instead encountered a card containing an asterisk (*) in column one.

120. (RDSTOR1)

The program expected a type-1 cross-section data card, but instead the card encountered did not have a 1 (one) in column 13.

121. (RDSTOR1)

The group number on a type-1 card is out of order.

124. (RDSTOR2)

The first card following the type-1 cards of a fissionable isotope was not a type-2 card.

125. (RDSTOR2)

The group number on a type-2 cross-section data card is greater than G.

126. (RDSTOR2)

Too many (i.e., more than [(G + 4)/5]) type-2 cards have been used for a single group number. NOTE: The brackets [...] indicate truncation to the next lowest integer.

127. (RDSTOR2)

There is a discrepancy involving the card number on a type-2 card of a material.

For example, cards of the same group number are not arranged by increasing card number, or the card number is omitted where necessary.

128. (RDSTOR2)

Either: a. The set names as given on the type-2 cards of a material are not the same as the "master" set name as obtained either from the *ADD SET card for a set being added, or from the specification record of an existing set.

Or: b. The material name as given on a type-2 card of a material is not the same as the "master" material name as obtained from the first type-0 card of the material.

129. (RDSTOR2)

There is a discrepancy involving the group number of a type-2 card. Either the cards are not arranged by increasing group number, or there is a mixture of nonzero and zero (or blank) group numbers

131. (READ357)

If the material under consideration has scattering, the type-3, -5, or -7 cards are not arranged by increasing group number.

132. (READ357)

The type-3, -5, or -7 cards are not arranged by increasing group number.

133. (READ357)

Same as error 132.

134. (READ357)

The type-3, -5, or -7 cards for some group number are not arranged by increasing ℓ-number.

135. (READ357)

All the set names on the type-3, -5, or -7 cards for a material are not the same as the "master" set name as obtained either from the *ADDSET card for a set being added, or from the specifications record of an existing set.

136. (READ357)

All the material names on the type-3, -5, or -7 cards for a material are not the same as the "master" material name as obtained from the first type-0 card of the material.

137. (READ357)

The type-3, -5, or -7 card with an ℓ-number of zero is missing from the input deck or is out of order.

140. (READ468)

The program expected a cross-section data card of type-4, -6, or -8 but encountered a card that did not have a 4, 6, or 8 (respectively) in column 13. This indicates that some or all of the type-4, -6, or -8 cards are missing from the input deck or are out of order.

141. (READ468)

The type-4, -6, or -8 cards are not arranged by increasing group number.

142. (READ468)

The type-4, -6, or -8 cards for some group number are not arranged by increasing ℓ-number, or an ℓ-number is missing from a type-4, -6, or -8 card.

143. (READ468)

The type-4, -6, or -8 cards for some group number and ℓ-number are not arranged by increasing card number; or a necessary card number is missing from a type-4, -6, or -8 card.

144. (READ468)

All the set names on the type-4, -6, or -8 cards for a material are not the same as the "master" set name as obtained either from the *ADD SET card for a set being added, or from the specifications record for an existing set.

145. (READ468)

All the material names on the type-4, -6, and -8 cards for a material are not the same as the "master" material name as obtained from the first type-0 card of the material.

201. (PRMATDIR)

The set name as given on a *PRINT MAT DIRECTORY, a *PRINT MAT, or a *PRINT SET card does not correspond to any of the names of the sets on the library tape that is to be edited. This can also occur if a *FORM LIBRARY card is used instead of a *FORM NEW EDITION card when an existing "old" library tape has been mounted.

202. (PRMATDIR)

The name of the set just located on the library tape does not correspond to the name of the set supposedly at that position, as obtained from the set directory. Suspect a hardware or tape error.

203. (PRINTMAT)

There is a duplication among the material names as given on the list card(s) following a *PRINT MAT card.

204. (PRINTMAT)

One or more of the material names given on the list card(s) following a *PRINT MAT card cannot be found in the set specified on the *PRINT MAT card.

237. (PUNCHSET)

The set name given on a *PUNCH SET card does not correspond to any of the names of the sets on the library tape to be edited.

238. (PUNCHSET)

The name of the set just located for punching does not agree with the name of the set supposedly at that position, as obtained from the set directory of the library tape. Suspect a hardware or tape error.

301. (PUNCHMAT)

Same as error 238.

302. (PUNCHMAT)

The set name as given on a *PUNCH MAT card does not correspond to any of the names of the sets on the library tape to be edited.

303. (PUNCHMAT)

The card following the list card(s) that follow a *PUNCHMAT card does not have an asterisk (*) in column one. The card should be a control card of some type.

304. (PUNCHMAT)

The number of new material names given on the list card(s) following a *RENAME MAT card is not the same as the number of material names on the list card(s) following a *PUNCH MAT card.

305. (PUNCHMAT)

There is a duplication among the material names given on the list card(s) following a *PUNCH MAT card.

306. (PUNCHMAT)

There is a duplication among the material names given on the list card(s) following a *RENAME MAT card.

307. (PUNCHMAT)

One of the names of materials to be punched (as given on the list card(s) following a *PUNCH MAT card) does not correspond to any of the names of materials on the library tape to be edited.

600. (READZERO)

The velocity value given on the second type-0 card (i.e., type-0, group number 1) was nonzero, but the velocity value on a subsequent type-0 card of the material was blank or zero.

601. (READZERO)

The velocity value given on the second type-0 card (i.e., type-0, group number 1) was nonzero, but the velocity value on a subsequent type-0 card of the material was blank or zero.

602. (READZERO)

The switch K1 is negative. Suspect a hardware error.

603. (ADDMATS)

The name of a material being added to a set duplicates the name of a material already in (or previously added to) the set. The duplicated material is the one before the material whose name is given on the card that accompanies the error message.

800. Called from a number of subroutines.

The switch BFLAG is zero. Suspect a hardware error.

801, 802, 803, 804, 806, 820, 821, 844. Called from a number of subroutines.

Let t = (any of the above values) - 800.

A parity error was sensed on the last buffered read operation on logical tape unit t (e.g., 806 refers to logical unit 6, etc.).

901, 902, 903, 904, 906, 920, 944. Called from a number of subroutines.

Let t = (any of the above values) - 900.

An end-of-file or end-of-tape was sensed on the last buffered read operation on logical unit t (e.g., 944 refers to logical unit 44, etc.).

814 and 914. (PRMATDIR)

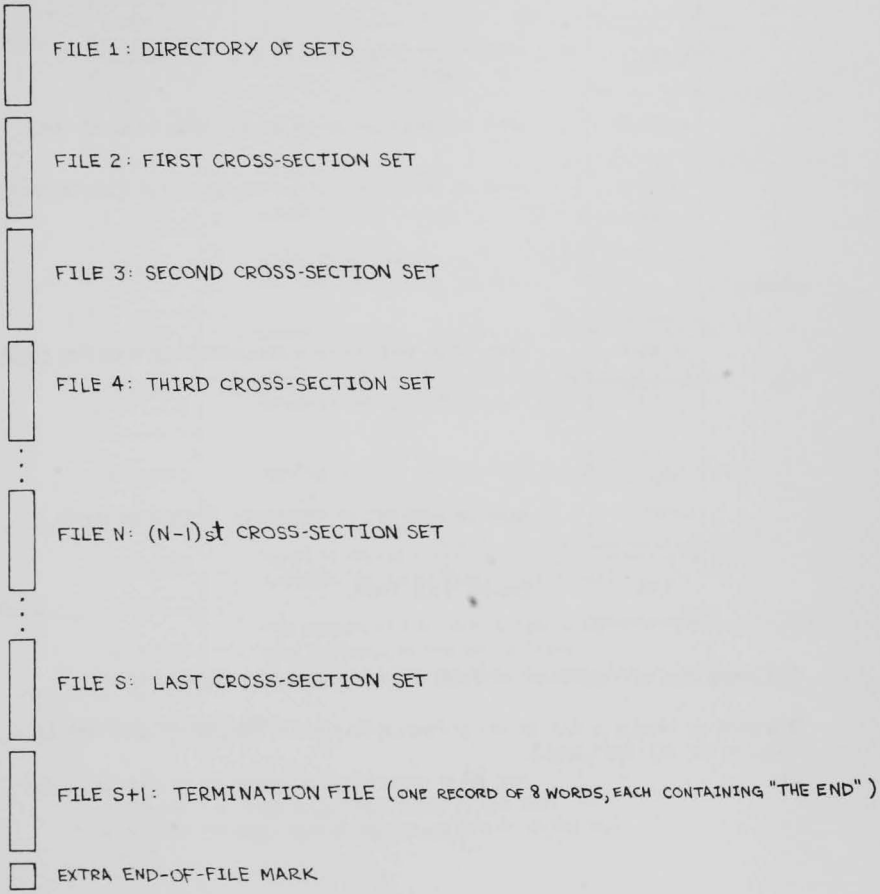
Same as 801 and 901, respectively, if the new library tape (unit 1) is being edited; or the same as 844 and 944, respectively, if the old library tape (unit 44) is being edited.

954. (MODSET)

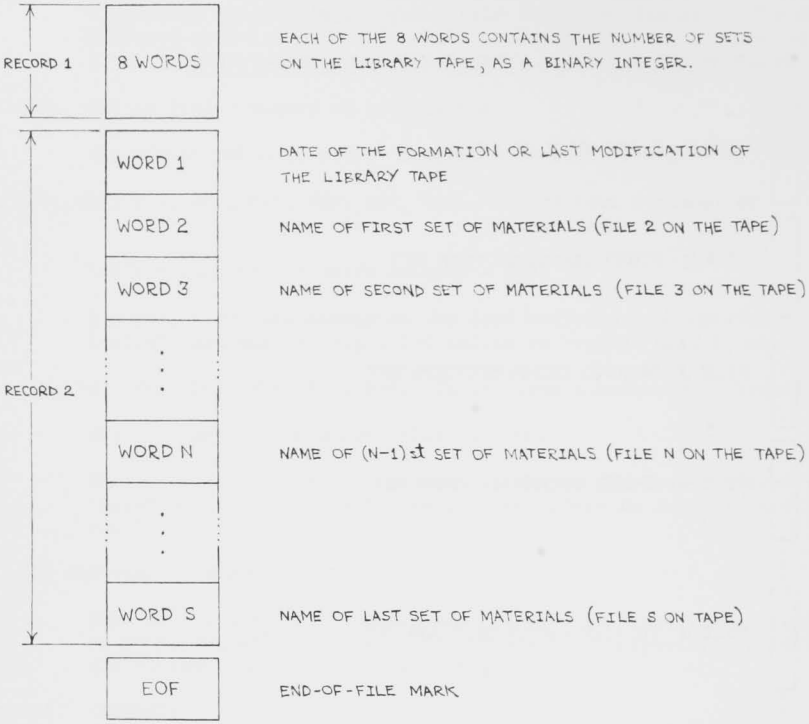
Same as for error 944 (i.e., refers to unit 44).

VII. STRUCTURE OF THE CROSS-SECTION LIBRARY TAPE PREPARED BY THE XLIBIT PROGRAM

A. ORDERING OF THE FILES COMPRISING THE LIBRARY TAPE.



B. DESCRIPTION OF THE DIRECTORY OF SETS FILE (FILE 1 ON THE LIBR



THE NAME OF A SET CONSISTS OF FROM 1 THRU 5 HOLLERITH CHARACTERS.

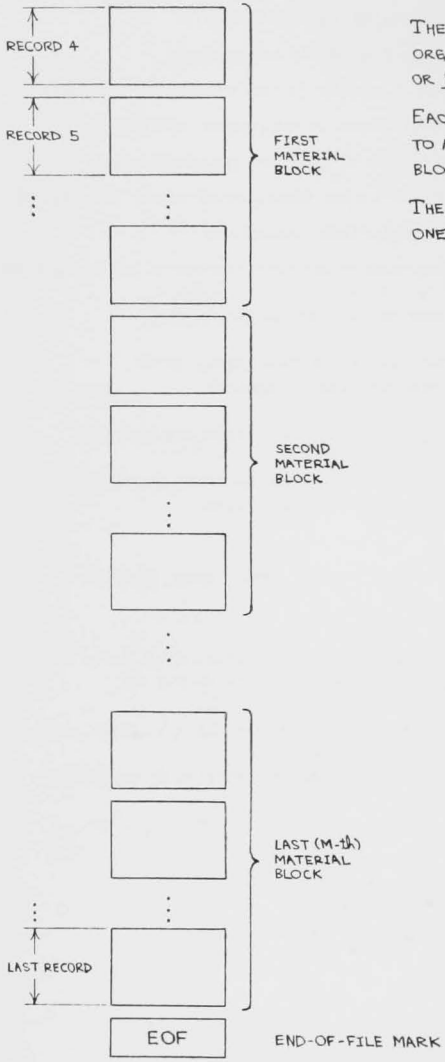
THE DATE IN RECORD 2 MAY BE ANY 1- THRU 8-CHARACTER HOLLERITH QUANTITY (E.G., 2MAR66 OR 19/06/67, ETC.).

C. DESCRIPTION OF A CROSS SECTION SET FILE (FILES 2 THRU 5 ON LIBRARY TAPE).

RECORD 1	WORD 1	SET NAME (1 THRU 5 HOLLERITH CHARACTERS)
	WORD 2	AUTHOR'S NAME (1 THRU 8 HOLLERITH CHARACTERS)
	WORD 3	DATE THIS SET ADDED TO LIBRARY (1 THRU 8 HOLLERITH CHARACTERS)
	WORD 4	J = NUMBER OF ENERGY GROUPS IN SET (BINARY INTEGER)
	WORD 5	M = NUMBER OF MATERIALS IN THIS SET (BINARY INTEGER)
	WORD 6	NAME OF SUBMITTER OF LATEST MODIFICATION TO SET (HOLLERITH)
	WORD 7	DATE OF LATEST MODIFICATION (HOLLERITH)
	WORD 8	NUMBER OF MODIFICATIONS OF SET SINCE ADDITION TO LIBRARY (BINARY)
RECORD 2	1 WORD	UPPER ENERGY LIMIT OF THIS SET, IN eV (BINARY)
	G WORDS	LOWER ENERGY LIMIT IN eV FOR EACH ENERGY GROUP, ARRANGED BY DECREASING ENERGY (BINARY)
	1 WORD	LOWER LETHARGY LIMIT OF SET, DIMENSIONLESS (BINARY)
	G WORDS	UPPER LETHARGY LIMIT FOR EACH ENERGY GROUP, DIMENSIONLESS, ARRANGED BY INCREASING LETHARGY (BINARY)
	G WORDS	MEAN VELOCITY IN cm/sec OF EACH ENERGY GROUP (BINARY)
RECORD 3	M WORDS	NAMES OF THE MATERIALS IN THIS SET (1 THRU 6 HOLLERITH CHAR.), ARRANGED BY THEIR ORDER OF APPEARANCE IN THE SET
	M WORDS	THE NUMBER OF THE FIRST RECORD IN THIS FILE ASSOCIATED WITH EACH MATERIAL IN THE SET (BINARY) (I.E., WHERE TO FIND THE DATA FOR EACH MATERIAL IN THE SET)

(CONTINUED ON NEXT PAGE)

NOTE: RECORD 1 IS THE SPECIFICATIONS RECORD OF THE SET.
RECORD 2 IS THE GROUP STRUCTURE RECORD OF THE SET.
RECORD 3 IS THE DIRECTORY OF MATERIALS RECORD OF THE SET.

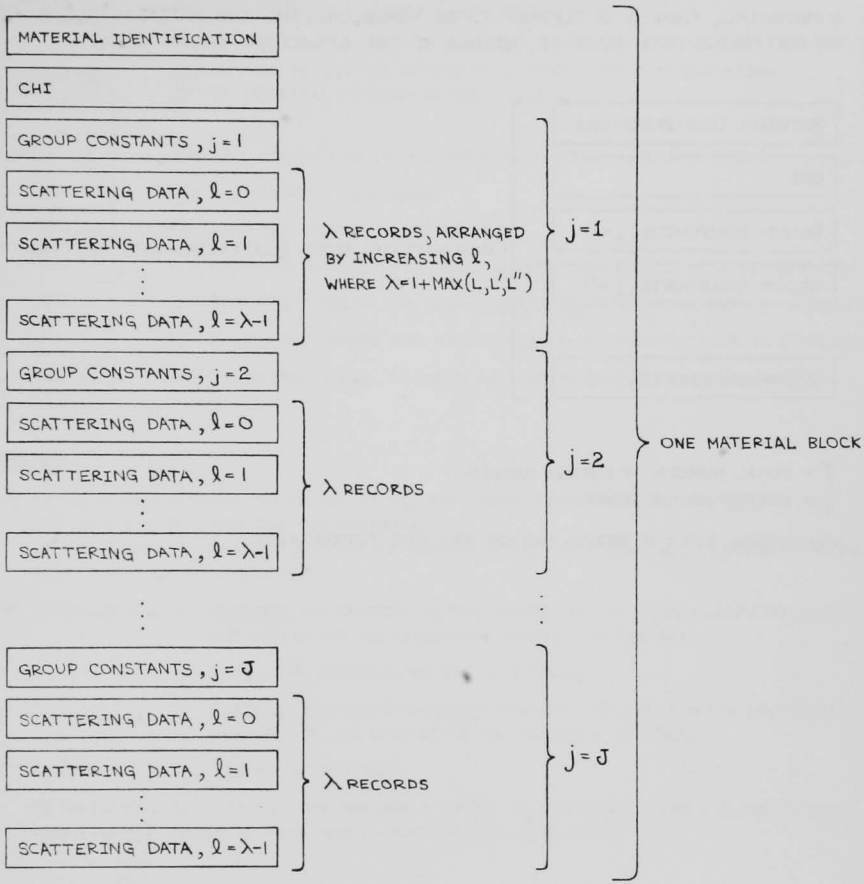


THE REMAINDER OF THE RECORDS IN A SET ARE ORGANIZED INTO SECTIONS CALLED BLOCKS OR MATERIAL BLOCKS.

EACH SUCH BLOCK CONTAINS THE DATA PERTAINING TO A SINGLE MATERIAL. THERE ARE M MATERIAL BLOCKS IN THE SET.

THE NUMBER OF RECORDS IN A BLOCK VARIES FROM ONE BLOCK TO THE NEXT.

D. STRUCTURE OF A MATERIAL BLOCK, BY MATERIAL IDENTIFICATION, CHI, GROUP CONSTANTS, AND SCATTERING DATA RECORDS.



J = TOTAL NUMBER OF ENERGY GROUPS

j = ENERGY GROUP INDEX

ℓ = SCATTERING INDEX, GIVING THE ORDER OF THE ANISOTROPIC SCATTERING (THE VALUE ℓ = 0 INDICATES ISOTROPIC SCATTERING ONLY)

THE QUANTITIES L, L', AND L'' ARE GIVEN IN THE MATERIAL IDENTIFICATION RECORD OF THE BLOCK. SEE SECTION E, BELOW.

NOTE: THE CHI RECORD MAY OR MAY NOT APPEAR, DEPENDING ON THE VALUE OF THE FISSION FLAG IN THE MAT. IDENT. RECORD. SEE SECTIONS E AND F, BELOW.
ALSO THE SCATTERING DATA RECORDS NEED NOT BE PRESENT (SEE SECTION D).

IF NO SCATTERING INFORMATION OF ANY KIND AND FOR ANY ℓ-ORDER IS SUPPLIED FOR A MATERIAL, THEN λ IS DEFINED TO BE MINUS ONE, AND THE MATERIAL BLOCK CONTAINS NO SCATTERING-DATA RECORDS, GIVING IT THE STRUCTURE SHOWN BELOW:

MATERIAL IDENTIFICATION	ONE MATERIAL BLOCK (SCATTERING-DATA OMITTED)
CHI	
GROUP CONSTANTS, $j = 1$	
GROUP CONSTANTS, $j = 2$	
⋮	
GROUP CONSTANTS, $j = J$	

J = TOTAL NUMBER OF ENERGY GROUPS

j = ENERGY GROUP INDEX

AS BEFORE, THE CHI RECORD MAY OR MAY NOT APPEAR, ETC.

E. DESCRIPTION OF THE MATERIAL IDENTIFICATION RECORD OF A BLOCK.

WORD 1	NAME OF MATERIAL (1 THRU 6 HOLLERITH CHARACTERS)
9 WORDS	THE ATOMIC DENSITY (1 THRU 12 HOLL. CHAR.) AND A DESCRIPTION OF THE MATERIAL (1 THRU 60 HOLL. CHAR.)
WORD 11	SPECIAL CHARACTER (1 HOLLERITH CHARACTER, LEFT-ADJUSTED)
WORD 12	T FLAG (BINARY) : SEE BELOW
WORD 13	FISSION FLAG (BINARY) : SEE BELOW
WORD 14	L = MAXIMUM ℓ-ORDER FOR WHICH ELASTIC-SCATTERING DATA IS SUPPLIED
WORD 15	L' = MAXIMUM ℓ-ORDER FOR WHICH INELASTIC-SCATTERING DATA IS SUPPLIED
WORD 16	L'' = MAXIMUM ℓ-ORDER FOR WHICH $(m,2m)$-SCATTERING DATA IS SUPPLIED
WORD 17	NUMBER OF ADDITIONAL RECORDS IN THIS MATERIAL BLOCK (BINARY)

NOTES:

(a) IF THE T FLAG IS ZERO, THE VALUES OF σ_{tr} ARE TAKEN DIRECTLY FROM COLS. 33-44 OF THE TYPE-1 DATA CARDS FOR THE MATERIAL.
IF THE T FLAG IS NON-ZERO, ANY VALUES GIVEN FOR σ_{tr} WERE IGNORED; σ_{tr} WAS COMPUTED FROM OTHER CROSS SECTIONS.

(b) IF FISSION FLAG IS: NEGATIVE, THE MATERIAL IS FISSIONABLE AND ITS FISSION SPECTRUM DOES NOT DEPEND ON THE ENERGY OF THE INCIDENT NEUTRON.

ZERO, THE MATERIAL IS NON-FISSIONABLE.

POSITIVE, THE MATERIAL IS FISSIONABLE AND ITS FISSION SPECTRUM DEPENDS ON THE ENERGY OF THE INCIDENT NEUTRON.

(c) L, L', AND L'' ARE ALL BINARY QUANTITIES.

IF NO ELASTIC SCATTERING DATA FOR ANY ℓ-ORDER WAS SUPPLIED, THEN L IS MINUS ONE; LIKEWISE FOR INELASTIC DATA AND L', AND $(m,2m)$ DATA AND L''.

F. DESCRIPTION OF THE CHI RECORD OF A MATERIAL BLOCK.

J WORDS	THE FISSION SPECTRUM VECTOR $[\chi_k]$.

IF THE FISSION FLAG IS NEGATIVE, THE FISSION SPECTRUM FRACTIONS ARE GROUP-INDEPENDENT AND HAVE THE FORM OF A VECTOR. IN THIS CASE, THE CHI RECORD OCCURS AND THE G WORDS $\chi_k(j \to k)$ IN EACH GROUP CONSTANTS RECORD (SEE SECTION G, BELOW) DO NOT OCCUR.

IF THE FISSION FLAG IS ZERO, THE MATERIAL, BEING NON FISSIONABLE, HAS NO FISSION SPECTRUM. NEITHER THE CHI RECORD NOR THE $\chi_k(j \to k)$ WORDS IN THE GROUP CONSTANTS RECORDS OCCUR.

IF THE FISSION FLAG IS POSITIVE, THE FISSION-SPECTRUM FRACTIONS ARE GROUP-DEPENDENT AND FORM A MATRIX. IN THIS CASE THE CHI RECORD DOES NOT OCCUR BUT THE G WORDS $\chi_k(j \to k)$ IN EACH GROUP CONSTANTS RECORD DO OCCUR (THESE GROUPS OF WORDS ARE THE ROWS OF THE MATRIX).

G. DESCRIPTION OF A GROUP-CONSTANTS RECORD OF A BLOCK.

1 WORD	j-VALUE (ENERGY GROUP INDEX)	
1 WORD	$\sigma_{tr,j}$	
1 WORD	$\sigma_{c,j}$	
1 WORD	D_j	
1 WORD	$\sigma_{f,j}$	THESE WORDS APPEAR ONLY IF THE FISSION FLAG IS NON-ZERO.
1 WORD	ν_j	
J WORDS	$\chi_k(j \to k)$	THESE WORDS APPEAR ONLY IF THE FISSION FLAG IS POSITIVE.
L+1 WORDS	$F^{\ell}_{el,j}$ $(\ell = 0)$ ⋮ $F^{\ell}_{el,j}$ $(\ell = L)$	THESE WORDS APPEAR ONLY IF L IS NON-NEGATIVE.
L+1 WORDS	$G^{\ell}_{el,j}$ $(\ell = 0)$ ⋮ $G^{\ell}_{el,j}$ $(\ell = L)$	
L+1 WORDS	$\sigma^{\ell}_{el,j}$ $(\ell = 0)$ ⋮ $\sigma^{\ell}_{el,j}$ $(\ell = L)$	
L′+1 WORDS	$F^{\ell}_{in,j}$ $(\ell = 0)$ ⋮ $F^{\ell}_{in,j}$ $(\ell = L')$	THESE WORDS APPEAR ONLY IF L′ IS NON-NEGATIVE.
L′+1 WORDS	$G^{\ell}_{in,j}$ $(\ell = 0)$ ⋮ $G^{\ell}_{in,j}$ $(\ell = L')$	
L′+1 WORDS	$\sigma^{\ell}_{in,j}$ $(\ell = 0)$ ⋮ $\sigma^{\ell}_{in,j}$ $(\ell = L')$	
L″+1 WORDS	$F^{\ell}_{(n,2n),j}$ $(\ell = 0)$ ⋮ $F^{\ell}_{(n,2n),j}$ $(\ell = L'')$	THESE WORDS APPEAR ONLY IF L″ IS NON-NEGATIVE.
L″+1 WORDS	$G^{\ell}_{(n,2n),j}$ $(\ell = 0)$ ⋮ $G^{\ell}_{(n,2n),j}$ $(\ell = L'')$	
L″+1 WORDS	$\sigma^{\ell}_{(n,2n),j}$ $(\ell = 0)$ ⋮ $\sigma^{\ell}_{(n,2n),j}$ $(\ell = L'')$	

GROUP-CONSTANTS RECORD (CONTINUED)

IF NO SCATTERING DATA IS SUPPLIED (SEE ALSO SECTION D, ABOVE), THE GROUP-CONSTANTS RECORDS OF A MATERIAL BLOCK HAVE THE FOLLOWING STRUCTURE :

1 WORD	j-VALUE (ENERGY GROUP INDEX)	
1 WORD	$\sigma_{tr,j}$	
1 WORD	$\sigma_{c,j}$	
1 WORD	D_j	
1 WORD	$\sigma_{f,j}$	ONLY IF FISSION FLAG IS NON-ZERO.
1 WORD	ν_j	
J WORDS	$\chi_k(j \rightarrow k)$	ONLY IF FISSION FLAG IS POSITIVE.

O

H. DESCRIPTION OF A SCATTERING-DATA RECORD.

1 WORD	ℓ-VALUE	
$F^{\ell}_{el,j} + G^{\ell}_{el,j} + 1$ WORDS	$T^{\ell}_{el}(j \to k)$	THESE WORDS APPEAR IF AND ONLY IF $L \geq 0$ AND $\ell \leq L$ AND AT LEAST ONE OF THE QUANTITIES $F^{\ell}_{el,j}$ AND $G^{\ell}_{el,j}$ IS NON-ZERO.
$F^{\ell}_{in,j} + G^{\ell}_{in,j} + 1$ WORDS	$T^{\ell}_{in}(j \to k)$	THESE WORDS APPEAR IF AND ONLY IF $L' \geq 0$ AND $\ell \leq L'$ AND AT LEAST ONE OF THE QUANTITIES $F^{\ell}_{in,j}$ AND $G^{\ell}_{in,j}$ IS NON-ZERO.
$F^{\ell}_{(m,2m),j} + G^{\ell}_{(m,2m),j} + 1$ WORDS	$T^{\ell}_{(m,2m)}(j \to k)$	THESE WORDS APPEAR IF AND ONLY IF $L'' \geq 0$ AND $\ell \leq L''$ AND AT LEAST ONE OF THE QUANTITIES $F^{\ell}_{(m,2m),j}$ AND $G^{\ell}_{(m,2m),j}$ IS NON-ZERO.

NOTE: THE j-VALUE ASSOCIATED WITH A SCATTERING DATA RECORD IS THAT OF THE PRECEDING GROUP CONSTANTS RECORD (SEE SECTION D, ABOVE).

IF ALL THREE GROUPS OF DATA (I.E., T_{el}, T_{in}, AND $T_{(m,2m)}$) DO NOT APPEAR FOR SOME PAIR OF j AND ℓ VALUES, THEN THE SCATTERING DATA RECORD FOR THAT COMBINATION OF j AND ℓ HAS THE STRUCTURE SHOWN BELOW:

ℓ-VALUE	
0	SEVEN WORDS, EACH CONTAINING A BINARY ZERO.
0	
0	
0	
0	
0	
0	

APPENDIX A

Sample XLIBIT Input Forms

SET [ ][ ][ ][ ][ ] (1 2 3 4 5) MATERIAL [ ][ ][ ][ ][ ][ ] (6 7 8 9 10 11) SPECIAL CHARACTER [ ] (12)

ANL CROSS SECTION LIBRARY

ATOMIC DENSITY COMMENTS

[+] [ ] (13–25) [ ] (26–80)

TYPE [0] (13)

GROUP # j (14 15 16)	CARD # (17 18 19 20)	ENERGY (E_j) UNITS (21–32)	LETHARGY (u_j) UNITS (33–44)	MEAN VELOCITY CM./SEC. (45–56)	E OPT. (a) (57)	D OPT. (b) (58)	TR OPT. (c) (59)
0							

(a) Leave E option blank, if group structure is in energy units. Put "x" in box if group structure is specified in lethargy units.

(b) Leave D option blank, if the diffusion coefficients as given are to be used for diffusion theory programs. Put "x" in box if the diffusion coefficients are to be calculated $= {}^{1}/3\sigma_{tr}$. If transport cross sections are not given and all other cross sections are given σ_{tr} will be calculated and used.

(c) Leave TR option blank, if transport cross sections as given are to be used in transport theory programs. Put "x" in box if the transport cross sections are to be computed from all other given cross sections.

DATE _____ NAME _____

PAGE _____ OF _____

SET [ ][ ][ ][ ][ ] (1 2 3 4 5) MATERIAL [ ][ ][ ][ ][ ][ ] (6 7 8 9 10 11) SPECIAL CHARACTER [ ] (12)

ANL CROSS SECTION LIBRARY

TYPE [1] (13)

GROUP # j (14 15 16)	CARD # (17 18 19)	(20)	DIFFUSION COEFFICIENT D_j (21–32)	TRANSPORT CROSS SECTION $\sigma_{tr,j}$ (33–44)	CAPTURE CROSS SECTION $\sigma_{c,j}$ (45–56)	FISSION CROSS SECTION $\sigma_{f,j}$ (57–68)	NU ν_j (69–80)

COMMENTS:

DATE _____ NAME _____

PAGE _____ OF _____

SET [ ][ ][ ][ ][ ] (1 2 3 4 5) MATERIAL [ ][ ][ ][ ][ ][ ] (6 7 8 9 10 11) SPECIAL CHARACTER [ ] (12)

ANL CROSS SECTION LIBRARY

TYPE [2] (13)

GROUP # j			CARD #				χ(j → k), FISSION SPECTRUM FRACTIONS, k →				
14	15	16	17	18	19	20	21 32	33 44	45 56	57 68	69 80

COMMENTS:

DATE _____ NAME _____

PAGE _____ OF _____

SET [ ][ ][ ][ ][ ] (1 2 3 4 5) MATERIAL [ ][ ][ ][ ][ ][ ] (6 7 8 9 10 11) SPECIAL CHARACTER [ ] (12)

TYPE [ ] (13) (a)

GROUP # j			CARD #			ℓ (b)	CROSS SECTION σ^{ℓ}	UP F^{ℓ} (c)			DOWN G^{ℓ} (d)		
14	15	16	17	18	19	20	21 — 32	33	34	35	36	37	38

COMMENTS:

(a) TYPE:
=3 ELASTIC
=5 INELASTIC
=7 (n,2n)

(b) ℓ=0,1,... indicating the order of the Legendre expansion.

(c) F^{ℓ}= the maximum number of groups into which upscattering cross sections are given.

(d) G^{ℓ}= the maximum number of groups into which downscattering cross sections are given.

$F^{\ell}=G^{\ell}=0$ implies no $T^{\ell}(j\rightarrow k)$ matrix and $\sigma^{\ell}(j\rightarrow j)$ is given in cols. 21-32.

DATE _____ NAME _____

PAGE _____ OF _____

SET [ ][ ][ ][ ][ ] (1 2 3 4 5) MATERIAL [ ][ ][ ][ ][ ][ ] (6 7 8 9 10 11) SPECIAL CHARACTER [ ] (12) ANL CROSS SECTION LIBRARY

TYPE [ ] (13) *TYPE (4=ELASTIC, 6=INELASTIC, 8=(n,2n))

GROUP # j			CARD #			ℓ	$T^{\ell}(j \rightarrow k)$, SCATTERING MATRIX, k →				
14	15	16	17	18	19	20	21 32	33 44	45 56	57 68	69 80

COMMENTS:

DATE _____ NAME _____

PAGE _____ OF _____

APPENDIX B

List of XLIBIT Subroutines and Common Block Assignments

1. List of Subroutines

Subroutine Name	Values of ERFLAG (Error Stops) in Subroutine
XLIBIT	1,90,91,820,844,920,944
ADDAMAT	12,13,14,105,106,107,108,109,110,801,802,803,901, 902,903
ADDMATS	603,804,806,904,906
ADDSET	16,17,18,19,804,904
AHALT	
BCD2BIN(BCD,BIN)	
BITS(N1,LHB,N2)	
CALCDTRX	
COMPARES	
DELETSET	34,35,36,37
DELMATS	1,26,27,28,29,30,804,844,904,944
ERRORZ	
FIXADD	
FIX2ADD	
GETSIZE	
GET1012	800
GREAD	800,820,920
MODSET	2,7,22,23,24,25,804,806,844,904,906,944,954
NEWPAGE	821
PCHTP468	802,803,804,902,903,904
PNCHMATS	801,844,901,944
PNCH1MAT	802,803,804,844,902,903,904,944
PRESET	
PRINTMAT	203,204,801,844
PRINTSET	
PRINT1	802,804,821,902,904
PRMATDIR	201,202,814,821,914
PRNTSCRH	
PRSETDIR	821
PUNCHMAT	301,302,303,304,305,306,307,801,844,901,944
PUNCHSET	3,237,238,801,844,901,944
PUNCH357	
PUNCH468	

1. List of Subroutines (Contd.)

Subroutine Name	Values of ERFLAG (Error Stops) in Subroutine
PUTAWAY	76,801,802,803,901,902,903
RCLDTH(L)	
READZERO	31,32,33,600,601,602,800
RDSTOR1	119,120,121,800
RDSTOR2	124,125,126,127,128,129,800
READ357	131,132,133,134,135,136,137,800
READ468	4,140,141,142,143,144,145,800
REORDER	38
ROWLTH	
SCANCARD	39,40,41,42
SCAN1	5,20,21
SCAN2	6
SKIPFILE	801,844,901,944
SKPFILES(IARG1,IARG2)	804,806,844
SPACEBY(TAPE)	844,944
STORGRP	
STORSCAT	
TESTSTOR	55
WRITELIB	49,50,51,801,804,806,844,901,904,906,944
WRITEMAT	48,801,806,844,901,906,944
WRITSCRH	74,801,802,803,806,901,902,903,906

2. Common Blocks Used by XLIBIT

Block No.	Variable Names in Block
5	MATNAME,MDESCRIP(9),SPCHAR,TFLAG,FFLAG,LMAX(3),MORERECS
10	SETNAME,AUTHOR,DATE,G,M,MODAUTHR,MODDATE,NUMOFMOD
15	ES(102),LS(102),VS(102)
20	MATIDS(500),MATLOC(501)
25	NUMSETS1,NUMSETS2,SETID(101),MATID(101),SPCH(101),TAPES
30	SETDIR1(301),SETDIR2(301)
35	CHI(101)
40	K
45	L
50	EOPT,DOPT,TROPT

2. Common Blocks Used by XLIBIT (Contd.)

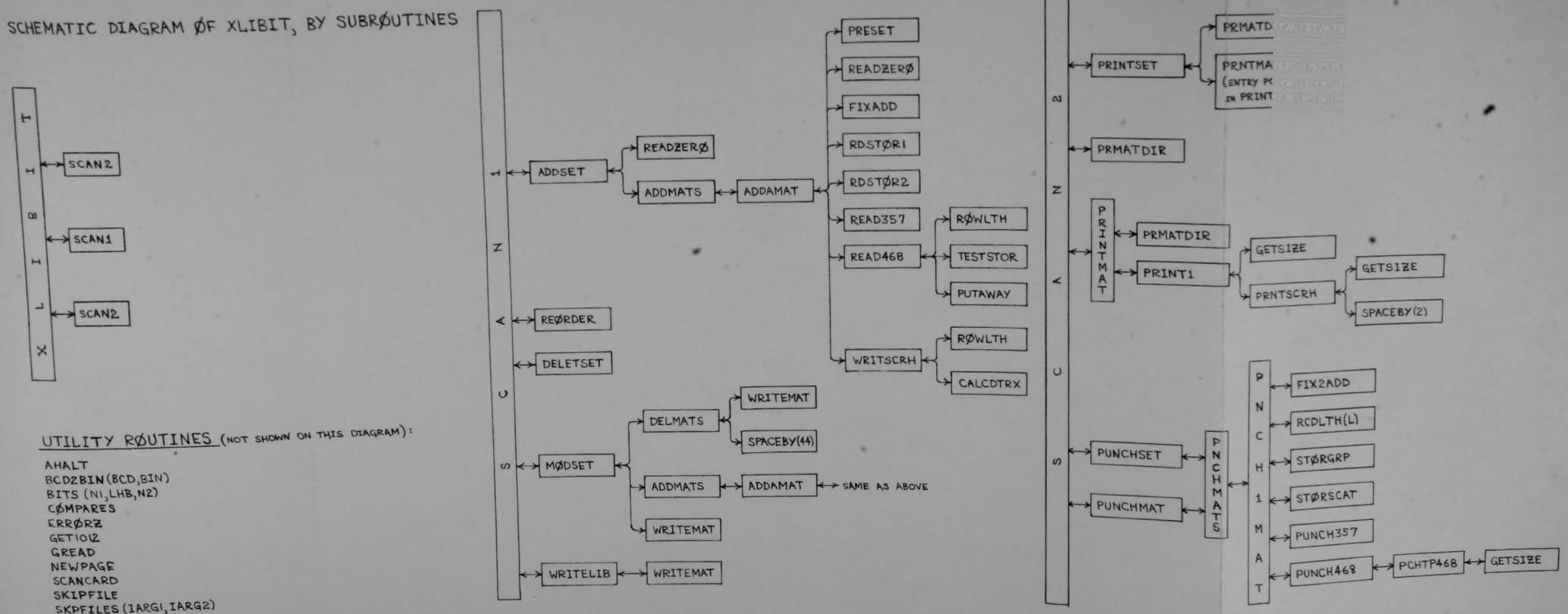
Block No.	Variable Names in Block
55	RCDS(20000)
60	BFLAG,COL1,COL2THR9,TYPE,CARD1(10),CARD2(10)
65	MAX
70	MN,IX(101)
75	PMATID(200),PM
80	NUMATS,NUMOFMAT
85	FSUM,NUSUM
90	LISTCARD,FIELD(100),FLD
95	FLGONE,FLGTWO,STORFLG(3),LNTH(3)
100	GFLAG
105	ERFLAG
110	DWN(10),UP(10),DWNSUM(3),UPSUM(3),DWNSM(3,10),UPSM(3,10),SXSEC(10)
115	LTH,LENGT,NN1
120	KK,KK1,K4
125	K3(101)
130	JJ,CNO
135	FORMCARD
140	SETLOC(301),S6FILES
145	COLS1011,COL12
150	LIB,PAGENO,DIR,PUNCHED,LASTLOC
155	MATABLE(200)
160	NEWNAME(200)
165	UPELIM,LWLLIM,KKI(3),LENTH(3),RLTH,LAMDA
170	BUFAREA(500)
175	BUFAREA2(304)
180	KK2,KK3
185	TYP
190	L1,NF(3),NG(3),I4(10),I7(3),I8(3)
195	J
200	I1,I2
205	IPARAM1(101),IPARAM2
210	NSCAN(6)
215	TSUM(101,3)

APPENDIX C

Flow Charts and Descriptions

(See also introductory paragraphs of Part V, Section D)

SCHEMATIC DIAGRAM ØF XLIBIT, BY SUBRØUTINES



UTILITY RØUTINES (NOT SHOWN ON THIS DIAGRAM):

AHALT
BCDZBIN(BCD,BIN)
BITS (N1,LHB,N2)
CØMPARES
ERRØRZ
GET1012
GREAD
NEWPAGE
SCANCARD
SKIPFILE
SKPFILES(IARG1,IARG2)

2. Description of XLIBIT Subroutines

XLIBIT controls the flow of the program, testing for the presence of cards of sections (1), (2), and (3) of the input deck (see p. 34).

Cards in section (1) of an input deck specify portions of the existing "old" library tape that are to be printed and/or punched.

Cards in section (2) of an input deck specify additions, deletions, or modifications that are to be made to the cross-section data on the old library tape. A new library tape which incorporates these changes is to be formed.

Cards in section (3) specify portions of the newly-formed library tape that are to be printed and/or punched.

An input deck may consist of only the following combinations of sections: (1) only; (2) only; (1) and (2); (2) and (3); (1), (2), and (3).

2. Description of XLIBIT Subroutines (Contd.)

1. A nonzero value of ERFLAG indicates an error in the input deck.

   FORMCARD indicates the presence (FORMCARD = 0) or absence (FORMCARD = 1) of an old library tape.

   PUNCHED is set to one if portions of the library are punched; it is zero otherwise.

   BFLAG regulates the reading and decoding of data cards.

   LASTLOC indicates the current position of an existing library tape. If LASTLOC equals N, then the library tape is positioned after the EOF terminating the Nth file on the tape.

2. Zero out the master storage array RCDS.

3. Read the first card of the data deck.

4. In GREAD the next input card is read into core; the card previously read is decoded.

5. The first input card must contain an asterisk in column one.

   ERRORZ terminates the program if an error is detected in the input deck.

6. If the first input card is *FORM LIBRARY, or *FORM NEW TAPE, then there is no section (1) in the input deck. In this case, SCAN2 is not called at this time.

   If the first input card is *FORM LIBRARY, then there is <u>no</u> existing "old" library tape. In this case, set FORMCARD = 1, insert Hollerith blanks in the set directories, set the numbers of sets to zero, and proceed to process the section (2) input cards.

   If the first input card is *FORM NEW TAPE, then there <u>is</u> an existing library tape. This library tape is on logical tape unit 44. This tape is rewound, and the directory of sets file is read; this file consists of two records and an end-of-file (EOF) mark.

   Leave FORMCARD equal to zero.

7. If the first input card is neither *FORM LIBRARY nor *FORM NEW TAPE, then there is a section (1) in the input deck. The section (1) input cards are processed by SCAN2. The presence of an old library tape is implied.

   LIB is set to the logical unit number of the tape whose contents are to be printed and/or punched.

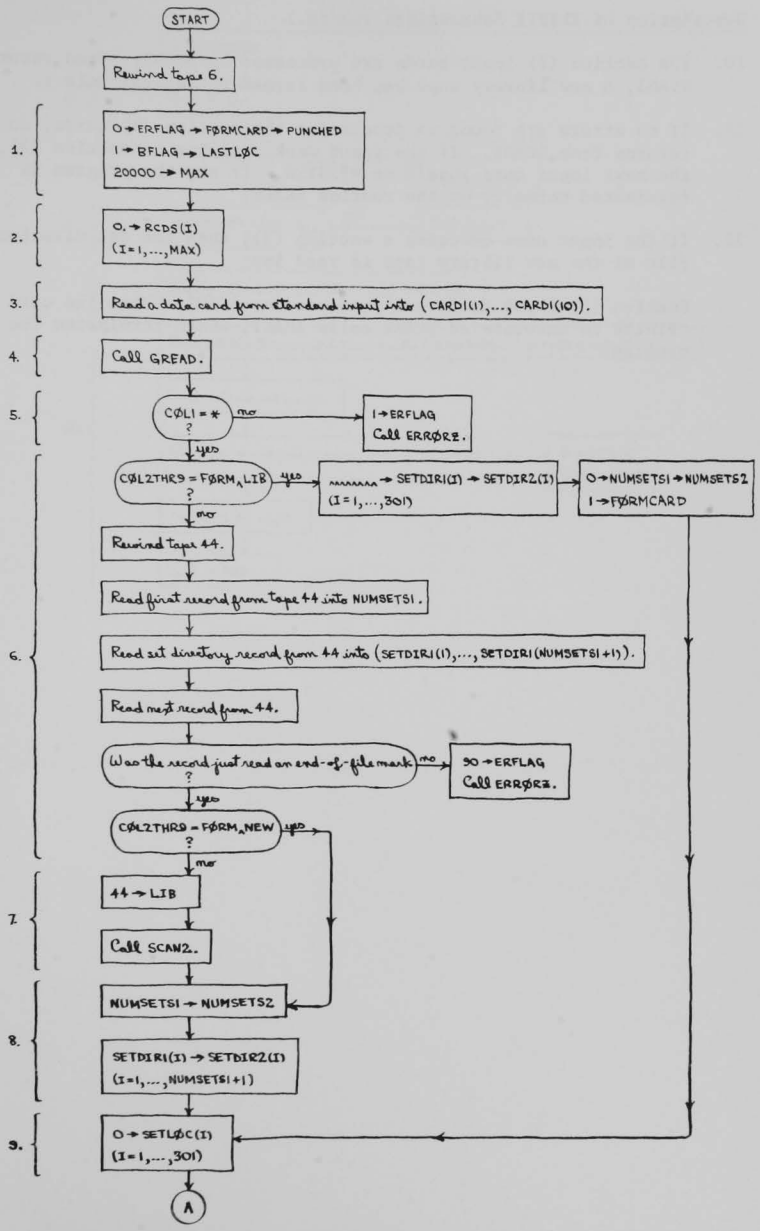
   If the input deck contains only a section (1), the program terminates normally in the subroutine AHALT, which is called by SCAN2.

8. If the input deck contains a section (2), control returns from SCAN2. The new set directory and the number of sets in the new library are made initially the same as those of the old library.

9. SETLOC (I) = 0 indicates that the Ith cross-section set in the library is to be found on the old library tape.

   If SETLOC (I) = N, then the Ith cross-section set in the library is to be found as the Nth file on each of two scratch tapes, logical tape units 4 and 6.

   This information is used by the routine WRITELIB to form the new edition of the library tapes.


START
Rewind tape 6.
1.
0 → ERFLAG → FØRMCARD → PUNCHED
1 → BFLAG → LASTLØC
20000 → MAX
2.
0. → RCDS(I)
(I=1,...,MAX)
3.
Read a data card from standard input into (CARD1(1),..., CARD1(10)).
4.
Call GREAD.
5.
CØL1 = * ?
no
1 → ERFLAG
Call ERRØR2.
yes
CØL2THRU9 = FØRM,LIB ?
yes
→ SETDIR1(I) → SETDIR2(I)
(I=1,...,301)
0 → NUMSETS1 → NUMSETS2
1 → FØRMCARD
no
Rewind tape 44.
Read first record from tape 44 into NUMSETS1.
6.
Read set directory record from 44 into (SETDIR1(1),..., SETDIR1(NUMSETS1+1)).
Read next record from 44.
Was the record just read an end-of-file mark ?
no
90 → ERFLAG
Call ERRØR3.
yes
CØL2THRU9 = FØRM,NEW ?
yes
no
7.
44 → LIB
Call SCAN2.
8.
NUMSETS1 → NUMSETS2
SETDIR1(I) → SETDIR2(I)
(I=1,...,NUMSETS1+1)
9.
0 → SETLØC(I)
(I=1,...,301)
A

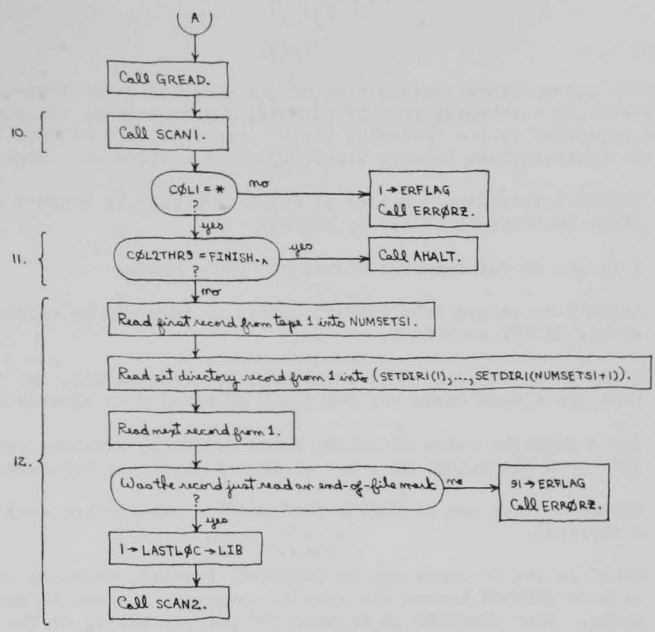
2. Description of XLIBIT Subroutines (Contd.)

10. The section (2) input cards are processed by SCAN1. Upon return from SCAN1, a new library tape has been formed on logical unit 1.

11. If no errors are found in processing the section (2) cards, control returns from SCAN1. If the input deck contains no section (3), then the next input card should be *FINISH. If so, the program is terminated normally by the routine AHALT.

12. If the input deck contains a section (3), then the set directory file of the new library tape is read in.

    Section (3) input cards are processed by SCAN2. When the card *FINISH is encountered SCAN2 calls AHALT, which terminates the program.


A
Call GREAD.
10.
Call SCAN1.
CØL1 = *
?
no
1 → ERFLAG
Call ERRØRZ.
yes
11.
CØL2THR3 = FINISH.
?
yes
Call AHALT.
no
12.
Read first record from tape 1 into NUMSETS1.
Read set directory record from 1 into (SETDIR1(1), ..., SETDIR1(NUMSETS1+1)).
Read next record from 1.
Was the record just read an end-of-file mark
?
no
91 → ERFLAG
Call ERRØRZ.
yes
1 → LASTLØC → LIB
Call SCAN2.

ADDAMAT

ADDAMAT accomplishes the addition of one material to a cross-section set. The cross-section data cards for the material are processed; the data from the cards are organized into a "material block" (see Section D of Part VII, Structure of the Cross-section Library Tape), which is written onto logical tape 6.

1. PRESET initializes a number of quantities used by ADDAMAT and the other subroutines called by ADDAMAT.

2. K is set to the expected number of type-0 cards.

3. ADDAMAT is called from ADDMATS, which in turn may be called from either ADDSET or MODSET.

   GFLAG is -1 when the call to ADDMATS is from ADDSETS, and indicates that the type-0 cards for the first material have already been read.

   For a negative value of GFLAG, those following sections that read and check the values on a set of type-0 cards are bypassed.

4. READZERO reads and processes the type-0 cross-section data cards of a material.

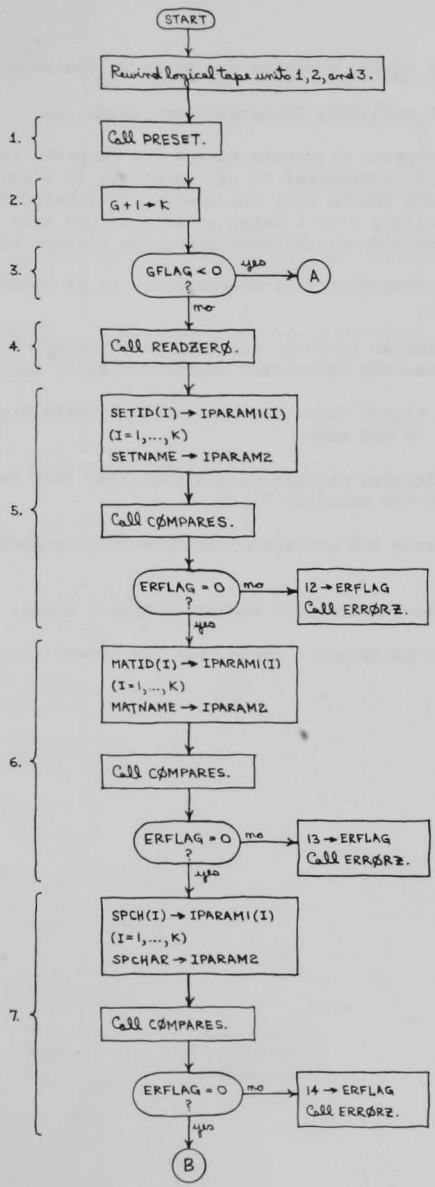
   GFLAG is set to minus one in READZERO; however, GFLAG is reset to zero in ADDMATS before the test in paragraph 3 above is encountered again. Thus READZERO is entered for the processing of the second and any subsequent materials in the set.

5. This section checks that the set identifications on all type-0 cards are identical to the name of the set in which the material is to be placed.

   COMPARES matches each of the quantities IPARAM1(I) against IPARAM2. If a discrepancy is found, a nonzero value is given to ERFLAG.

6. This section checks that the material identifications on all type-0 cards are identical.

7. This section checks that the special characters on all type-0 cards are identical.


START
Rewind logical tape units 1, 2, and 3.
1.
Call PRESET.
2.
G + 1 → K
3.
GFLAG < 0 ?
yes
A
no
4.
Call READZERØ.
5.
SETID(I) → IPARAM1(I)
(I = 1, ..., K)
SETNAME → IPARAM2
Call CØMPARES.
ERFLAG = 0 ?
no
12 → ERFLAG
Call ERRØRZ.
yes
6.
MATID(I) → IPARAM1(I)
(I = 1, ..., K)
MATNAME → IPARAM2
Call CØMPARES.
ERFLAG = 0 ?
no
13 → ERFLAG
Call ERRØRZ.
yes
7.
SPCH(I) → IPARAM1(I)
(I = 1, ..., K)
SPCHAR → IPARAM2
Call CØMPARES.
ERFLAG = 0 ?
no
14 → ERFLAG
Call ERRØRZ.
yes
B

ADDAMAT (Contd.)

8. If EOPT is blank, energies are given on the type-0 cards.

   If EOPT is nonblank, lethargies are given.

9. The energy-group structure (which may be given in terms of either energies or lethargies) of all materials in a set must be similar. This section checks that the energy or lethargy structure of the material arrays E or L being added does not vary by more than 0.1% from the already-established structure (arrays ES or LS).

   If such a discrepancy is encountered, it is noted and processing continues.

10. A check similar to that described in paragraph 9 above is made on the mean neutron velocities corresponding to each group.

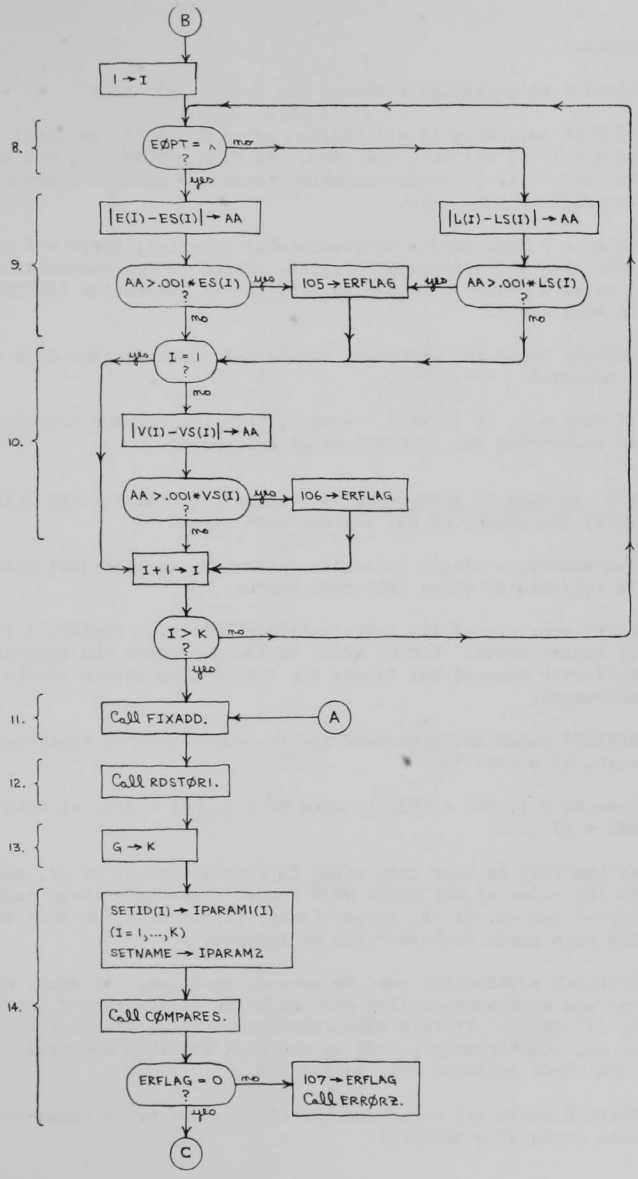
    The first type-0 card does not contain a velocity, so for this card the check is not made.

11. FIXADD allocates storage within the array RCDS for the group-constant records of the material block.

12. RDSTOR1 reads and processes the type-1 cross-section data cards of a material.

13. K is set to the expected number of type-1 cards.

14. Similar to paragraph 5 above, but for type-1 cards.


B
1 → I
8.
EØPT = ?
no
yes
9.
|E(I) − ES(I)| → AA
|L(I) − LS(I)| → AA
AA > .001 * ES(I) ?
yes
105 → ERFLAG
yes
AA > .001 * LS(I) ?
no
no
10.
I = 1 ?
yes
no
|V(I) − VS(I)| → AA
AA > .001 * VS(I) ?
yes
106 → ERFLAG
no
I + 1 → I
I > K ?
no
yes
11.
Call FIXADD.
A
12.
Call RDSTØRI.
13.
G → K
14.
SETID(I) → IPARAM1(I)
(I = 1, ..., K)
SETNAME → IPARAM2
Call CØMPARES.
ERFLAG = 0 ?
no
107 → ERFLAG
Call ERRØRZ.
yes
C

ADDAMAT (Contd.)

15. Similar to paragraph 6 above, but for type-1 cards.

16. FSUM is zero only if all fission cross sections are zero; NUSM is zero only if all nu's are zero. If FSUM = NUSM = 0, the material is assumed to be nonfissionable; there are no type-2 cards for the material in this case.

    FFLAG = 0 indicates a nonfissionable material; FFLAG = 1 or -1 indicates a fissionable material. FFLAG is set unconditionally to 1 here; a minus-one value is assigned in subroutine RDSTOR2, if it is applicable.

17. RDSTOR2 reads and processes the type-2 cross-section data cards for a material.

17.5. If COL1 = *, or if TYPE = + or 0, then the current material has no scattering and is complete at this point.

18. KK1, as used in paragraphs 18 through 23, contains (in Hollerith code) the number of the current card type.

    The number, a single Hollerith character, is left-justified and is followed by seven Hollerith blanks.

    KDELT consists of the left-justified Hollerith numeral 1 followed by binary zeros. Adding KDELT to KK1 increases the left-justified Hollerith numeral but leaves the trailing Hollerith blanks unchanged.

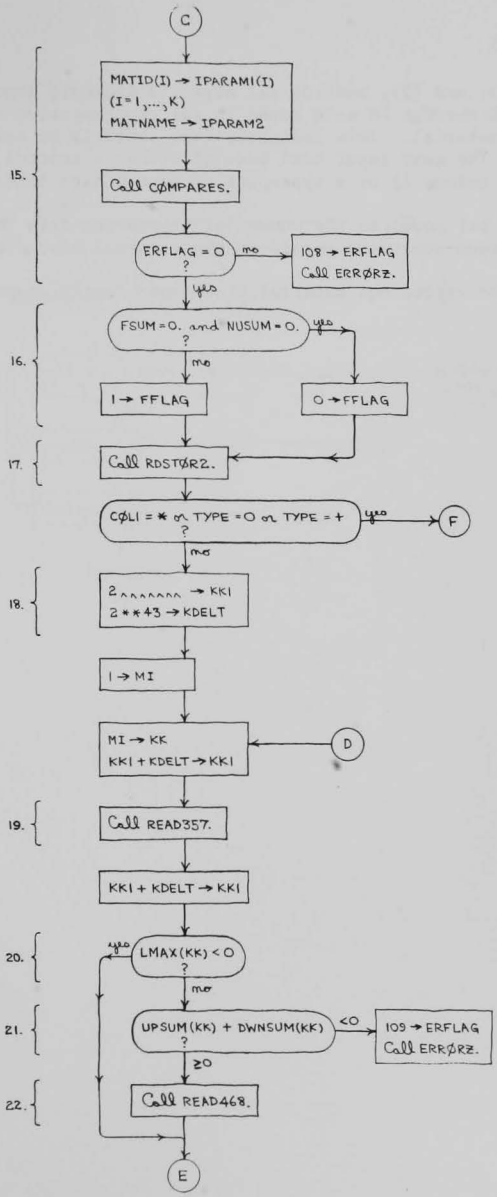
19. READ357 reads and processes the type-3, -5, or -7 cross-section data cards of a material.

    When MI = 1, KK1 = 8H3...; when MI = 2, KK1 = 8H5...; when MI = 3, KK1 = 87... .

20. If LMAX(KK) is less than zero, no type-3 (or -5, or -7, according to the value of MI) cards were found. Therefore there need be no type-4 (or -6, or -8, respectively) input cards for this material. Any such cards included will be ignored.

21. UPSUM(KK)+DWNSUM(KK) must be zero or positive. If zero, then no up- and no downscattering was indicated on the type-3 (or -5, or -7) cards. In this case, there need be no type-4 (or -6, or -8, respectively) cards in the deck for this material. Any such cards included will be ignored.

22. READ468 reads and processes the type-4, -6, or -8 cross-section data cards of a material.

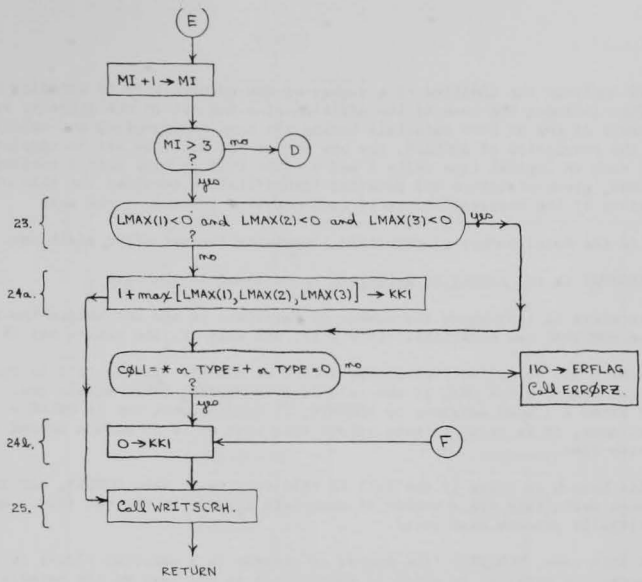

C
MATID(I) → IPARAM1(I)
(I = 1,...,K)
MATNAME → IPARAM2
15.
Call CØMPARES.
ERFLAG = 0 ?
no
108 → ERFLAG
Call ERRØRZ.
yes
FSUM = 0. and NUSUM = 0. ?
yes
16.
no
1 → FFLAG
0 → FFLAG
17.
Call RDSTØR2.
CØL1 = * or TYPE = 0 or TYPE = +
yes
F
no
18.
2^^^^^^^ → KK1
2**43 → KDELT
1 → MI
MI → KK
KK1 + KDELT → KK1
D
19.
Call READ357.
KK1 + KDELT → KK1
20.
yes
LMAX(KK) < 0 ?
no
21.
UPSUM(KK) + DWNSUM(KK) ?
<0
109 → ERFLAG
Call ERRØRZ.
≥0
22.
Call READ468.
E

ADDAMAT (Contd.)

23. LMAX(1), and (2), and (3) all negative indicate that no cards of types-3 through -8 were found in the cross-section data-card deck for a material. This indicates that there is no scattering of any kind. The next input card must be either a control card (with a * in column 1) or a type-plus or type-0 data card.

24. KK1 is set equal to the number of scattering data that will follow each group-constants record in the material block being formed.

25. WRITSCRH writes one material block onto logical tape unit 6.


E
MI + 1 → MI
MI > 3 ?
no
D
yes
23.
LMAX(1) < 0 and LMAX(2) < 0 and LMAX(3) < 0 ?
yes
no
24a.
1 + max [LMAX(1), LMAX(2), LMAX(3)] → KK1
CØLI = * or TYPE = + or TYPE = 0 ?
no
110 → ERFLAG
Call ERRØRZ.
yes
24b.
0 → KK1
F
25.
Call WRITSCRH.
RETURN

ADDMATS

ADDMATS controls the addition of a number of new materials to an existing set of materials. This includes the case of the addition of a new set to the library, since a set consists merely of one or more materials having the same energy-group and velocity structures. At the conclusion of ADDMATS, the new set or the modified set is complete; it exists as one file each on logical tape units 4 and 6. The file on tape unit 4 consists of the specifications, group-structure and material-identification records; the file on tape unit 6 consists of the material blocks of each material comprising the set.

1. M is the total number of materials comprising the set after additions.

   NUMOFMAT is the number of materials to be added to the set.

   Therefore LA represents the number of materials in the set before the addition of the NUMOFMAT new materials. LA = 0 if, and only if, the entire set is being added.

2. ADDMATS may be called from either ADDSET or MODSET. If the call is from ADDSET, GFLAG equals minus one; if the call is from MODSET, GFLAG equals one. Note that if GFLAG = 1 upon entrance to ADDMATS, it remains one, but if GFLAG = -1 upon entrance, it is reset to zero before this test on it is made a second or a later time.

3. This branch is taken if the call to this routine is from MODSET, and implies that there definitely are a number of materials already in the set (i.e., LA is initially greater than zero).

   In this case, MORERECS (the number of records in a material block) is already known; the material to which it corresponds is the last of the materials already in the set. Its value is saved in N before ADDAMAT is called.

   ADDAMAT adds one material to the set. The number of records in the material block of this material is stored in MORERECS by the subroutine WRITSCRH, which is called by ADDAMAT.

4. This branch is taken if ADDMATS is called from ADDSET.

   In this case there are no materials already in the set; initially LA is zero, and as yet no value of MORERECS is available. Therefore the initial storage of MORERECS in N is redundant; note that for this case N is not used.

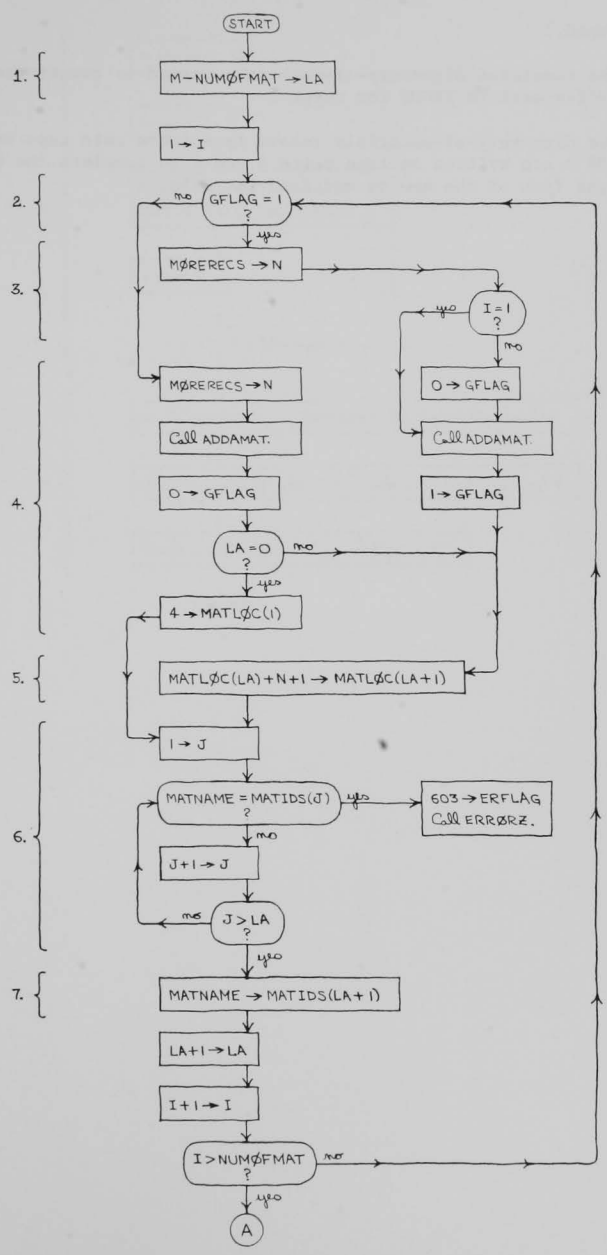
   If GFLAG is not one, it is either zero or minus one. In this branch it is necessary that GFLAG = 0 the second and subsequent times ADDAMAT is called. This is done to insure that the type-0 cards for the next material are read properly; a test on GFLAG for this is made at the start of READZERO.

   If LA = 0, the material being added is the first in the set; MATLOC (1) is always four. Note that the incrementing of LA insures that LA is zero only once. (See also paragraph 5 below.)

5. The number of the first record of the material block corresponding to the Ith material in the set is placed in MATLOC(I). The array MATLOC comprises the latter half of the directory of materials record. Note that each time this branch is traversed, there are LA materials already in the set, so that the material currently being added is the (LA + 1)st.

6. A check is made to insure that the name of the material just added does not duplicate the name of any material already in the set.

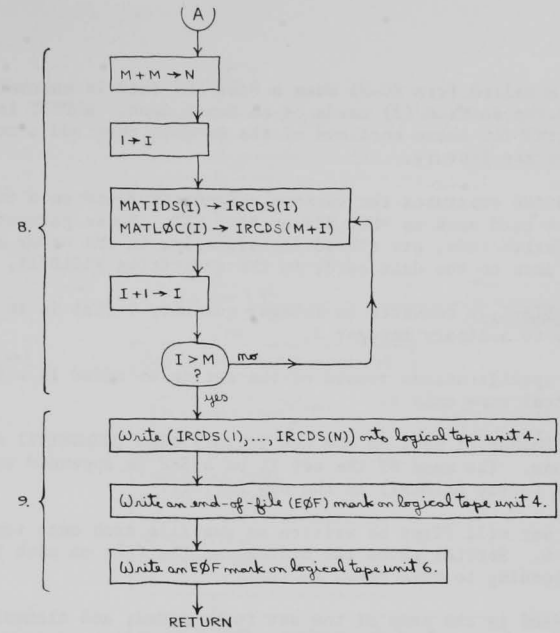
7. The name of the Ith material in the set is placed in MATIDS(I). The array MATIDS comprises the first half of the director-of-materials record.


START
1.
M-NUMØFMAT → LA
1 → I
2.
GFLAG = 1 ?
no
yes
3.
MØRERECS → N
I = 1 ?
yes
no
4.
MØRERECS → N
Call ADDAMAT.
0 → GFLAG
LA = 0 ?
no
yes
0 → GFLAG
Call ADDAMAT.
1 → GFLAG
4 → MATLØC(1)
5.
MATLØC(LA) + N + 1 → MATLØC(LA + 1)
6.
1 → J
MATNAME = MATIDS(J) ?
yes
603 → ERFLAG
Call ERRØRZ.
no
J + 1 → J
J > LA ?
no
yes
7.
MATNAME → MATIDS(LA + 1)
LA + 1 → LA
I + 1 → I
I > NUMØFMAT ?
no
yes
A

ADDMATS (Contd.)

8. The completed directory-of-materials record is transferred to a buffer area in IRCDS for output.

9. The directory-of-materials record is written onto tape unit 4. EOF's are written on tape units 4 and 6 to complete the scratch-tape form of the new or modified set file.


A
M + M → N
1 → I
MATIDS(I) → IRCDS(I)
MATLØC(I) → IRCDS(M+I)
8.
I + 1 → I
I > M ?
no
yes
Write (IRCDS(1), ..., IRCDS(N)) onto logical tape unit 4.
Write an end-of-file (EØF) mark on logical tape unit 4.
9.
Write an EØF mark on logical tape unit 6.
RETURN

ADDSET

ADDSET is called from SCAN1 when a *ADD SET card is encountered during processing of the section (2) cards of an input deck. ADDSET is the controlling routine for those sections of the program that add a new cross-section set to the library.

1. SCANCARD separates the various parameters found on a control-type input card such as *MOD SET or *ADD SET. These parameters, still in Hollerith code, are stored individually, in the order of their appearance on the data card, in the quantities FIELD(1), FIELD(2), etc.

2. BCD2BIN(I,J) converts an integer quantity I that is in Hollerith code to a binary integer J.

3. The specifications record of the set to be added is written onto logical tape unit 4.

4. The number of sets on the new library tape (NUMSETS2) is increased by one. The name of the set to be added is appended to the list of sets (array SETDIR2) in the new library.

   The set will first be written as one file each onto tape units 4 and 6. S6FILES gives the ordinal of the file on each tape corresponding to this set.

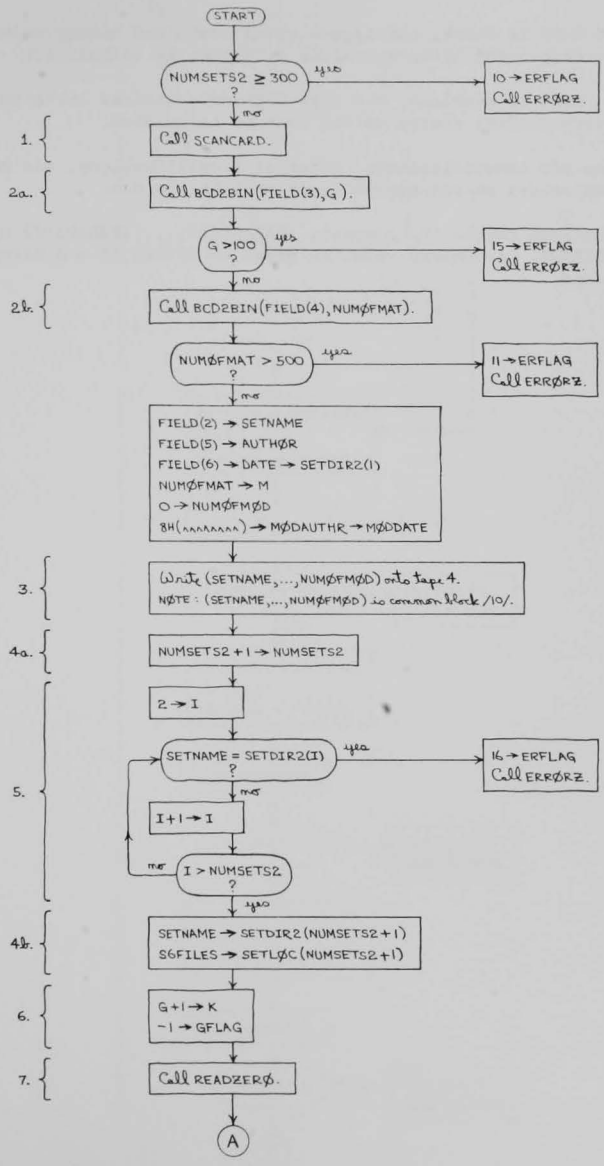
5. SETNAME is the name of the set to be added, and elements 2 through NUMSETS2 of the array SETDIR2 contain the list of names of sets already on the new library tape.

   SETNAME must not duplicate any of SETDIR2(2),...,SETDIR2(NUMSETS2).

6. K is set to the expected number of type-0 cards. A nonzero value of GFLAG indicates to READZERO that the first type-0 card has not yet been read.

7. READZERO reads, and processes the data on, the type-0 data cards for one material.

ADDSET


START
NUMSETS2 ≥ 300 ?
yes
10 → ERFLAG
Call ERRØRZ.
no
1.
Call SCANCARD.
2a.
Call BCD2BIN(FIELD(3),G).
G > 100 ?
yes
15 → ERFLAG
Call ERRØRZ.
no
2b.
Call BCD2BIN(FIELD(4), NUMØFMAT).
NUMØFMAT > 500 ?
yes
11 → ERFLAG
Call ERRØRZ.
no
FIELD(2) → SETNAME
FIELD(5) → AUTHØR
FIELD(6) → DATE → SETDIR2(1)
NUMØFMAT → M
0 → NUMØFMØD
8H(^^^^^^^^) → MØDAUTHR → MØDDATE
3.
Write (SETNAME,...,NUMØFMØD) onto tape 4.
NØTE: (SETNAME,...,NUMØFMØD) in common block /10/.
4a.
NUMSETS2 + 1 → NUMSETS2
5.
2 → I
SETNAME = SETDIR2(I) ?
yes
16 → ERFLAG
Call ERRØRZ.
no
I + 1 → I
no
I > NUMSETS2 ?
yes
4b.
SETNAME → SETDIR2(NUMSETS2 + 1)
S6FILES → SETLØC(NUMSETS2 + 1)
6.
G + 1 → K
−1 → GFLAG
7.
Call READZERØ.
A

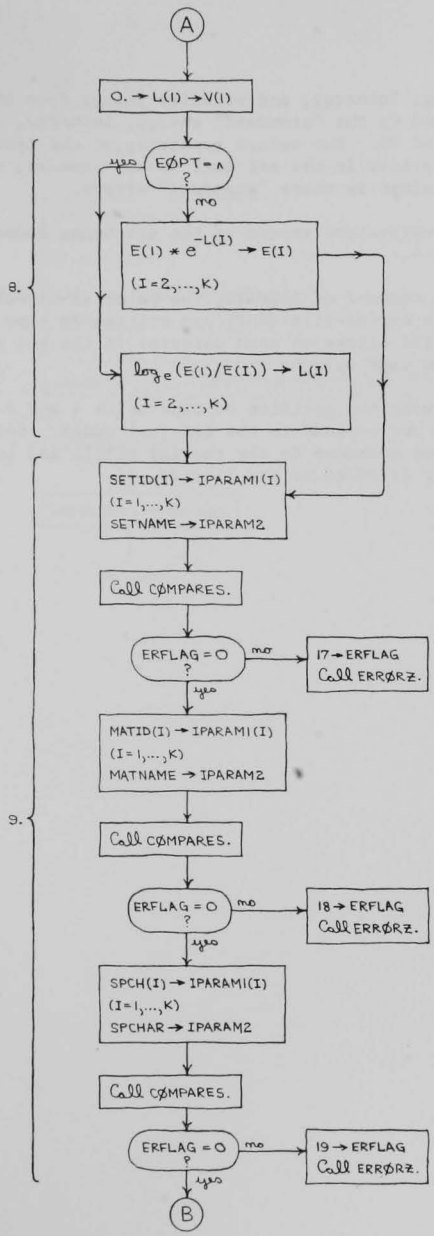
ADDSET (Contd.)

8. If EOPT is blank, the type-0 cards contained energy values E(I). Corresponding lethargy values L(I) must be calculated.

   If EOPT is nonblank, the type-0 cards contained lethargy values; corresponding energy values must be calculated.

9. The set identifications, material identifications, and special characters on all type-0 cards must be the same.

   COMPARES checks the elements IPARAMI(1),...,IPARAMI(K) against IPARAM2. A nonzero value is given to ERFLAG if a discrepancy is found.


A
0. → L(I) → V(I)
EØPT = 1 ?
yes
no
E(I) * e^-L(I) → E(I)
(I = 2,...,K)
8.
log_e (E(I)/E(I)) → L(I)
(I = 2,...,K)
SETID(I) → IPARAM1(I)
(I = 1,...,K)
SETNAME → IPARAM2
Call CØMPARES.
ERFLAG = 0 ?
no
17 → ERFLAG
Call ERRØRZ.
yes
MATID(I) → IPARAM1(I)
(I = 1,...,K)
MATNAME → IPARAM2
9.
Call CØMPARES.
ERFLAG = 0 ?
no
18 → ERFLAG
Call ERRØRZ.
yes
SPCH(I) → IPARAM1(I)
(I = 1,...,K)
SPCHAR → IPARAM2
Call CØMPARES.
ERFLAG = 0 ?
no
19 → ERFLAG
Call ERRØRZ.
yes
B

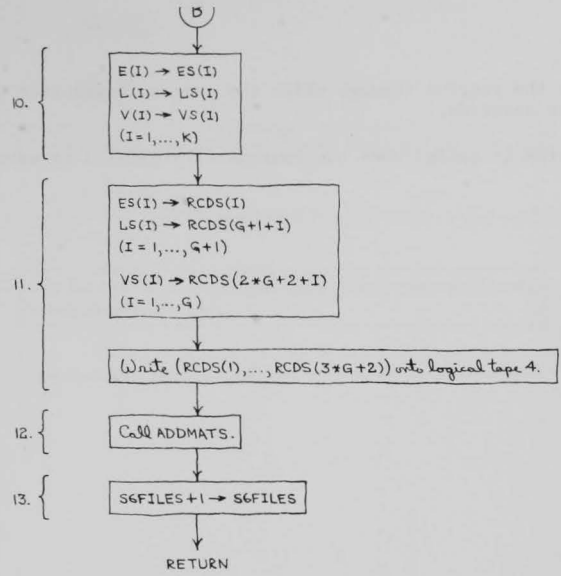
ADDSET (Contd.)

10. The energy, lethargy, and velocity values from the type-0 cards are transferred to the "standard" energy, lethargy, and velocity arrays ES, LS, and VS. The values appearing on the type-0 cards of all other materials in the set must be the same as, and will be compared to, the values in these "standard" arrays.

11. The group-structure record of the set being added is written onto tape unit 4.

12. Under the control of ADDMATS, the material-directory record of the set and an end-of-file (EOF) are written on tape unit 4; in addition, the material blocks of each material in the set and an EOF are written on tape unit 6.

13. S6FILES marks the position on tape units 4 and 6 of the file on each tape that corresponds to the set just added. S6FILES is initialized to one upon entrance to the routine SCAN1, and is incremented each time a set is added to the library.

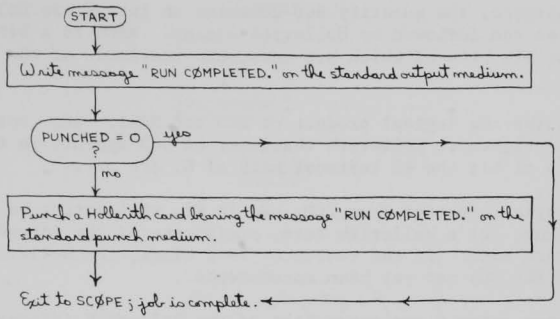

B
10.
E(I) → ES(I)
L(I) → LS(I)
V(I) → VS(I)
(I = 1,...,K)
11.
ES(I) → RCDS(I)
LS(I) → RCDS(G+1+I)
(I = 1,...,G+1)
VS(I) → RCDS(2*G+2+I)
(I = 1,...,G)
Write (RCDS(1),...,RCDS(3*G+2)) onto logical tape 4.
12.
Call ADDMATS.
13.
SGFILES+1 → SGFILES
RETURN

AHALT

AHALT is the routine through which the program terminates normally, when no errors have occurred.

The routine is called when the control card *FINISH is encountered.

AHALT


START
Write message "RUN CØMPLETED." on the standard output medium.
PUNCHED = 0 ?
yes
no
Punch a Hollerith card bearing the message "RUN CØMPLETED." on the standard punch medium.
Exit to SCØPE; job is complete.

BCD2BIN (BCD,BIN)

Upon entrance, the quantity BCD contains an integer in Hollerith code, left-justified and followed by Hollerith blanks. MASK is a binary quantity, the rightmost six bits of which are ones, the remainder of the bits being zeros.

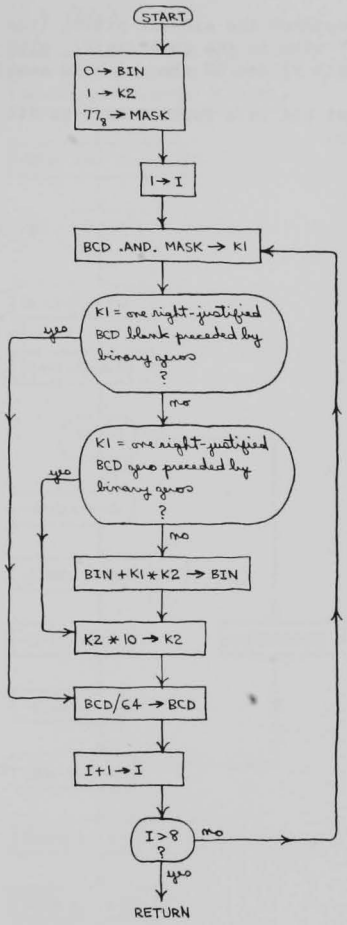
K1 contains the logical product of BCD and MASK. Its contents are as follows: the rightmost Hollerith character of BCD appears in the rightmost six bits of K1; the 42 leftmost bits of K1 are zeros.

This single character from BCD, now in K1, will either be (1) a Hollerith blank, (2) a Hollerith zero, or (3) one of the integers 1 through 9, in Hollerith code. If the character is a blank, the Hollerith integer contained in BCD has not yet been encountered.

The binary integer representation of the Hollerith integer contained in BCD is built up in BIN; the method used is based on the fact that the six-bit Hollerith code for the integers 1 through 9 is equivalent to their binary representation. (For example, six in Hollerith is in octal form 06. In binary form, this 06 is 000110, which conforms to the machine format for binary integers.)

Dividing BCD by 64 has the effect of shifting BCD six bits to the right, so that K1 contains each of the eight characters comprising BCD, one at a time, starting from the right.

BCD2BIN

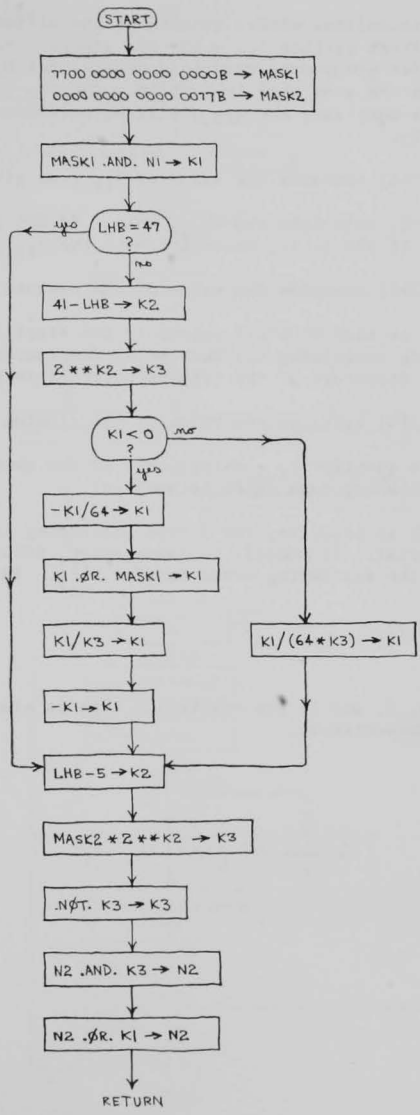

START
0 → BIN
1 → K2
77₈ → MASK
I → I
BCD .AND. MASK → K1
K1 = one right-justified BCD blank preceded by binary zeros ?
yes
no
K1 = one right-justified BCD zero preceded by binary zeros ?
yes
no
BIN + K1 * K2 → BIN
K2 * 10 → K2
BCD/64 → BCD
I+1 → I
I > 8 ?
no
yes
RETURN

BITS (N1,LHB,N2)

This subroutine replaces the six-bit string (the index of the leftmost bit of which is LHB) of bits in the quantity N2, with the leftmost six bits of the quantity N1. Both N1 and N2 are one-word quantities.

NOTE: The leftmost bit in a machine word is bit 47; the rightmost bit is bit zero.

BITS


START
7700 0000 0000 0000B → MASK1
0000 0000 0000 0077B → MASK2
MASK1 .AND. N1 → K1
LHB = 47 ?
yes
no
41 − LHB → K2
2 ** K2 → K3
K1 < 0 ?
no
yes
− K1/64 → K1
K1 .ØR. MASK1 → K1
K1/K3 → K1
K1/(64 * K3) → K1
− K1 → K1
LHB − 5 → K2
MASK2 * 2 ** K2 → K3
.NØT. K3 → K3
N2 .AND. K3 → N2
N2 .ØR. K1 → N2
RETURN

CALCDTRX

This routine calculates either or both of the diffusion coefficients D_J and the transport cross section $\sigma_{tr,J}$ for all groups J = 1, ..., G. If the D option on the first group-zero card is nonblank, the D_J will be calculated; if the TR option on the same card is nonblank, the $\sigma_{tr,J}$ will be calculated. If both options are nonblank, the $\sigma_{tr,J}$ will be calculated first and then used to calculate D_J.

1. RCDS(IX(1)+1) contains the value of $\sigma_{tr,I}$ as given on the input card.

2. If NN1 is 1, calculate the $\sigma_{tr,J}$ only. If NN1 is ten, calculate the D_J only. If NN1 is 11, calculate both the $\sigma_{tr,J}$ and the D_J.

3. RCDS(IX(I)+2) contains the value of the capture cross section $\sigma_{c,I}$.

4. L1 is set so that RCDS(L1) points to the start of the groups of $\ell + 1$ words containing the scattering information (see Section F of Part VII, Structure of the Cross-section Library Tape).

   RCDS(IX(I)+4) contains the value of the fission cross section $\sigma_{f,I}$.

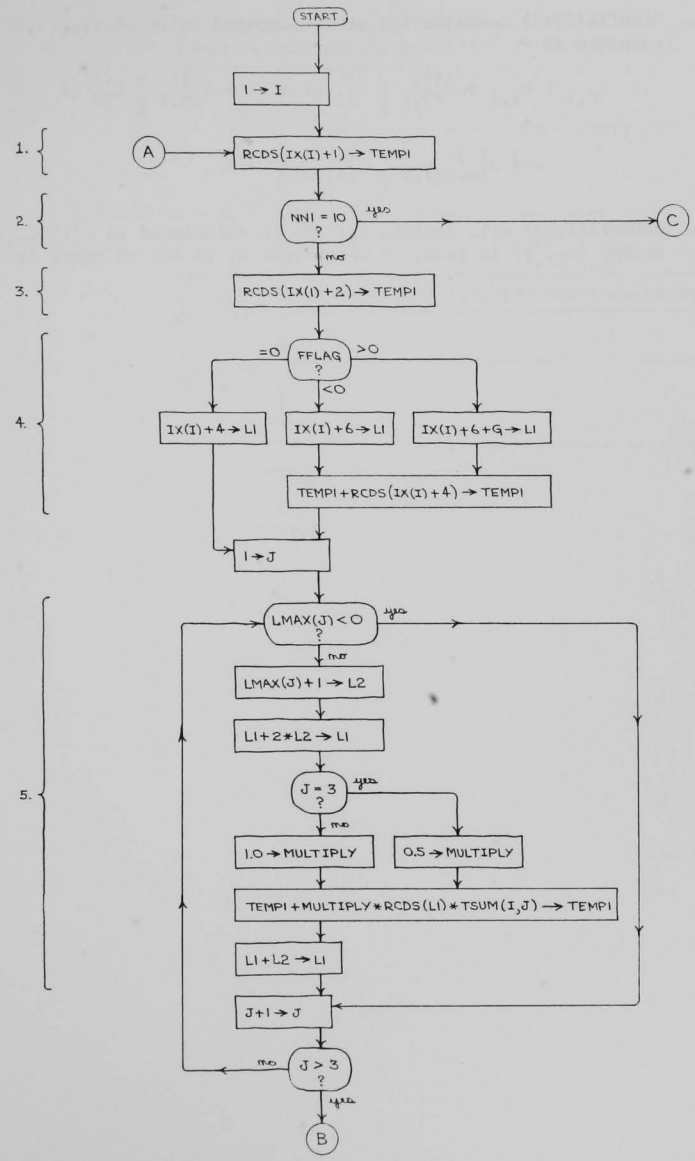
   NOTE: The quantity $\sigma_{f,I}$ exists only if the material is fissionable (in which case FFLAG is nonzero).

5. If LMAX(J) is negative, the J-type scattering is not considered for this material. If LMAX(J) is nonnegative, RCDS(L1) contains the value of the scattering cross section $\sigma_{J,I}^{(0)}$. TSUM(I,J) contains the sum

$$\sum_J T_J^{(0)}(I \rightarrow J).$$

   For J = 1, 2, and 3, the scattering type is elastic, inelastic, and (n,2n), respectively.

CALCDTRX


START
1 → I
1.
A
RCDS(IX(I)+1) → TEMPI
2.
NN1 = 10 ?
yes
C
no
3.
RCDS(IX(I)+2) → TEMPI
4.
FFLAG ?
=0
>0
<0
IX(I)+4 → L1
IX(I)+6 → L1
IX(I)+6+G → L1
TEMPI+RCDS(IX(I)+4) → TEMPI
1 → J
5.
LMAX(J) < 0 ?
yes
no
LMAX(J)+1 → L2
L1+2*L2 → L1
J = 3 ?
yes
no
1.0 → MULTIPLY
0.5 → MULTIPLY
TEMPI+MULTIPLY*RCDS(L1)*TSUM(I,J) → TEMPI
L1+L2 → L1
J+1 → J
J > 3 ?
no
yes
B

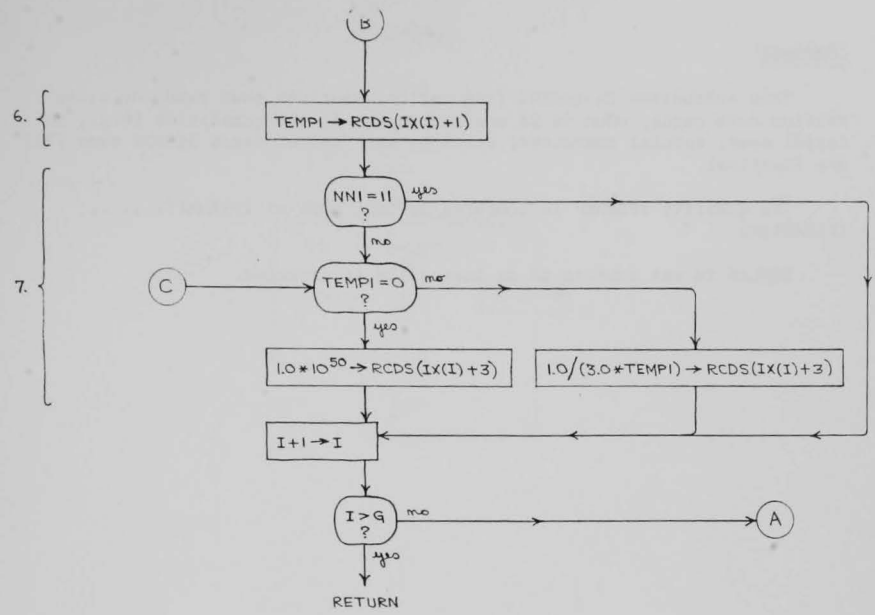
CALCDTRX (Contd.)

6. RCDS(IX(I)+1) contains the newly-computed value of $\sigma_{tr,I}$; it is computed as

$$\sigma_{c,I} + \sigma_{f,I} + \sigma^{(0)}_{el,I} \sum_K T^{(0)}_{el}(I \to K) + \sigma^{(0)}_{in,I} \sum_K T^{(0)}_{in}(I \to K)$$

$$+ \frac{1}{2} \sigma^{(0)}_{(n,2n),I} \sum_K T^{(0)}_{(n,2n)}(I \to K).$$

7. RCDS(IX(I)+3) will contain D_I. D_I is calculated as $1/(3\sigma_{tr,I})$ unless $|\sigma_{tr,I}|$ is zero, in which case D_I is set to equal to 10^{50}.


B
6.
TEMPI → RCDS(IX(I)+1)
7.
NNI = 11 ?
yes
no
C
TEMPI = 0 ?
no
yes
1.0 * 10^50 → RCDS(IX(I)+3)
1.0/(3.0*TEMPI) → RCDS(IX(I)+3)
I+1 → I
I > G ?
no
A
yes
RETURN

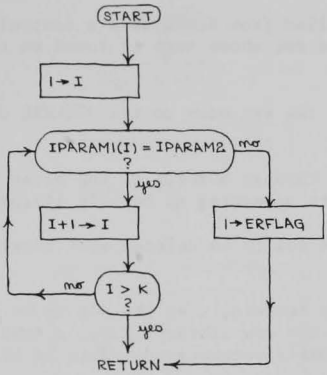
COMPARES

This subroutine is called from various routines that read the cross-section data cards, when it is desired to see if the quantities (e.g., material name, special character, etc.) on each set of cards of the same TYPE are identical.

The quantity IPARAM2 is compared against each of IPARAM1(1),..., IPARAM1(K).

ERFLAG is set nonzero if an inequality is detected.

COMPARES


START
1 → I
IPARAM1(I) = IPARAM2 ?
no
yes
I+1 → I
1 → ERFLAG
I > K ?
no
yes
RETURN

DELETSET

This routine is called from SCAN2 when a control card of the type *DELETE SET is encountered. The set whose name is found on the card is deleted from the new library.

1. SCANCARD reads the set name on the *DELETE SET card and places it in FIELD(2).

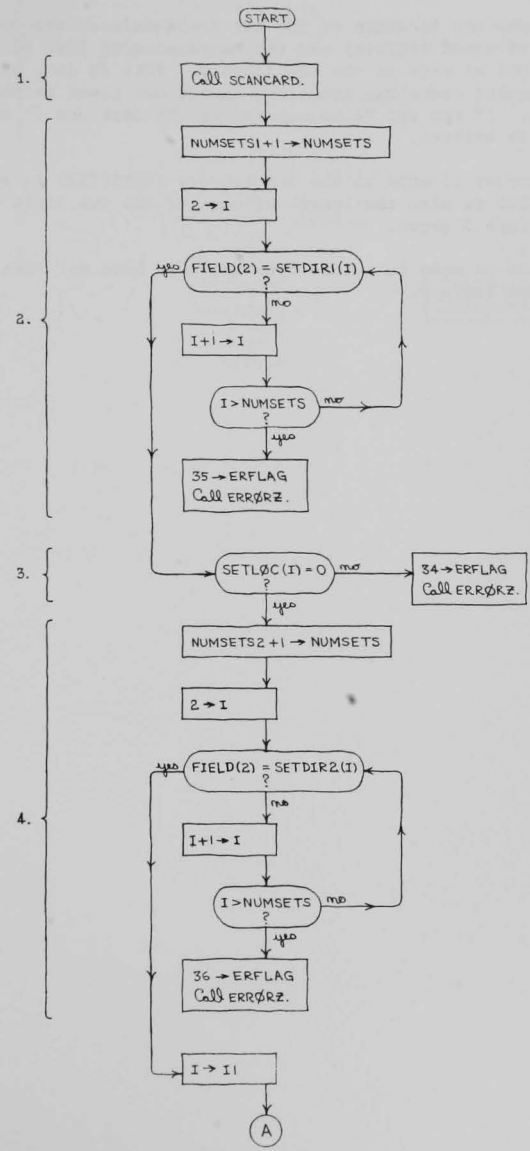
2. The elements 2 through NUMSETS of the array SETDIR1 contain the list of names of sets appearing on the old library.

   The name of the set to be deleted must correspond to some set in the old library.

3. If SETLOC(I) is nonzero, then the set to be deleted has already been transferred to the new library; i.e., a *MOD SET card for this set has been previously encountered. This is illegal usage of the control cards.

4. The elements 2 through NUMSETS of the array SETDIR2 contain the list of names of sets that are to appear on the new library. The name of the set to be deleted must appear somewhere in this list. Its absence from the new library and its presence in the old library (see paragraph 2 above) indicate that the set has already been deleted from the new library; i.e., a *DELETE SET card for this set has been previously encountered. This is illegal usage of the control cards.

DELETSET

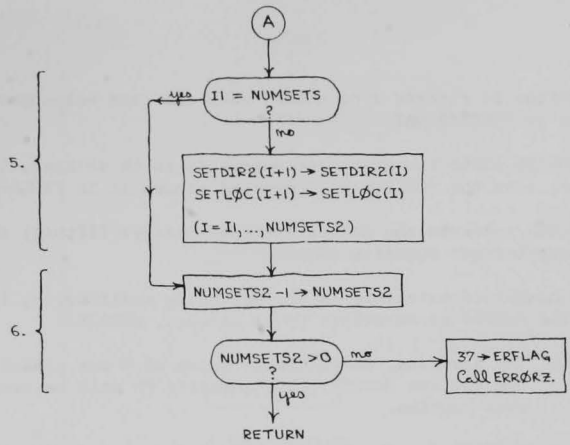

START
1.
Call SCANCARD.
2.
NUMSETS1+1 → NUMSETS
2 → I
FIELD(2) = SETDIR1(I) ?
yes
no
I+1 → I
I > NUMSETS ?
no
yes
35 → ERFLAG
Call ERRØRZ.
3.
SETLØC(I) = 0 ?
no
34 → ERFLAG
Call ERRØRZ.
yes
4.
NUMSETS2+1 → NUMSETS
2 → I
FIELD(2) = SETDIR2(I) ?
yes
no
I+1 → I
I > NUMSETS ?
no
yes
36 → ERFLAG
Call ERRØRZ.
I → I1
A

DELETSET (Contd.)

5. The name and location of the set to be deleted are removed from the list of names (SETDIR) and the corresponding list of locations (SETLOC) of sets on the new library. This is done by relocating all subsequent names and locations upward one place in the respective lists. If the set to be deleted is the last one in the list, this step is omitted.

6. The number of sets in the new library (NUMSETS2) is reduced by one; NUMSETS2 is also the length of each of the two lists mentioned in paragraph 5 above.

   A check is made to insure that all sets have not been deleted from the new library.


A
I1 = NUMSETS
?
yes
no
5.
SETDIR2(I+1) → SETDIR2(I)
SETLØC(I+1) → SETLØC(I)
(I = I1,...,NUMSETS2)
6.
NUMSETS2 -1 → NUMSETS2
NUMSETS2 > 0
?
no
37 → ERFLAG
Call ERRØR2.
yes
RETURN

DELMATS

This routine is entered from MODSET when the card encountered following a *MOD SET card is *DELETE MAT.

1. SCANCARD picks the number of materials to be deleted, in Hollerith code, from the *DELETE MAT card and places it in FIELD(2).

   BCD2BIN converts the Hollerith-coded integer FIELD(2) into the binary-integer quantity NUMATS.

2. The number of materials in the set being modified, M, is reduced by the number of materials to be deleted, NUMATS.

   NOTE: The previous, undiminished value of M was placed in PM in the routine MODSET; this quantity PM will be used later in this routine.

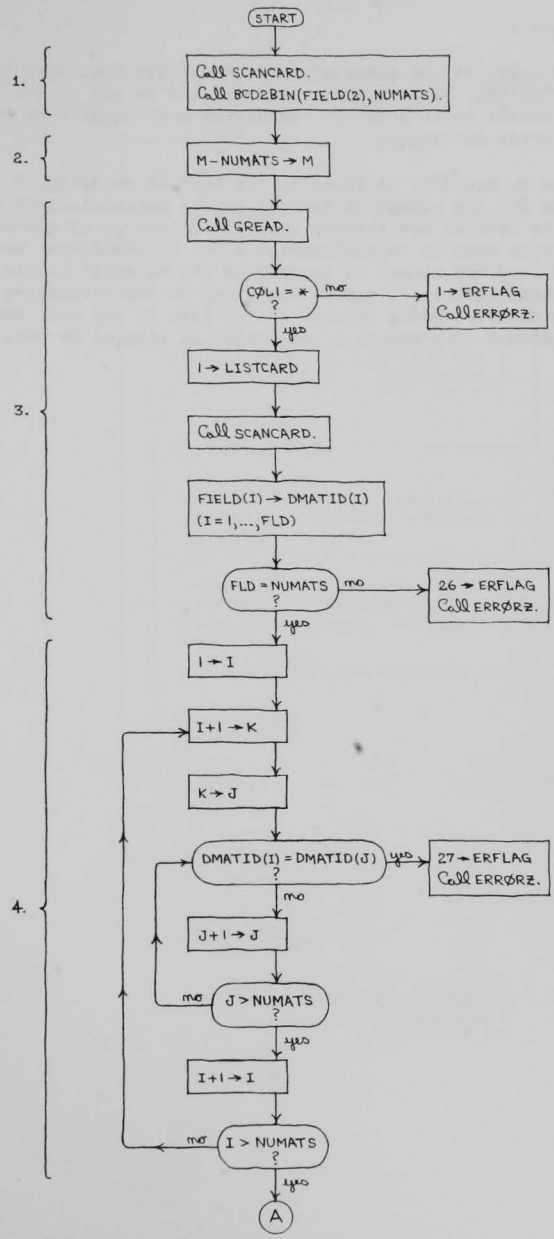
3. The card following the *DELETE MAT card is examined. It should be a list card containing the names of the materials that are to be deleted from the set.

   LISTCARD is a flag used by the routine SCANCARD.

   The number of names on the list-card(s) is placed in the quantity FLD. The names themselves are obtained by SCANCARD, placed in FIELD(1),...,FIELD(FLD), and then transferred to DMATID(1),..., DMATID(FLD).

   The number of names on the card (FLD) must be the same as the number of materials to be deleted (NUMATS), as specified on the *DELETE MAT card.

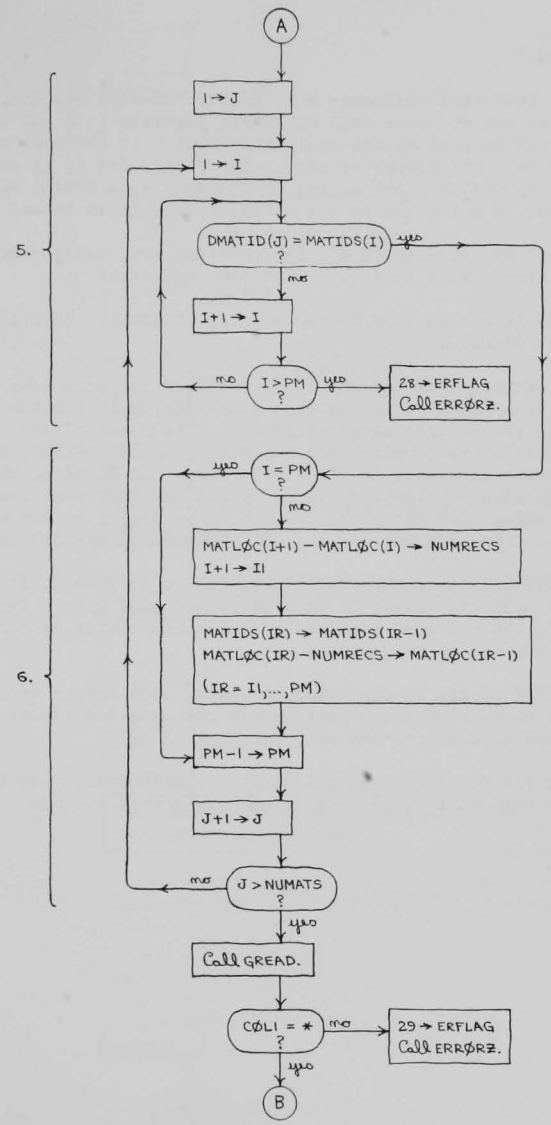
4. The quantities DMATID(I) are compared with each other to insure that there are no duplications.


START
1.
Call SCANCARD.
Call BCD2BIN(FIELD(2), NUMATS).
2.
M-NUMATS → M
3.
Call GREAD.
COL1 = * ?
no
1 → ERFLAG
Call ERRØRZ.
yes
1 → LISTCARD
Call SCANCARD.
FIELD(I) → DMATID(I)
(I = 1,..., FLD)
FLD = NUMATS ?
no
26 → ERFLAG
Call ERRØRZ.
yes
4.
1 → I
I+1 → K
K → J
DMATID(I) = DMATID(J) ?
yes
27 → ERFLAG
Call ERRØRZ.
no
J+1 → J
J > NUMATS ?
no
yes
I+1 → I
I > NUMATS ?
no
yes
A

DELMATS (Contd.)

5. The names of the materials in the set are contained in MATIDS(1),..., MATIDS(PM). Each quantity is compared to all of the names of the materials to be deleted. DMATID(J) must equal some MATIDS(I), or an error is flagged.

6. When an equality is found in the test of paragraph 5 above, the name and the number of records in the material block of the material to be deleted are removed from the directory of materials of the set. This is done by relocating the names of subsequent materials in the set, and the number of records in the material blocks of these subsequent materials, upward one place in the respective lists. (If the material being deleted is the last in the set, this step is omitted.) The length of each list is reduced by one.


A
1 → J
1 → I
5.
DMATID(J) = MATIDS(I) ?
yes
no
I+1 → I
no
I > PM ?
yes
28 → ERFLAG
Call ERRØRZ.
yes
I = PM ?
no
6.
MATLØC(I+1) − MATLØC(I) → NUMRECS
I+1 → I1
MATIDS(IR) → MATIDS(IR−1)
MATLØC(IR) − NUMRECS → MATLØC(IR−1)
(IR = I1,...,PM)
PM−1 → PM
J+1 → J
no
J > NUMATS ?
yes
Call GREAD.
CØL1 = * ?
no
29 → ERFLAG
Call ERRØRZ.
yes
B

DELMATS (Contd.)

7. The input card following the *DELETE MAT card and its list card may or may not be a *ADD MAT card; the flag L is set accordingly (L will be used by the routine MODSET). If the next card is *ADD MAT, the number of materials to be added is obtained from the card by SCANCARD and placed in NUMOFMAT by BCD2BIN; M, the net number of materials in the modified set, is increased by NUMOFMAT.

8. M must be positive; i.e., the modified set, after deletions and/or additions, must contain one or more materials.

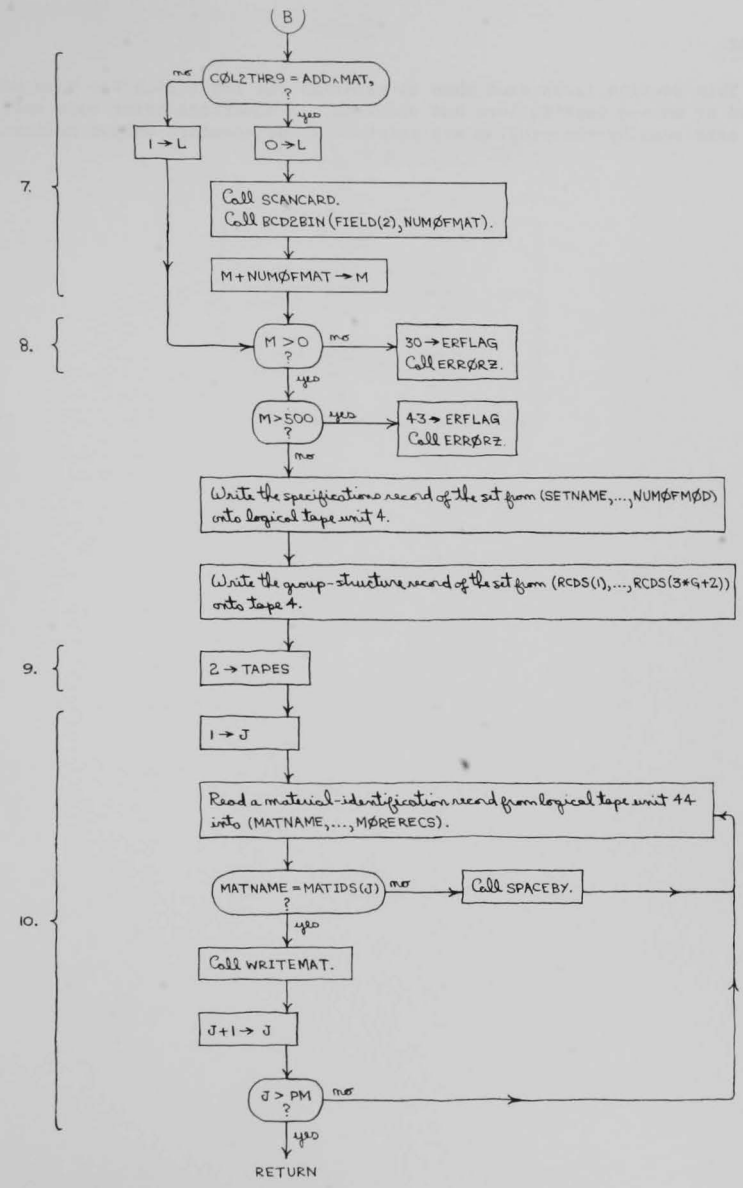
9. TAPES is a flag used by the routing WRITEMAT. (See also paragraph 10 below.)

10. The materials in the set being modified are read into core one-by-one from the old library (the material-identification record is the first record in a material block). The names of the nondeleted materials are contained in MATIDS(1),...,MATIDS(PM), in the order of their occurence in the old library. If MATNAME, the name of the material whose identification record has just been read, is one of the nondeleted materials that is to be placed on the new library, the routine WRITEMAT is called. Otherwise, SPACEBY is called.

    WRITEMAT reads a material block, record by record, from one logical tape unit and writes it on a different tape unit. For TAPES equal to 2, the material block is read from tape unit 44 (the old library) and written on tape unit 6.

    SPACEBY spaces by one material block; upon return from SPACEBY, tape unit 44 is positioned just before the material-identification record that starts the next material block.

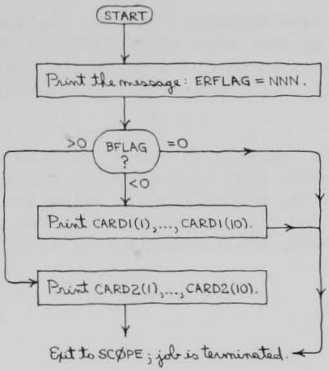
    When all the materials MATIDS(1),...,MATIDS(PM) have been written onto tape unit 6, an exit is made from this routine.


B
CØL2THR9 = ADDΔMAT. ?
no
yes
1 → L
0 → L
7.
Call SCANCARD.
Call BCD2BIN (FIELD(2), NUMØFMAT).
M + NUMØFMAT → M
8.
M > 0 ?
no
30 → ERFLAG
Call ERRØRZ.
yes
M > 500 ?
yes
43 → ERFLAG
Call ERRØRZ.
no
Write the specifications record of the set from (SETNAME, ..., NUMØFMØD) onto logical tape unit 4.
Write the group-structure record of the set from (RCDS(1), ..., RCDS(3*G+2)) onto tape 4.
9.
2 → TAPES
10.
1 → J
Read a material-identification record from logical tape unit 44 into (MATNAME, ..., MØRERECS).
MATNAME = MATIDS(J) ?
no
Call SPACEBY.
yes
Call WRITEMAT.
J + 1 → J
J > PM ?
no
yes
RETURN

ERRORZ

This routine is entered when an error in the input deck has been detected or when a tape failure has occurred. A numerical error code and the last card read by the program are printed on the standard output medium.

ERRØRZ

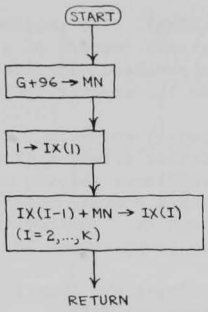

START
Print the message: ERFLAG = NNN.
BFLAG ?
>0
=0
<0
Print CARD1(1),...,CARD1(10).
Print CARD2(1),...,CARD2(10).
Exit to SCØPE; job is terminated.

FIXADD

FIXADD is entered from ADDAMAT. In this routine a set of indices IX(1),...,IX(G) is set up such that RCDS(IX(I)) is the first cell occupied by the Ith group-constants record of a material. IX(G + 1) is the index of the first unused cell in the array RCDS, following the last (G^{th}) group-constants record.

The number of cells reserved for each group-constants record, MN, represents the maximum size that such a record may attain. If the records do not need all this space, and additional cells are required for the storage of the scattering-data records of the material, the group-constants records are packed upwards in RCDS, and the indices IX(I) are reset by the routine TESTSTOR.

FIXADD



FIX2ADD

FIX2ADD is called by PNCH1MAT. The routine allocates storage in the array RCDS for the group-constants records of a material, the cross-section data cards of which are to be punched. It also sets up pointers that are used in allocating storage for the scattering-data records of the material.

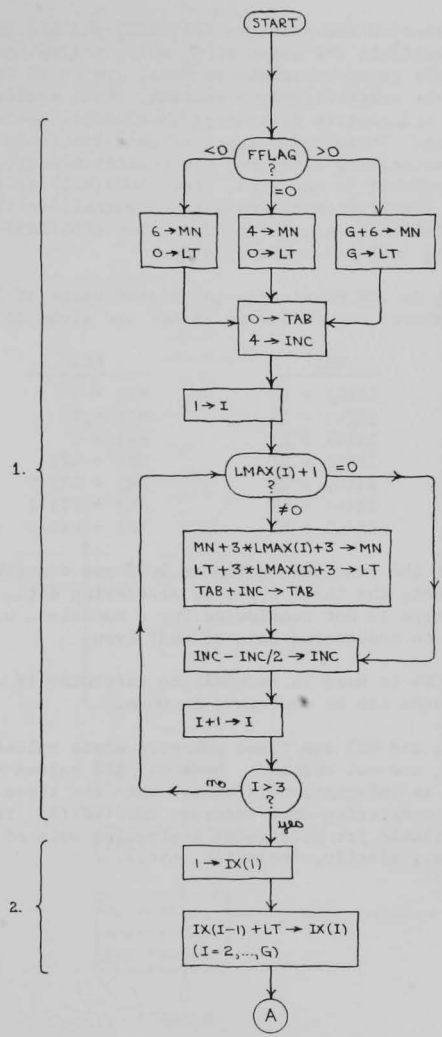
1. MN contains the number of words in a group-constant record of the material. The first four (FFLAG = 0) or six (FFLAG = 1 or -1) words of the group-constants records will not be stored; the type-1 data cards, which contain these initial words, are punched as the records are read in. The remaining LT words of each group-constants records are to be stored in RCDS.

   The calculations involving the quantities TAB and INC result in the following set of values for TAB:

TAB	Elastic Scattering?	Inelastic Scattering?	(n,2n) Scattering?
1	No	No	Yes
2	No	Yes	No
3	No	Yes	Yes
4	Yes	No	No
5	Yes	No	Yes
6	Yes	Yes	No
7	Yes	Yes	Yes

2. RCDS(IX(I)) is the first cell used for storage of (the last LT words of) the Ith group-constants record. There are G group-constants records.

FIX2ADD


START
FFLAG ?
<0
=0
>0
6 → MN
0 → LT
4 → MN
0 → LT
G+6 → MN
G → LT
0 → TAB
4 → INC
1 → I
1.
LMAX(I)+1 ?
=0
≠0
MN+3*LMAX(I)+3 → MN
LT+3*LMAX(I)+3 → LT
TAB+INC → TAB
INC - INC/2 → INC
I+1 → I
no
I > 3 ?
yes
1 → IX(1)
2.
IX(I-1) + LT → IX(I)
(I = 2,...,G)
A

FIX2ADD (Contd.)

3. In this section the pointers KK1, KK2, and KK3 are given values. The LT1 cells in the array RCDS, which follow the cells that are to contain the group-constants records, are to be used to store the data in the scattering-data records. Each scattering-data record contains information pertaining to elastic, inelastic, and (n,2n) scattering. These three types of data are to be separated; the elastic-scattering data from all scattering-data records are to be placed together in one area, etc. RCDS(KKI) is the first cell that contains elastic-scattering data, RCDS(KK2) is the first cell containing inelastic-scattering data, and RCDS(KK3) is the first cell containing (n,2n)-scattering data.

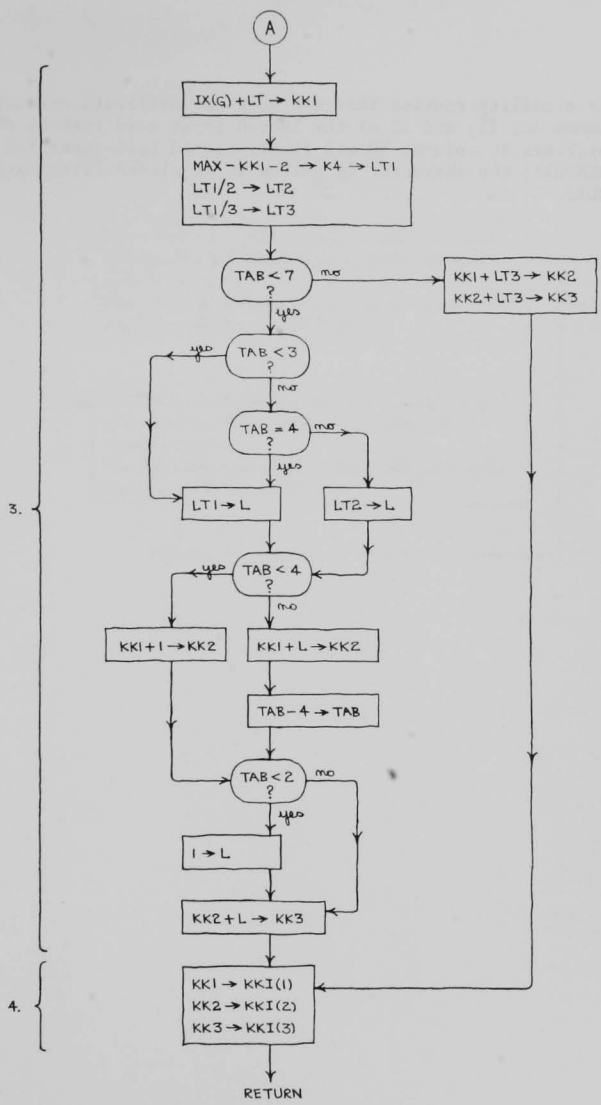
   Depending on the previously-calculated value of TAB (see paragraph 2 above) the following values are given to KK1, KK2, and KK3:

TAB	KK1	KK2	KK3
1	IX(G) + LT	KK1 + 1	KK2 + 1
2	IX(G) + LT	KK1 + 1	KK2 + LT1
3	IX(G) + LT	KK1 + 1	KK2 + LT1/2
4	IX(G) + LT	KK1 + LT1	KK2 + 1
5	IX(G) + LT	KK1 + LT1/2	KK2 + 1
6	IX(G) + LT	KK1 + LT1/2	KK2 + LT1/2
7	IX(G) + LT	KK1 + LT1/3	KK2 + LT1/3

   Note that the available cells in RCDS are distributed in an optimum manner among the three types of scattering data. If scattering of a certain type is not considered for a material, only a single cell is allotted to scattering data of that type.

   The flag K4 is used in PNCH1MAT to determine if all the scattering-data records can be contained in core.

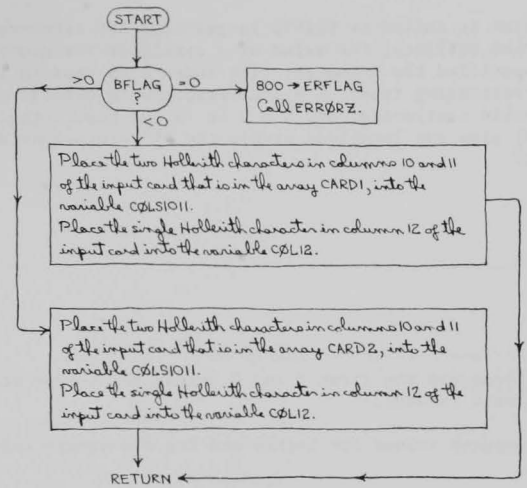
4. KK1, KK2, and KK3 are fixed pointers whose values, once set for a material, are not changed. However, the values of the KKI(I) are advanced as information is stored into the three regions of RCDS from the scattering-data records; RCDS(KKI(I)) is the first unused cell available for storage of scattering data of the Ith scattering type (i.e., elastic, inelastic, etc.).


A
IX(G) + LT → KK1
MAX - KK1 - 2 → K4 → LT1
LT1/2 → LT2
LT1/3 → LT3
TAB < 7 ?
no
KK1 + LT3 → KK2
KK2 + LT3 → KK3
yes
TAB < 3 ?
yes
no
TAB = 4 ?
no
yes
LT1 → L
LT2 → L
TAB < 4 ?
yes
no
KK1 + 1 → KK2
KK1 + L → KK2
TAB - 4 → TAB
TAB < 2 ?
no
yes
1 → L
KK2 + L → KK3
KK1 → KKI(1)
KK2 → KKI(2)
KK3 → KKI(3)
RETURN
3.
4.

GET1012

This is a utility routine that extracts the Hollerith characters that were in columns 10, 11, and 12 of the latest input card read by the program. The two characters in columns 10 and 11 are placed left-justified in the quantity COLS1011; the character in column 12 is placed left-justified in the quantity COL12.

GET1012


START
BFLAG ?
>0
=0
<0
800 → ERFLAG
Call ERRØRZ.
Place the two Hollerith characters in columns 10 and 11 of the input card that is in the array CARD1, into the variable CØLS1011.
Place the single Hollerith character in column 12 of the input card into the variable CØL12.
Place the two Hollerith characters in columns 10 and 11 of the input card that is in the array CARD2, into the variable CØLS1011.
Place the single Hollerith character in column 12 of the input card into the variable CØL12.
RETURN

GETSIZE

This routine is called by PRINT1 in printing the cross-section data of a material. Upon entrance, the value of J specifies the group number and the value L1 - 1 specified the ℓ-number. The index K as used in this routine specifies the scattering type: K = 1 corresponds to elastic scattering, K = 2 to inelastic scattering, and K = 3 to (n,2n) scattering. The arrays NF(K) and NG(K) give the locations within the Jth group-constants record for the material of

$$F^{(0)}_{K,J}$$

and

$$G^{(0)}_{K,J},$$

respectively (these are the first F and G values for K-type scattering in the Jth group-constants record).

GETSIZE computes values for I4(L1) and for the arrays I7(K) and I8(K), K = 1, 2, 3.

The value of I7(K) is

$$F^{(L1-1)}_{K,J},$$

and the value of I8(K) is

$$G^{(L1-1)}_{K,J}.$$

$$I4(L1) = 3 + \Sigma_{K=1}^{3}[I7(K) + I8(k)] \text{ if at least one I7(K) and I8(K) is nonzero;}$$

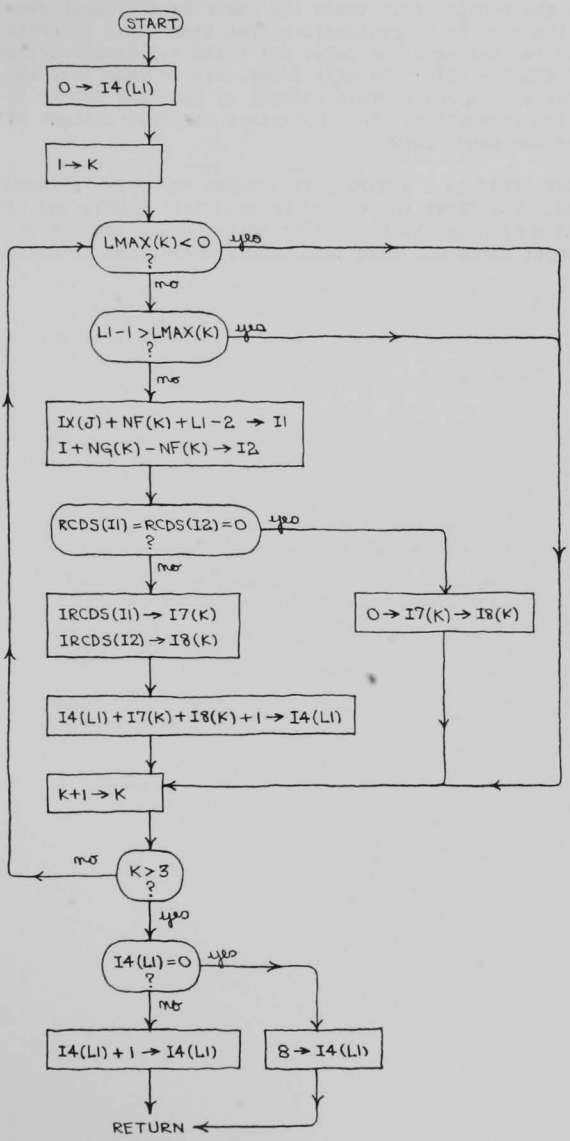
$$= 8 \text{ if all I7(K) and I8(K) are zero.}$$

In any case, I4(L1) represents the number of words in the L1-st scattering-data record that follows the Jth group-constants records of the material.

NOTE: If LMAX(K) < 0, neither of the quantities I7(K) and I8(K) are set to any value in GETSIZE, and so exit with the same values they posessed on entrance.

In certain subroutines, I7(K) and I8(K) are set to -1 before a call to GETSIZE is made. Upon return from GETSIZE, one or the other of I7(K) or I8(K) is tested; if it is still -1, this is used to signify that LMAX(K) < 0, i.e., there is no scattering information of any kind for scattering-type K.

GETSIZE

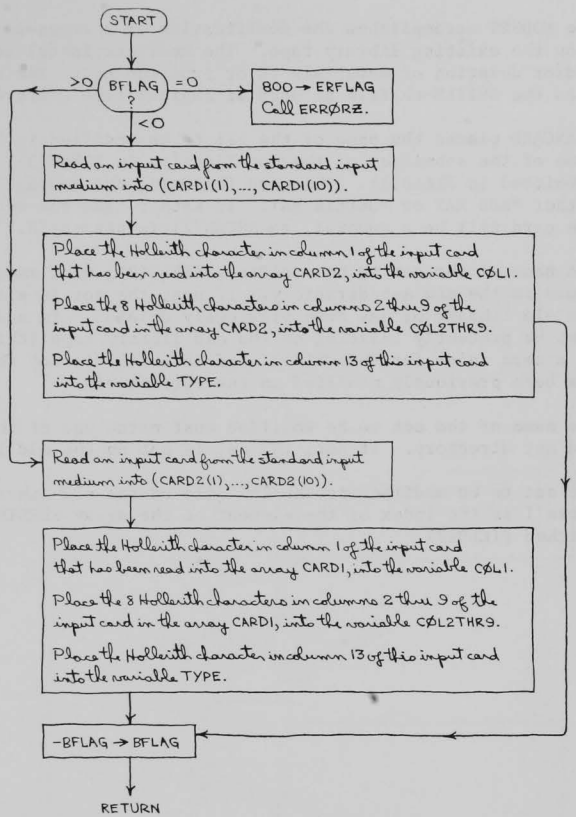

START
0 → I4(LI)
1 → K
LMAX(K) < 0 ?
yes
no
LI-1 > LMAX(K) ?
yes
no
IX(J) + NF(K) + LI - 2 → I1
I + NG(K) - NF(K) → I2
RCDS(I1) = RCDS(I2) = 0 ?
yes
no
IRCDS(I1) → I7(K)
IRCDS(I2) → I8(K)
0 → I7(K) → I8(K)
I4(LI) + I7(K) + I8(K) + 1 → I4(LI)
K+1 → K
K > 3 ?
no
yes
I4(LI) = 0 ?
yes
no
I4(LI) + 1 → I4(LI)
8 → I4(LI)
RETURN

GREAD

This is the routine that reads all cards in the input deck except the first one. The card to be processed at the time GREAD is called is already in one of the two buffer areas CARD1 (if BFLAG = 1 upon entrance to GREAD) or CARD2 (if BFLAG = -1). The next input card is read into the other of the two buffer areas, and certain columns of the card now to be processed are decoded for inspection. The characters in these columns will determine the nature of the input card.

Note that BFLAG is a binary switch whose value is "flipped" between one and minus one. The first input card is read into CARD1, and BFLAG is set to one in the main program, XLIBIT. Thereafter, under control of GREAD, the subsequent input cards are read into CARD2, CARD1, CARD2, CARD1, ..., etc.

GREAD


START
BFLAG ?
>0
=0
<0
800 → ERFLAG
Call ERRØRZ.
Read an input card from the standard input medium into (CARD1(1),..., CARD1(10)).
Place the Hollerith character in column 1 of the input card that has been read into the array CARD2, into the variable CØL1.
Place the 8 Hollerith characters in columns 2 thru 9 of the input card in the array CARD2, into the variable CØL2THR9.
Place the Hollerith character in column 13 of this input card into the variable TYPE.
Read an input card from the standard input medium into (CARD2(1),..., CARD2(10)).
Place the Hollerith character in column 1 of the input card that has been read into the array CARD1, into the variable CØL1.
Place the 8 Hollerith characters in columns 2 thru 9 of the input card in the array CARD1, into the variable CØL2THR9.
Place the Hollerith character in column 13 of this input card into the variable TYPE.
-BFLAG → BFLAG
RETURN

MODSET

Routine MODSET accomplishes the modification of a cross-section set that is already on the existing library tape. The modification(s) made are the addition and/or deletion of materials to or from the set. The modified set is written as the S6FILE-th file on each of logical tape units 4 and 6.

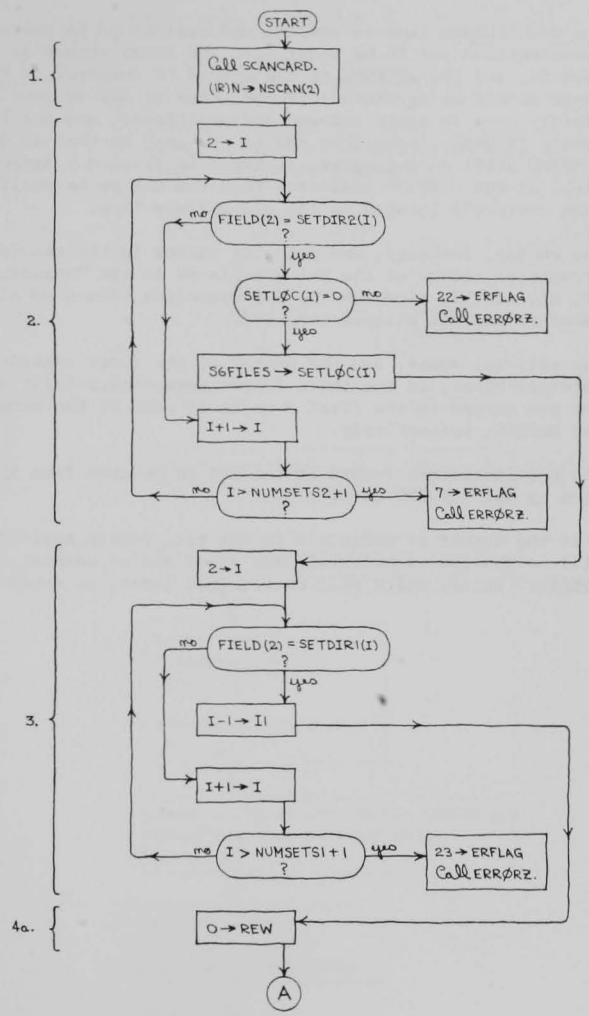
1. SCANCARD places the name of the set to be modified in FIELD(2), the name of the submitter of the modification in FIELD(3), and the date submitted in FIELD(4). The card following the *MODSET card will be either *ADD MAT or *DELETE MAT. In either case the second field on the card will be a numeral, so NSCAN(2) is set to "N."

2. The name of the set to be modified, FIELD(2), must match one of the names in the new set directory. If not, the set is either not in the old library or has been previously deleted. In addition, the set must be presently residing on the old library tape [this is implied by a zero value for SETLOC(I)]. If SETLOC(I) is not zero, the set has been previously modified in the current run.

3. The name of the set to be modified must match one of the names in the old set directory. If not, the set is not on the old library tape.

   The set to be modified is the Ith file on the old library tape, where I is the index of the element of the array SETDIR1 that matched FIELD(2).

MØDSET


START
1.
Call SCANCARD.
(IR)N → NSCAN(2)
2.
2 → I
FIELD(2) = SETDIR2(I)
?
no
yes
SETLØC(I) = 0
?
no
22 → ERFLAG
Call ERRØRZ.
yes
S6FILES → SETLØC(I)
I+1 → I
I > NUMSETS2 + 1
?
no
yes
7 → ERFLAG
Call ERRØRZ.
3.
2 → I
FIELD(2) = SETDIR1(I)
?
no
yes
I-1 → I1
I+1 → I
I > NUMSETS1 + 1
?
no
yes
23 → ERFLAG
Call ERRØRZ.
4a.
0 → REW
A

MODSET (Contd.)

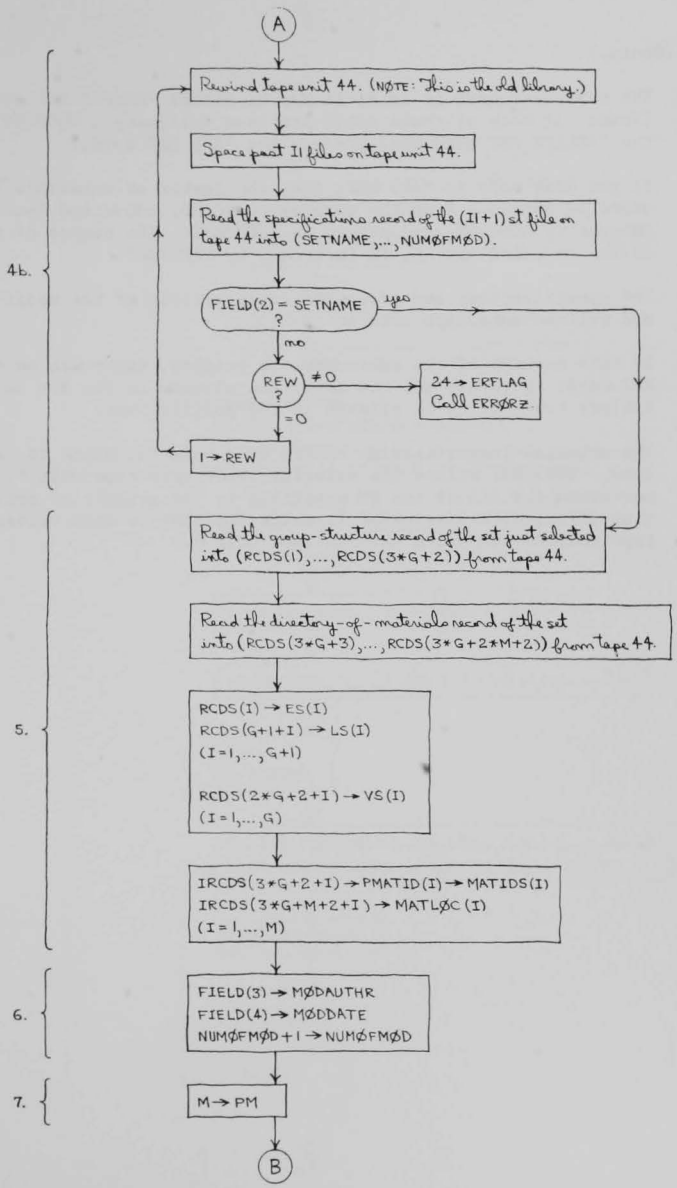
4a,b. The old library tape is rewound and positioned to the start of the cross-section set to be modified. The first record of this set is read in, and the SETNAME in the record is compared to FIELD(2). There should be agreement; if not, then if REW is zero, the old library tape is again rewound and positioned, and the initial record is read. Note that REW is now one, so that if FIELD(2) and SETNAME still do not agree, an error is flagged. Agreement of FIELD(2) and SETNAME indicates that the set to be modified has been correctly located on the old library tape.

5. The energy, lethargy, and velocity values in the second (group-structure) record of the set are placed in the "standard" arrays ES, LS, and VS, to which the corresponding values of all materials added to the set will be compared.

   The material names, and the number of the first record of each material block, in the third (directory-of-materials) record of the set are stored in the first M cells of each of the arrays MATIDS and MATLOC, respectively.

6. The specifications record of the set is updated from the information on the MOD SET card.

7. M is the number of materials in the set, before modification. M will be changed as materials are added and/or deleted, and its original value, which will be required later, is saved in PM.


A
Rewind tape unit 44. (NØTE: This is the old library.)
Space past II files on tape unit 44.
Read the specifications record of the (II+1) st file on tape 44 into (SETNAME,..., NUMØFMØD).
4b.
FIELD(2) = SETNAME ?
yes
no
REW ?
≠0
=0
24 → ERFLAG
Call ERRØRZ
1 → REW
Read the group-structure record of the set just selected into (RCDS(1),..., RCDS(3*G+2)) from tape 44.
Read the directory-of-materials record of the set into (RCDS(3*G+3),..., RCDS(3*G+2*M+2)) from tape 44.
5.
RCDS(I) → ES(I)
RCDS(G+1+I) → LS(I)
(I=1,...,G+1)
RCDS(2*G+2+I) → VS(I)
(I=1,...,G)
IRCDS(3*G+2+I) → PMATID(I) → MATIDS(I)
IRCDS(3*G+M+2+I) → MATLØC(I)
(I=1,...,M)
6.
FIELD(3) → MØDAUTHR
FIELD(4) → MØDDATE
NUMØFMØD+1 → NUMØFMØD
7.
M → PM
B

MODSET (Contd.)

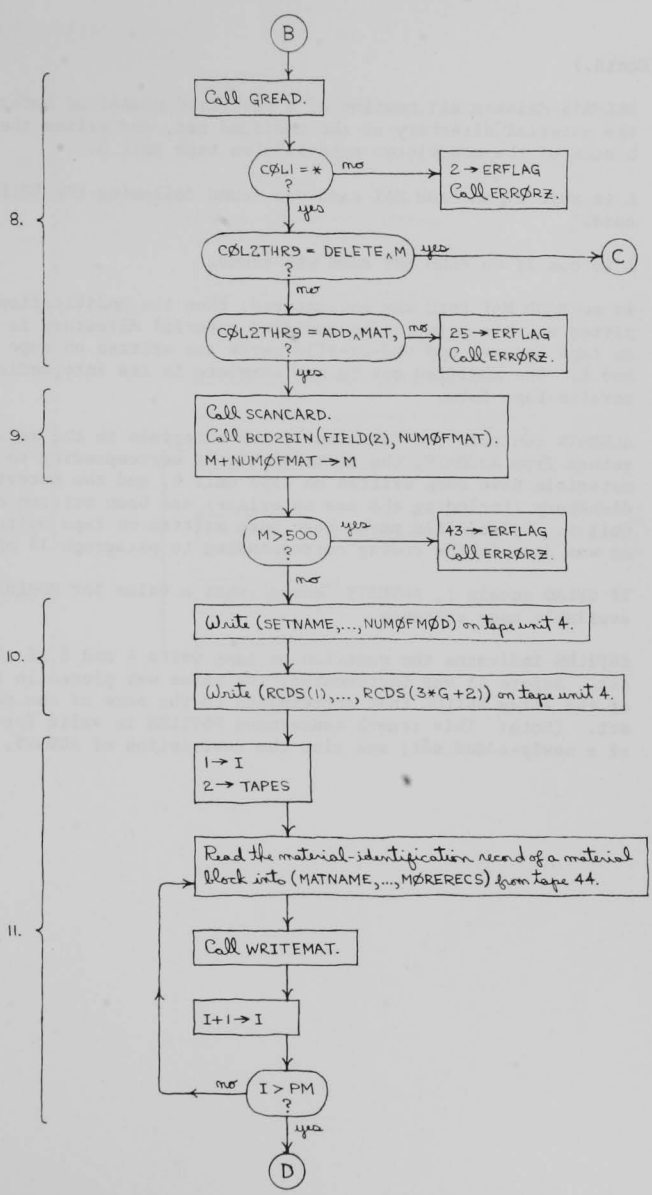
8. The next data card is read; it may be either *DELETE MAT or *ADD MAT (Note: If both of these cards are used following a *MOD SET card, the *DELETE MAT card must precede the *ADD MAT card.)

9. If the next card is *ADD MAT, then the number of materials to be added is obtained from the card by SCANCARD, converted into a binary integer by BCD2BIN, and placed in NUMOFMAT. The number of materials in the modified set, M, is increased by NUMOFMAT.

10. The specifications and group-structure records of the modified set are written onto tape unit 4.

11. If this section of the subroutine is reached, there was no *DELETE MAT card; therefore all the materials already in the set on the old library tape are to be present in the modified set.

    The material-identification record of a material block is read into core. WRITEMAT writes the material block onto tape unit 6. This is performed for all of the PM materials in the unmodified set. The flag TAPES, with a value of 2, tells WRITEMAT to read records from tape unit 44 and write them on tape unit 6.


B
Call GREAD.
CØL1 = * ?
no
2 → ERFLAG
Call ERRØRZ.
yes
CØL2THR9 = DELETE∧M ?
yes
C
no
CØL2THR9 = ADD∧MAT, ?
no
25 → ERFLAG
Call ERRØRZ.
yes
Call SCANCARD.
Call BCD2BIN (FIELD(2), NUMØFMAT).
M+NUMØFMAT → M
M > 500 ?
yes
43 → ERFLAG
Call ERRØRZ.
no
Write (SETNAME,..., NUMØFMØD) on tape unit 4.
Write (RCDS(1),..., RCDS(3*G+2)) on tape unit 4.
1 → I
2 → TAPES
Read the material-identification record of a material block into (MATNAME,..., MØRERECS) from tape 44.
Call WRITEMAT.
I+1 → I
I > PM ?
no
yes
D
8.
9.
10.
11.

MODSET (Contd.)

12. DELMATS deletes all mention of a specified number of materials from the material directory of the modified set, and writes the material blocks of the nondeleted materials on tape unit 6.

13. L is zero if an *ADD MAT card was found following the DELETE MAT card.

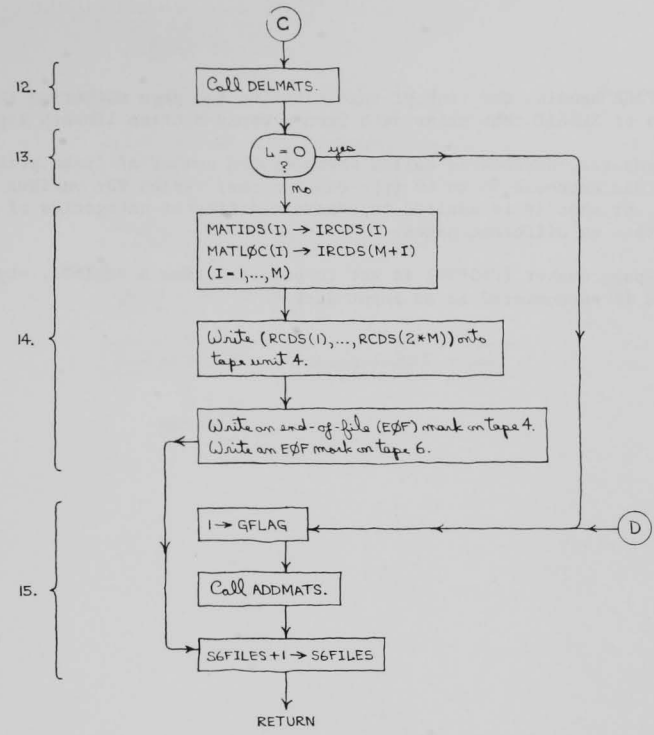
    L is one if no *ADD MAT card was found.

14. If no *ADD MAT card was encountered, then the modification is completed at this point. The modified material directory is written on tape unit 4, and end-of-file marks are written on tape units 4 and 6. The modified set is now complete in its intermediate scratch-tape form.

15. ADDMATS controls the addition of new materials to the set. Upon return from ADDMATS, the material blocks corresponding to the new materials have been written on tape unit 6, and the material directory (including the new materials) has been written on tape unit 4. End-of-file marks have been written on tape units 4 and 6, as was done in the coding corresponding to paragraph 14 above.

    If GFLAG equals 1, ADDMATS "knows" that a value for MORERECS is available upon entrance.

    S6FILES indicates the position on tape units 4 and 6 of the modified set. Before it was incremented, its value was placed in that entry of the array SETLOC that corresponds to the name of the modified set. (Note: This remark concerning S6FILES is valid for the case of a newly-added set; see also the description of ADDSET, pp. 96-97.)


C
12.
Call DELMATS.
13.
L = 0 ?
yes
no
14.
MATIDS(I) → IRCDS(I)
MATLØC(I) → IRCDS(M+I)
(I = 1,..., M)
Write (RCDS(1),..., RCDS(2*M)) onto tape unit 4.
Write an end-of-file (EØF) mark on tape 4.
Write an EØF mark on tape 6.
15.
1 → GFLAG
D
Call ADDMATS.
S6FILES + 1 → S6FILES
RETURN

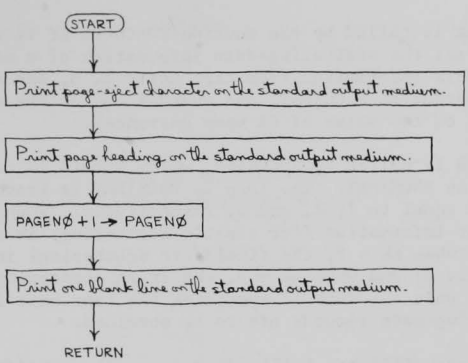
NEWPAGE

NEWPAGE handles the task of page ejecting and page numbering for all functions of XLIBIT that print data from a cross-section library tape.

In general, NEWPAGE is called whenever the number of lines printed on one page has exceeded 55 to 60 (the exact number varies for various types of data), or when it is desired to separate different categories of data by placing them on different pages.

The page number (PAGENO) is set to one each time a *PRINT...-type control card is encountered in an input deck.

NEWPAGE


START
Print page-eject character on the standard output medium.
Print page heading on the standard output medium.
PAGENØ + 1 → PAGENØ
Print one blank line on the standard output medium.
RETURN

PCHTP468

This routine is called by the routine PUNCH468 if it was not possible to contain in core all the scattering-data information of a material. The routine retrieves this information from tape units 2, 3, and 4 (where it was placed in the subroutine PNCH1MAT) and punches one of the card types-4, -6, or -8 (depending on the value of KK upon entrance).

1. PCHTP468 is called by PUNCH468, which itself is called from within a loop in PNCH1MAT. The loop in PNCH1MAT is traversed three times, with KK equal to 1, 2, and 3, respectively. When KK = 1, the relevant information (for elastic scattering) is contained on tape unit 2; when KK = 2, the (inelastic scattering) information is on tape unit 3; and when KK = 3, the (n,2n scattering) information is on tape unit 4. Thus I1 specifies the tape unit from which the scattering-data records are to be obtained.

2. The arrays NF(I) and NG(I), (I = 1,...,3), specify the location (within the group-constants record for group J) of $F^{(0)}_{I,J}$, and $G^{(0)}_{I,J}$, respectively, where I = 1 corresponds to elastic scattering, I = 2 to inelastic scattering, and I = 3 to (n,2n) scattering.

3. GETSIZE, when supplied with the group number J, the ℓ-number L1, and the arrays NF(I) and NG(I), (I = 1,...,3) returns values for I7(KK) and I8(KK). These quantities give the values of

   $$F^{(L1-1)}_{KK,J}$$

   and

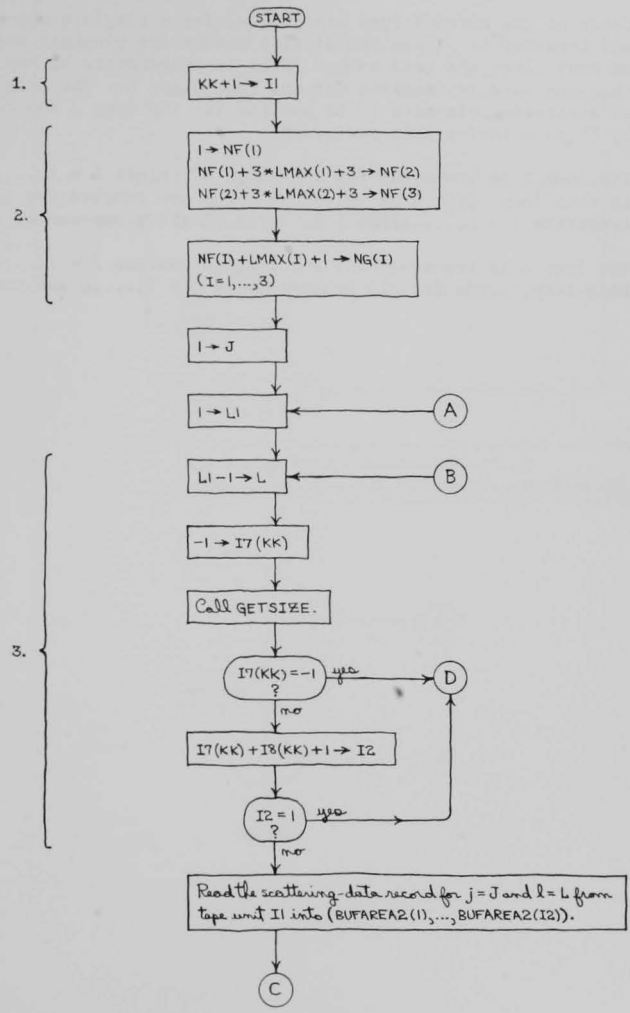
   $$G^{(L1-1)}_{KK,J},$$

   respectively; i.e., the number of groups into which upscattering and downscattering occur, for group J, an ℓ-value of L1 - 1, and the KK-th scattering type.

   If I7(KK) is -1 upon return from GETSIZE, then scattering of type KK and for an ℓ-number of L1 - 1 was not considered for any group for this material, and no corresponding scattering-data information exists. Similarly, if I7(KK) = I8(KK) = 0 (i.e., if I2 = 1), then there was no up- or downscattering for this group, ℓ-number, and scattering type, and again no corresponding scattering information exists.

   If I2 is greater than 1, then it represents the length of the record to be read from tape unit I1.

PCHTP468

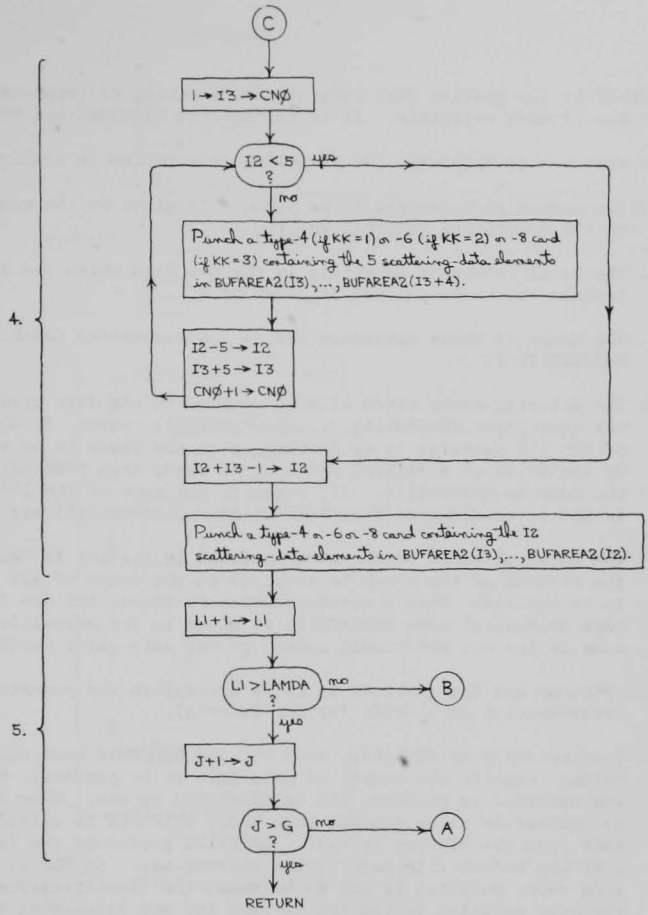

START
1.
KK+1 → I1
2.
1 → NF(1)
NF(1) + 3*LMAX(1) + 3 → NF(2)
NF(2) + 3*LMAX(2) + 3 → NF(3)
NF(I) + LMAX(I) + 1 → NG(I)
(I=1,...,3)
1 → J
1 → L1
A
3.
L1 - 1 → L
B
-1 → I7(KK)
Call GETSIZE.
I7(KK) = -1 ?
yes
D
no
I7(KK) + I8(KK) + 1 → I2
I2 = 1 ?
yes
no
Read the scattering-data record for j = J and l = L from tape unit I1 into (BUFAREA2(1),...,BUFAREA2(I2)).
C

PCHTP468 (Contd.)

4. Cards of the correct type are punched for a single group-number J and ℓ-number L. A maximum of five scattering elements are punched on each card; the card number (CNO) is incremented by one if more than one card is required for the same J and L. The total number of scattering elements to be punched for the same J and L is given by I2 on entering this paragraph.

5. The loop B is traversed for the range of values L = 1,...,LAMDA. In this loop, type-4 or -6, or -8 cards are punched for all ℓ-numbers ℓ = 0,...,LAMDA - 1, for a single group-number J.

   The loop A is traversed for the range of values J = 1,...,G. In this loop, cards for all group-numbers J = 1,...,G are punched.


C
1 → I3 → CNØ
I2 < 5 ?
yes
no
Punch a type-4 (if KK = 1) or -6 (if KK = 2) or -8 card (if KK = 3) containing the 5 scattering-data elements in BUFAREA2(I3), ..., BUFAREA2(I3 + 4).
I2 − 5 → I2
I3 + 5 → I3
CNØ + 1 → CNØ
I2 + I3 − 1 → I2
Punch a type-4 or -6 or -8 card containing the I2 scattering-data elements in BUFAREA2(I3), ..., BUFAREA2(I2).
4.
L1 + 1 → L1
L1 > LAMDA ?
no
B
yes
J + 1 → J
J > G ?
no
A
yes
RETURN
5.

PNCHMATS

PNCHMATS is the routine that controls the punching of cross-section data cards for one or more materials. It is called from PUNCHMAT and PUNCHSET.

Upon entrance to PNCHMATS, the following information is available:

a. The number of materials to be punched is given by the common value of the quantities NUMOFMAT and FLD.

b. The total number of materials in the set from which the FLD materials are to be punched is given by M.

c. The names of these materials are in the quantities MATABLE(1),..., MATABLE(FLD).

d. The material names which will be punched on the data cards are in the quantities NEWNAME(1),...,NEWNAME(FLD). Note: If the name of the Ith material is to be changed on the cards to be punched, by inclusion of a *RENAME MAT control card, then NEWNAME(I) is not the same as MATABLE(I). If, however, the name of the Ith material is not to be changed, then NEWNAME(I) and MATABLE(I) are identical.

1. The material block of the next material in the set is read in, and the MATNAME of the block is compared to the names of all materials to be punched. When a correspondence is found, the new (or perhaps unchanged) name NEWNAME(I) is given to the material. This name is the one which will appear on the data cards punched.

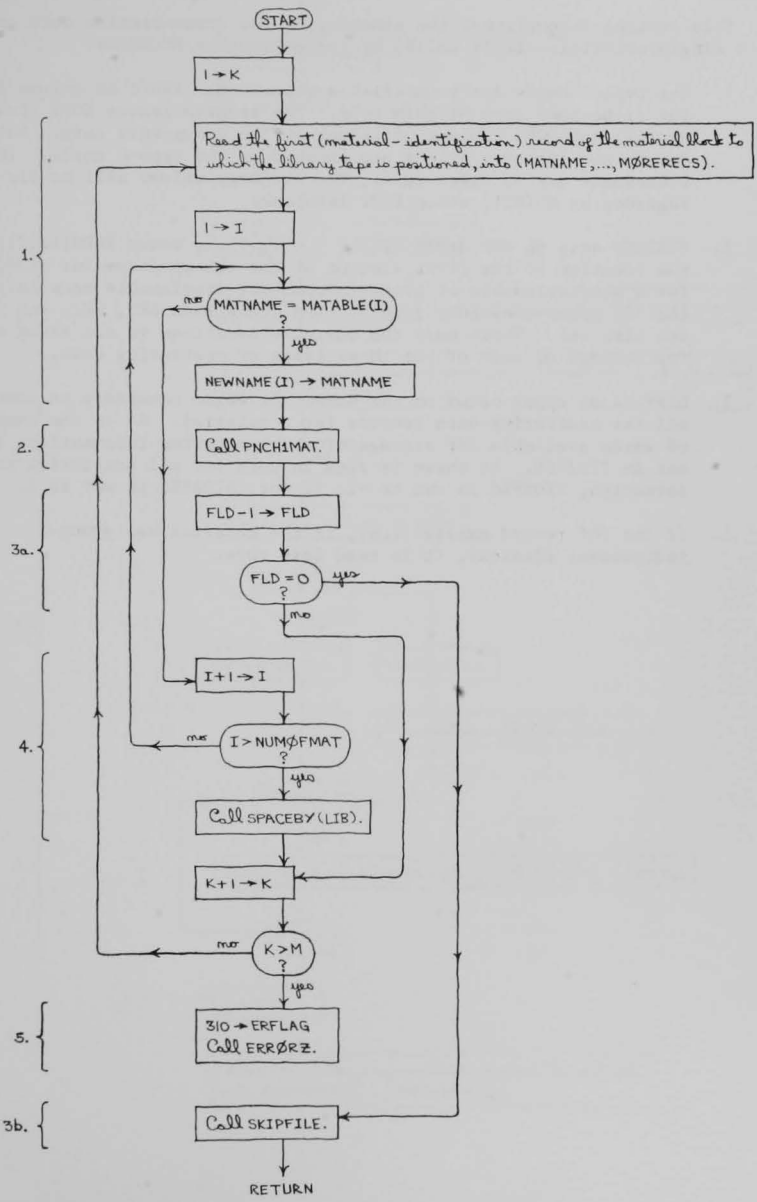
2. PNCH1MAT and the routines it calls accomplish the punching of the cross-section data cards for one material.

3a,b. Upon entrance to PNCH1MAT, both FLD and NUMOFMAT have the same value: namely, the number of materials to be punched. Each time one material is punched, FLD is decreased by one. When FLD is zero, all materials to be punched have been; SKIPFILE is called, and an exit from the routine is made. SKIPFILE positions the library tape past the end-of-file mark of the current set. If FLD is greater than zero, punching is yet to be done; the identification record of the next material in the set is read in, and processing continues.

4. If the current MATNAME does not correspond to the names of any of the materials to be punched, then the library tape is spaced past the remainder of the current material block; the identification record of the next material in the set is read in, and processing continues.

5. If the name of some material to be punched does not correspond to the name of any material in the set, then an error is flagged.

PNCHMATS


START
1 → K
Read the first (material-identification) record of the material block to which the library tape is positioned, into (MATNAME,..., MØRERECS).
1. {
1 → I
MATNAME = MATABLE(I) ?
no
yes
NEWNAME (I) → MATNAME
2. {
Call PNCH1MAT.
3a. {
FLD-1 → FLD
FLD = 0 ?
yes
no
4. {
I+1 → I
I > NUMØFMAT ?
no
yes
Call SPACEBY(LIB).
K+1 → K
K > M ?
no
yes
5. {
310 → ERFLAG
Call ERRØRZ.
3b. {
Call SKIPFILE.
RETURN

PNCH1MAT

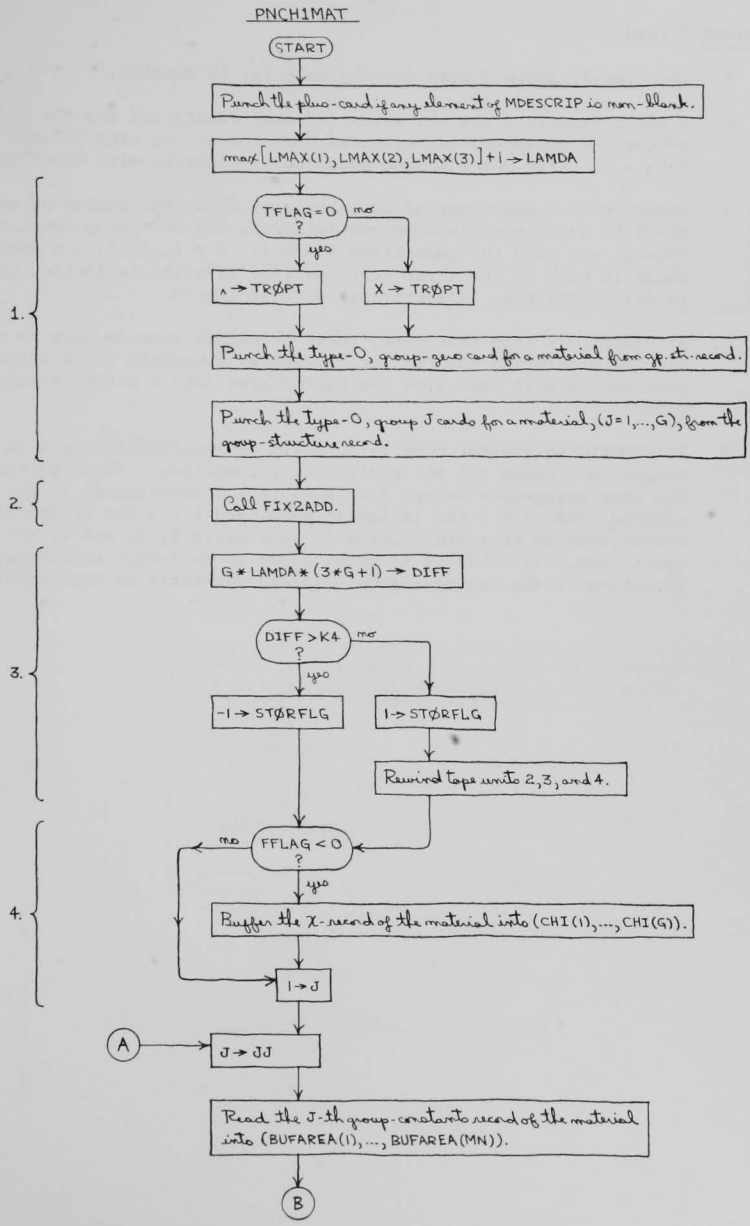
This routine accomplishes the punching of the cross-section data cards for a single material. It is called by the subroutine PNCHMATS.

1. The type-0 cards for a material are punched. TROPT is column 59 on the group-zero card of this type. The program leaves EOPT (column 57) and DOPT (column 58) blank on the group-zero card. Both energy and lethargy values are punched on the type-0 cards. (On subsequent use of these cards, the lethargy values will be disregarded by XLIBIT, since EOPT is blank).

2. FIX2ADD sets up the array IX(J), J = 1,...,G, where RCDS(IX(J)) is the location of the first element of the chi-spectrum (or of $F^{(0)}_{el,J}$ for a nonfissionable or group-independent fissionable material) in the Jth group-constants record. The quantities KK1, KK2, and KK3 are also set. These mark the starting locations in the array RCDS for storage of each of the three types of scattering data.

3. DIFF is an upper bound on the number of words necessary to contain all the scattering-data records for a material. K4 is the number of words available for storage of the scattering information; it is set in FIX2ADD. If there is room in core for all scattering information, STORFLG is set to -1; if not, STORFLG is set to 1.

4. If the chi record exists (i.e., if the material has group-independent fission), it is read into core.


PNCH1MAT
START
Punch the plus-card if any element of MDESCRIP is non-blank.
max[LMAX(1), LMAX(2), LMAX(3)] + 1 → LAMDA
TFLAG = 0 ?
no
yes
Λ → TRØPT
X → TRØPT
Punch the type-0, group-zero card for a material from gp. str. record.
Punch the type-0, group J cards for a material, (J = 1,..., G), from the group-structure record.
1.
Call FIX2ADD.
2.
G * LAMDA * (3 * G + 1) → DIFF
DIFF > K4 ?
no
yes
-1 → STØRFLG
1 → STØRFLG
Rewind tape units 2, 3, and 4.
3.
FFLAG < 0 ?
no
yes
Buffer the χ-record of the material into (CHI(1),..., CHI(G)).
4.
1 → J
A
J → JJ
Read the J-th group-constants record of the material into (BUFAREA(1),..., BUFAREA(MN)).
B

PNCH1MAT (Contd.)

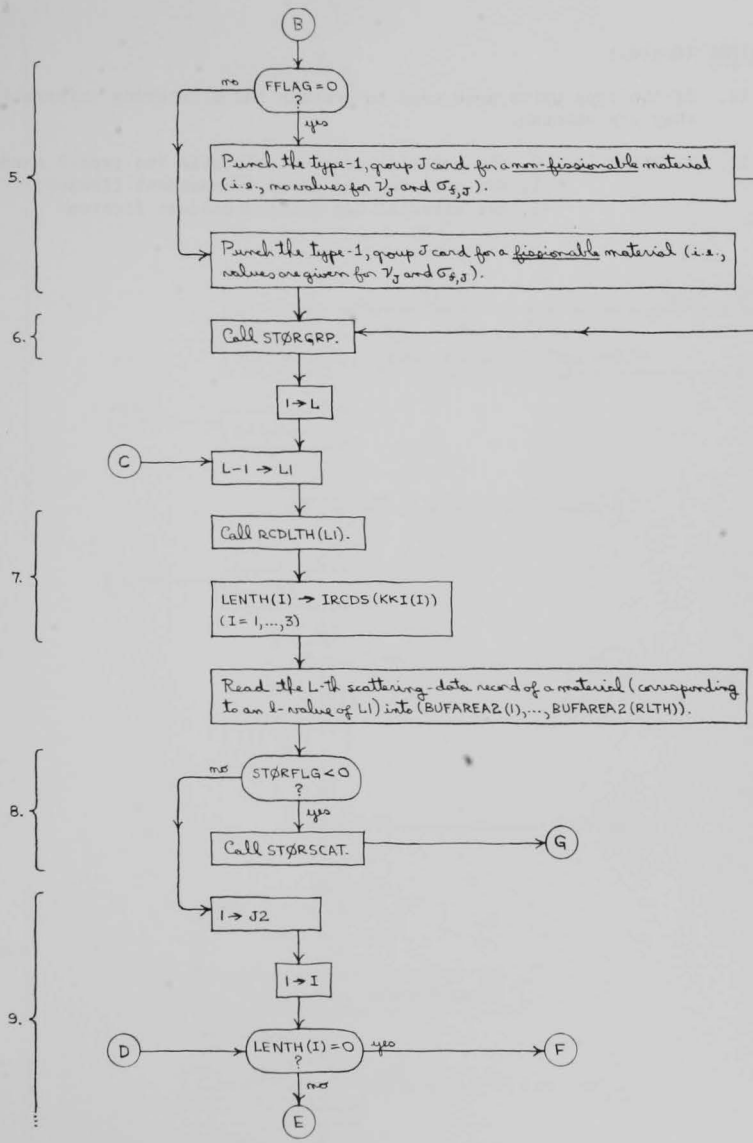
5. The type-1, group J card for the material is punched.

6. STORGRP transfers the chi words (if they exist) and the F's and G's of the Jth group-constants record from the buffer area BUFAREA into a block of words in the array RCDS, starting with RCDS(IX(J)).

7. RCDLTH with a parameter of L1 calculates RLTH, the number of words in the $L1^{st}$ scattering-data record following the Jth group-constants record, and also the quantities LENTH(I), I = 1, 2, 3, the number of words in each of the three portions (for elastic, inelastic, and (n,2n) scattering, respectively) of this record.

8. If STORFLG is less than zero, there is enough room in core to store all the scattering information. STORSCAT transfers the scattering-data record just read from the buffer area into a block of words in RCDS.

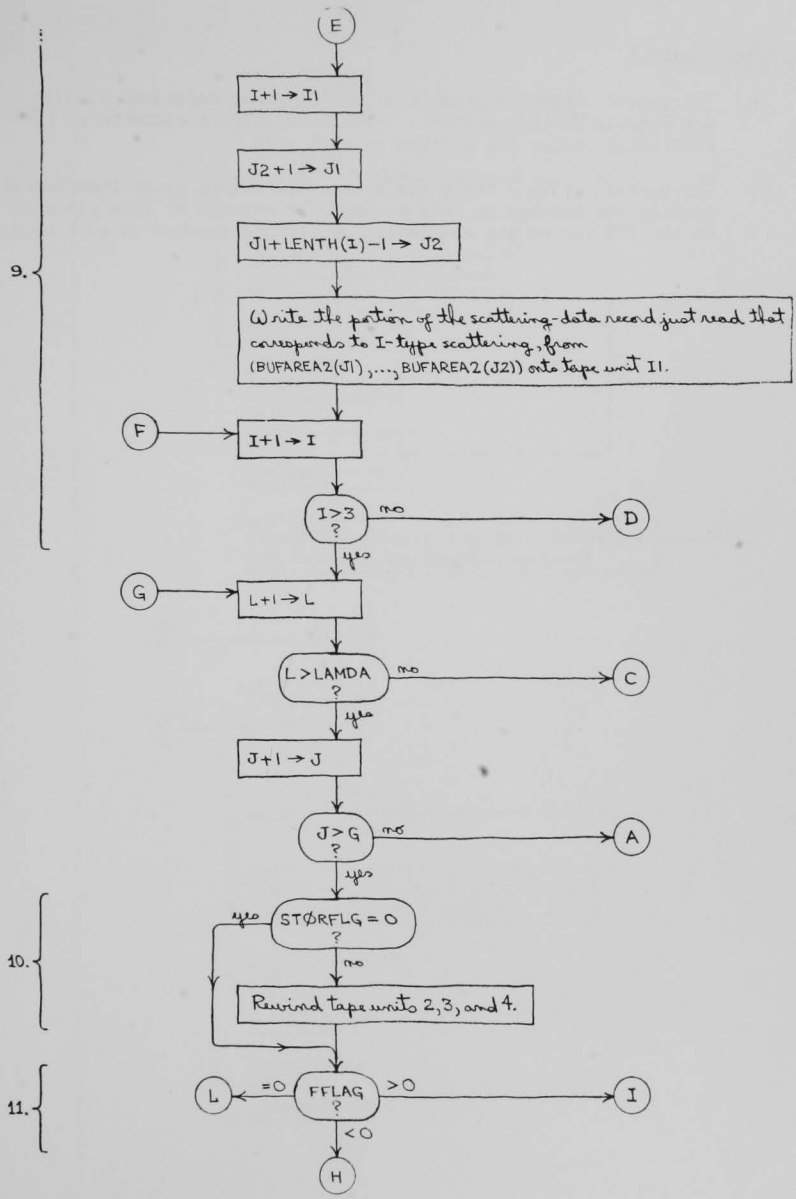
9. If STORFLG is greater than zero, there is insufficient room in memory to contain all the scattering information. Those portions of the scattering-data record just read (I = 1 corresponds to the elastic data, I = 2 the inelastic data, and I = 3 the (n,2n) data) are written as separate records on tape units 2, 3, and 4, respectively. If LENTH(I) is zero, there is no I-type scattering information in the scattering-data record currently in the buffer area.


B
FFLAG = 0 ?
no
yes
5.
Punch the type-1, group J card for a non-fissionable material (i.e., no values for ν_J and $\sigma_{f,J}$).
Punch the type-1, group J card for a fissionable material (i.e., values are given for ν_J and $\sigma_{f,J}$).
6.
Call STØRGRP.
1 → L
C
L-1 → L1
Call RCDLTH(L1).
7.
LENTH(I) → IRCDS(KKI(I))
(I = 1,...,3)
Read the L-th scattering-data record of a material (corresponding to an l-value of L1) into (BUFAREA2(1),..., BUFAREA2(RLTH)).
STØRFLG < 0 ?
no
yes
8.
Call STØRSCAT.
G
1 → J2
1 → I
9.
D
LENTH(I) = 0 ?
yes
F
no
E

PNCH1MAT (Contd.)

10. If the tape units were used to contain the scattering information, they are rewound.

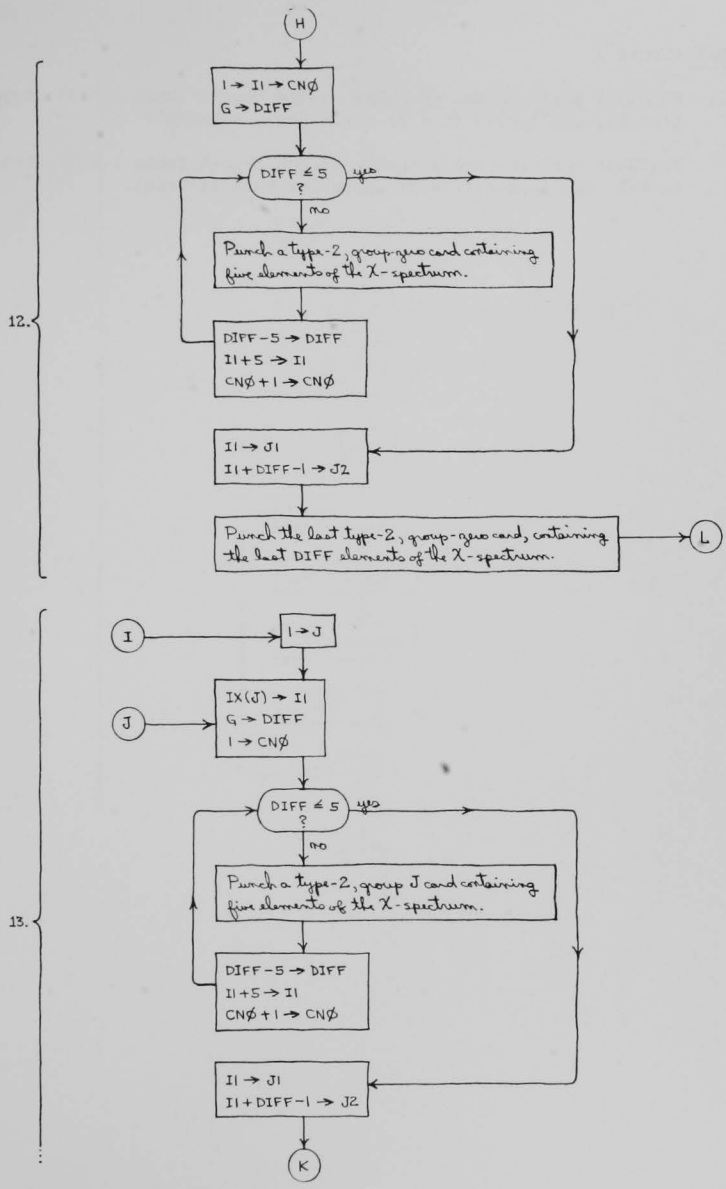
11. If FFLAG is: 0, the material is nonfissionable (no type-2 cards);
1, the material has group-independent fission;
-1, the material has group-dependent fission.


E
I+1 → I1
J2+1 → J1
J1+LENTH(I)−1 → J2
Write the portion of the scattering-data record just read that corresponds to I-type scattering, from (BUFAREA2(J1), ..., BUFAREA2(J2)) onto tape unit I1.
9.
F
I+1 → I
I>3 ?
no
D
yes
G
L+1 → L
L>LAMDA ?
no
C
yes
J+1 → J
J>G ?
no
A
yes
10.
yes
STØRFLG = 0 ?
no
Rewind tape units 2, 3, and 4.
11.
L
=0
FFLAG ?
>0
I
<0
H

PNCH1MAT (Contd.)

12. The type-2 cards for a material having group-dependent fission are punched in this section. A maximum of five elements of the scattering vector are punched on each card.

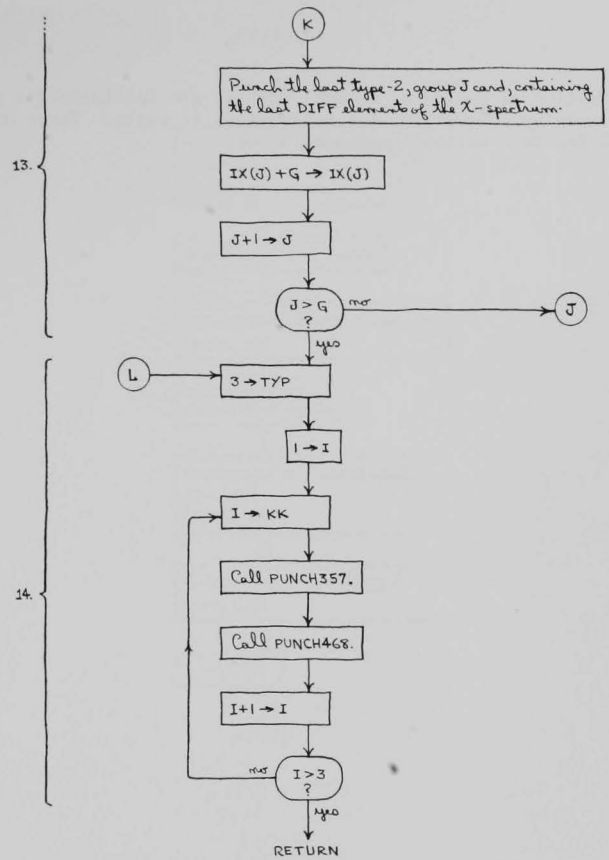
13. The type-2, group J cards for a material having group-independent fission are punched in this section. A maximum of five elements of the Jth row of the scattering matrix are punched on each card.


H
1 → I1 → CNØ
G → DIFF
DIFF ≤ 5 ?
yes
no
Punch a type-2, group-zero card containing five elements of the X-spectrum.
DIFF − 5 → DIFF
I1 + 5 → I1
CNØ + 1 → CNØ
I1 → J1
I1 + DIFF − 1 → J2
Punch the last type-2, group-zero card, containing the last DIFF elements of the X-spectrum.
L
12.
I
1 → J
J
IX(J) → I1
G → DIFF
1 → CNØ
DIFF ≤ 5 ?
yes
no
Punch a type-2, group J card containing five elements of the X-spectrum.
DIFF − 5 → DIFF
I1 + 5 → I1
CNØ + 1 → CNØ
I1 → J1
I1 + DIFF − 1 → J2
K
13.

PNCH1MAT (Contd.)

14. PUNCH357 performs the punching of the type-3 (when I = 1), type-5 (I = 2), and type-7 (I = 3) cards for a material.

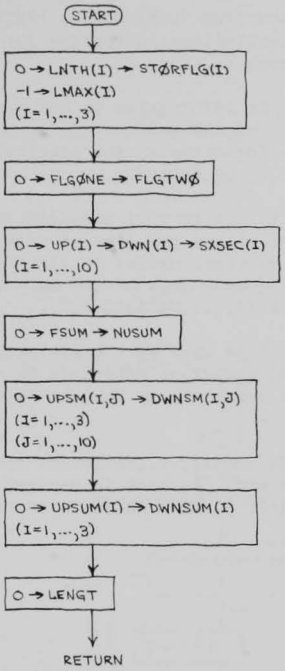
    PUNCH468 performs the punching of the type-4 (when I = 1), type-6 (I = 2) and type-8 (I = 3) cards for this material.


K
Punch the last type-2, group J card, containing the last DIFF elements of the X-spectrum.
13.
IX(J) + G → IX(J)
J+1 → J
J > G ?
no
J
yes
L
3 → TYP
1 → I
I → KK
Call PUNCH357.
14.
Call PUNCH468.
I+1 → I
no
I > 3 ?
yes
RETURN

PRESET

This routine is called by ADDAMAT. In it are initialized a number of quantities used by ADDAMAT and the subroutines it calls. These must be initialized for each material added to a set.

PRESET


START
0 → LNTH(I) → STØRFLG(I)
-1 → LMAX(I)
(I=1,...,3)
0 → FLGØNE → FLGTWØ
0 → UP(I) → DWN(I) → SXSEC(I)
(I=1,...,10)
0 → FSUM → NUSUM
0 → UPSM(I,J) → DWNSM(I,J)
(I=1,...,3)
(J=1,...,10)
0 → UPSUM(I) → DWNSUM(I)
(I=1,...,3)
0 → LENGT
RETURN

PRINTMAT

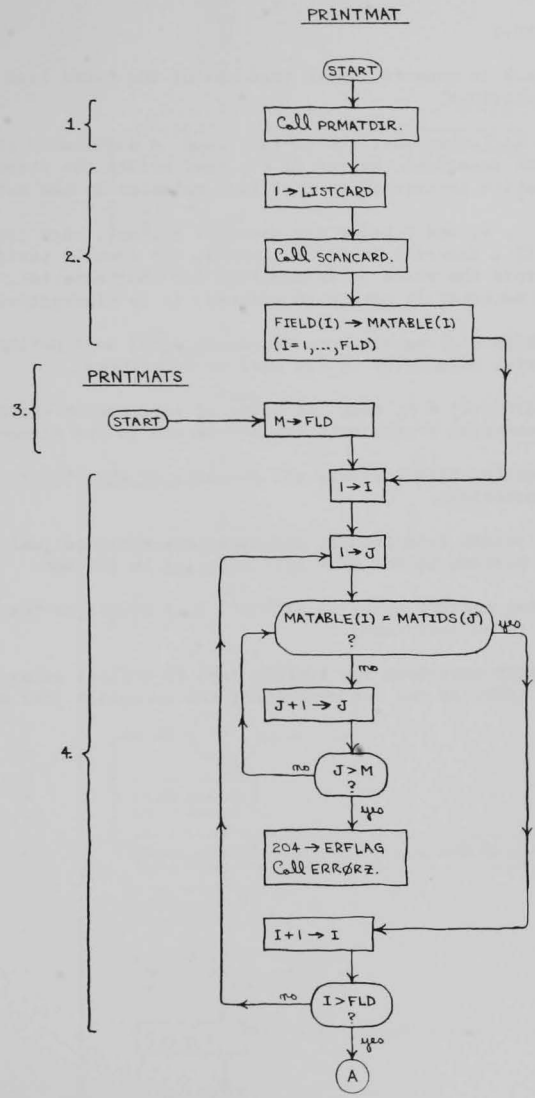
This routine is called from SCAN2 when a *PRINT MAT control card is encountered. It is the controlling subroutine for the print edit for one or more materials of a cross-section set.

1. PRMATDIR prints the information in the specifications and group-structure records of the set containing the materials to be printed, and prints the information in the material identification record of each material to be edited.

2. LISTCARD is set to one to tell SCANCARD that a list card is expected. SCANCARD places the list of material names on the list card (there are FLD such names) in FIELD(1),...,FIELD(FLD). This list, representing the names of the materials to be edited, is transferred to MATABLE(1),...,MATABLE(FLD).

3. The entry PRNTMATS is used by the subroutine PRINTSET. The quantity M gives the total number of materials in the set. Setting FLD to M implies that all materials in the set (i.e., the entire set) are to be printed.

4. The quantities MATIDS(1),...,MATIDS(M) contain the names of all materials in the set. A check is performed here to insure that each material MATABLE(I) to be printed is one of the MATIDS(J) materials in the set.


PRINTMAT
START
1.
Call PRMATDIR.
2.
1 → LISTCARD
Call SCANCARD.
FIELD(I) → MATABLE(I)
(I=1,...,FLD)
3.
PRNTMATS
START
M → FLD
4.
1 → I
1 → J
MATABLE(I) = MATIDS(J)
?
yes
no
J+1 → J
no
J > M
?
yes
204 → ERFLAG
Call ERRØRZ.
I+1 → I
no
I > FLD
?
yes
A

PRINTMAT (Contd.)

5. A check is made to insure that all of the names from the list card are distinct.

6. This indicates that the library tape is positioned just before the fourth record in the set (i.e., just before the start of the information pertaining to the first material in the set).

Sections 7, 8, and 9 below are repeated M times, once for each material in the set. If a material is to be printed, the tape is positioned to a point just before the block of information for that material, and PRINT1 is called. If a material is not to be printed, it is disregarded.

7. If MATABLE(J) equals some (does not equal any) MATIDS(I), then the material MATABLE(J) is (is not) to be printed.

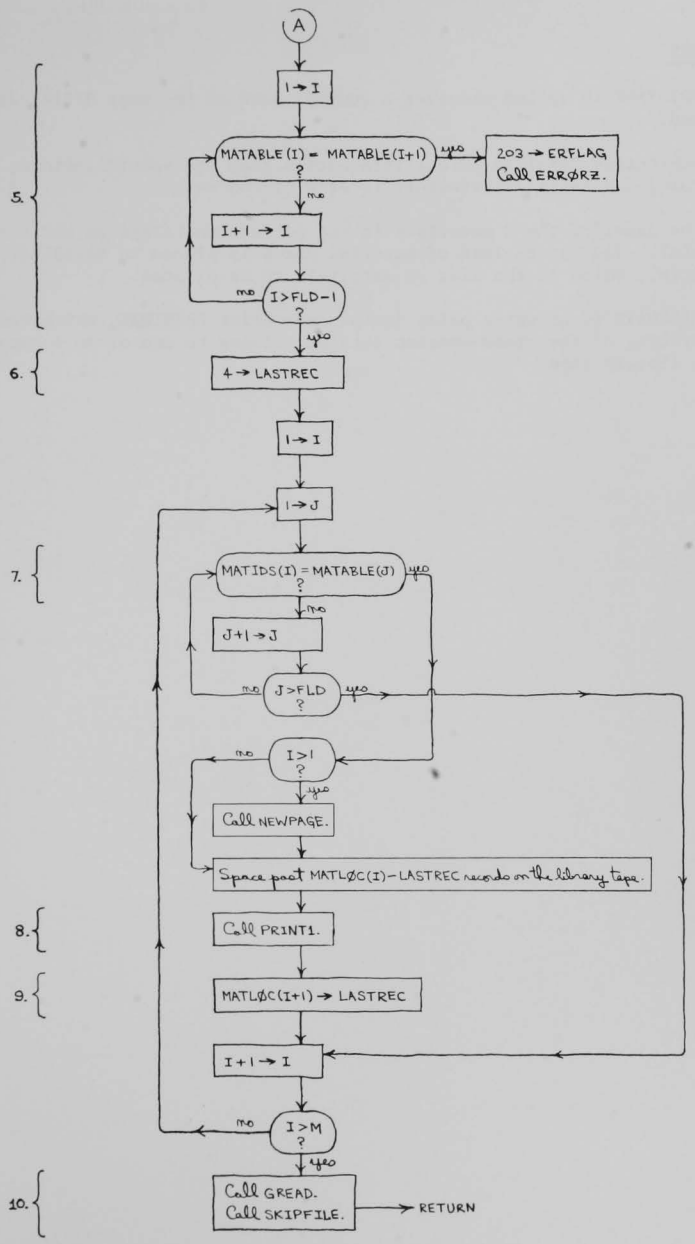
   If MATLOC(I) = N, then the block of information corresponding to the Ith material begins with the Nth record in the cross-section set.

8. Subroutine PRINT1 prints all the data on the library pertaining to one material.

9. Upon return from PRINT1, the tape is positioned just before the data that pertain to the (I + 1)st material in the set.

   Setting LASTREC equal to MATLOC(I + 1) specifies that the library tape is so positioned.

10. SKIPFILE positions the library tape to a point after the end-of-file (EOF) of the set containing the materials just edited.


A
1 → I
MATABLE(I) = MATABLE(I+1) ?
yes
203 → ERFLAG
Call ERRØRZ.
no
I+1 → I
no
I > FLD-1 ?
yes
5.
6.
4 → LASTREC
1 → I
1 → J
7.
MATIDS(I) = MATABLE(J) ?
yes
no
J+1 → J
no
J > FLD ?
yes
no
I > 1 ?
yes
Call NEWPAGE.
Space past MATLØC(I)-LASTREC records on the library tape.
8.
Call PRINT1.
9.
MATLØC(I+1) → LASTREC
I+1 → I
no
I > M ?
yes
10.
Call GREAD.
Call SKIPFILE.
RETURN

PRINTSET

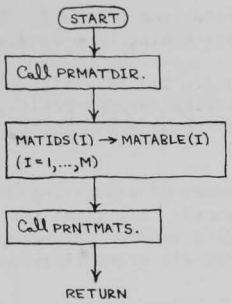
PRINTSET is called whenever a control card of the type *PRINT SET is detected.

Subroutine PRMATDIR prints information from the specifications, group-structure, and material-directory records of the set.

The names of the M materials in the set are contained in MATIDS(1),..., MATIDS(M). The entire list of material names is placed in MATABLE(1),..., MATABLE(M), which is the list of materials to be printed.

PRNTMATS is an entry point in the subroutine PRINTMAT, which controls the printing of the cross-section data pertaining to one or more materials on the library tape.

PRINTSET

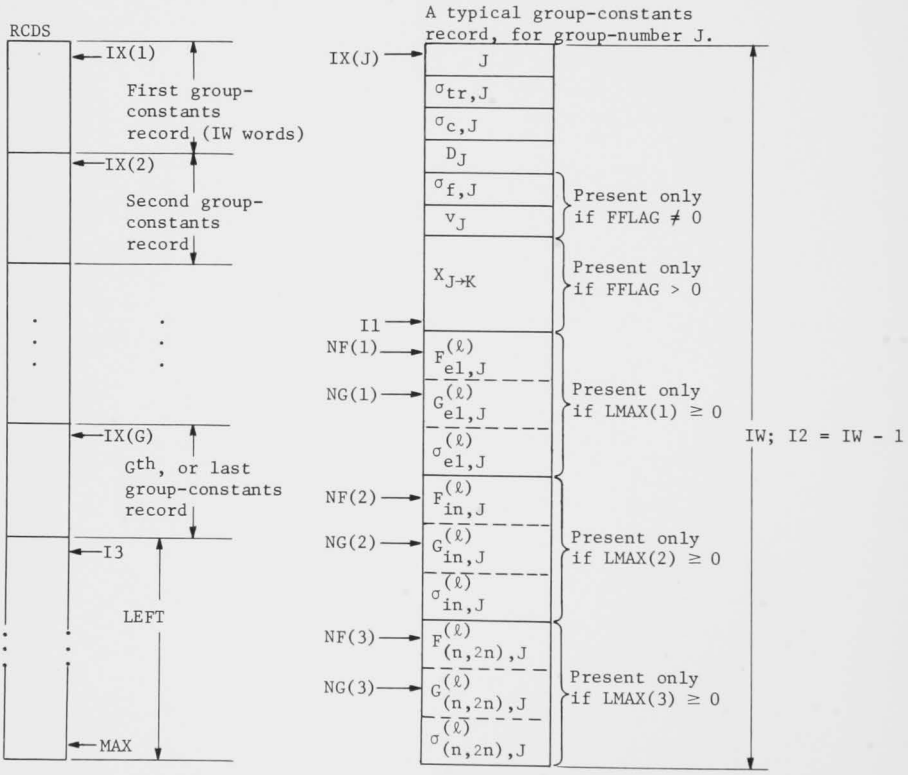

START
Call PRMATDIR.
MATIDS(I) → MATABLE(I)
(I = 1,...,M)
Call PRNTMATS.
RETURN

PRINT1

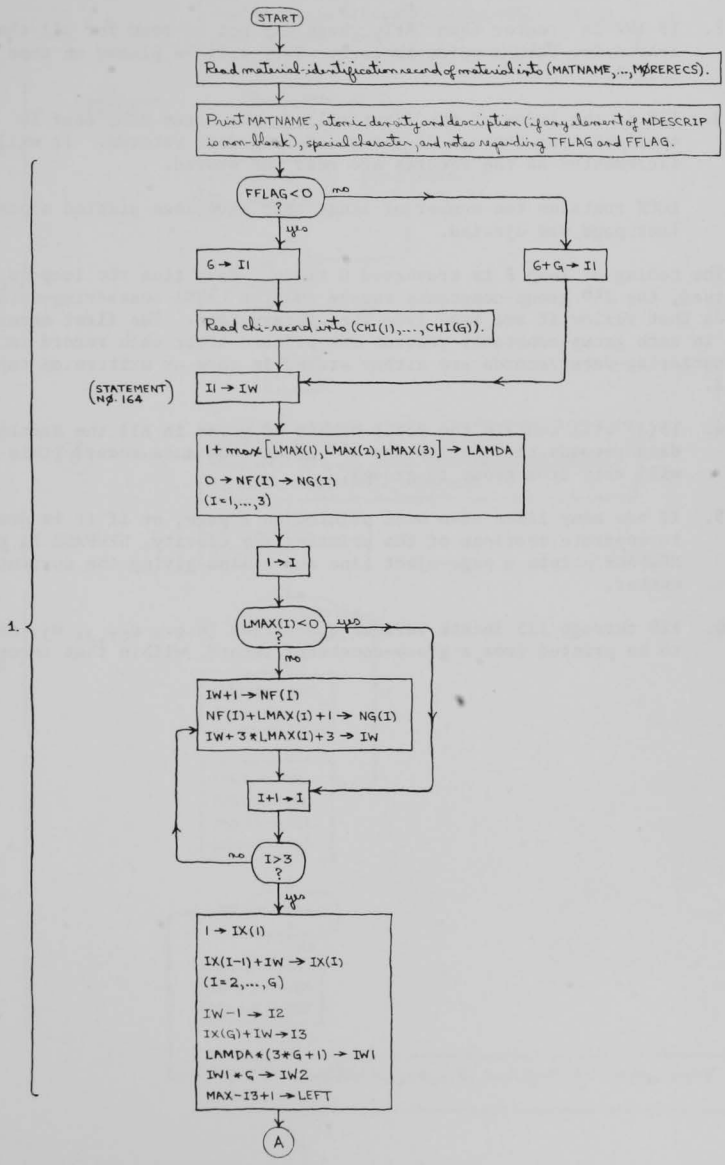
This subroutine is called from PRINTMAT. It accomplishes the printing of all cross-section data pertaining to a single material.

1. The quantities I1, I2, I3, IW, IX(I), NF(I), NG(I), MAX, and LEFT are pointers indicating various positions in the array RCDS where the group-constants records are stored. These are indicated in the diagram below.

   LAMDA gives the number of scattering-data records following each group-constants record. IW1 gives the maximum number of words that LAMDA scattering-data records may occupy. IW2 gives the maximum number of words that all scattering-data records in the material block may occupy.



PRINT1


START
Read material-identification record of material into (MATNAME,...,MØRERECS).
Print MATNAME, atomic density and description (if any element of MDESCRIP is non-blank), special character, and notes regarding TFLAG and FFLAG.
FFLAG<0 ?
no
yes
6 → I1
G+6 → I1
Read chi-record into (CHI(1),...,CHI(G)).
(STATEMENT Nº 164)
I1 → IW
1+ max [LMAX(1), LMAX(2), LMAX(3)] → LAMDA
0 → NF(I) → NG(I)
(I=1,...,3)
1 → I
LMAX(I)<0 ?
yes
no
IW+1 → NF(I)
NF(I)+LMAX(I)+1 → NG(I)
IW+3*LMAX(I)+3 → IW
I+1 → I
I>3 ?
no
yes
1 → IX(1)
IX(I-1)+IW → IX(I)
(I=2,...,G)
IW-1 → I2
IX(G)+IW → I3
LAMDA*(3*G+1) → IW1
IW1*G → IW2
MAX-I3+1 → LEFT
A
1.

PRINT1 (Contd.)

2. If IW2 is greater than LEFT, there may not be room for all the scattering-data records in core. They will be placed on tape unit 2.

3. I28 is the index of the first cell in the array RCDS that is available for storage of the scattering-data records. It will be incremented as the records are read and stored.

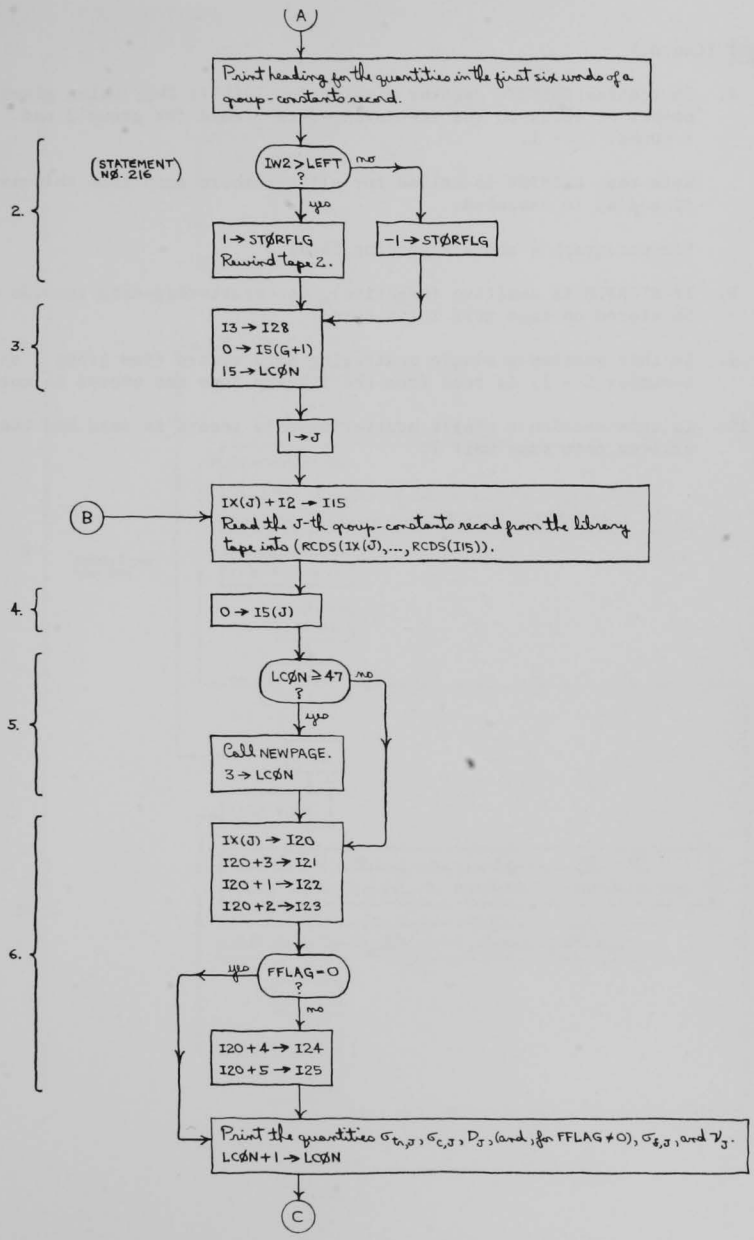
   LCON contains the number of lines that have been printed since the last page was ejected.

The coding of loop B is traversed G times. Each time the loop is traversed, the Jth group-constants record and the LAMDA scattering-data records that follow it are read from the library tape. The first several words in each group-constants records are printed after each record is read. The scattering-data records are either stored in core or written on tape unit 2.

4. I5(J) will contain the total number of words in all the scattering-data records that follow the Jth group-constants record (this value will vary from group to group).

5. If too many lines have been printed on a page, or if it is decided to separate sections of the printing for clarity, NEWPAGE is called. NEWPAGE prints a page-eject line and a line giving the current page number.

6. I20 through I25 locate various quantities (e.g., $\sigma_{tr,J}$, D_J, etc.) to be printed from a group-constants record, within that record.


A
Print heading for the quantities in the first six words of a group-constants record.
(STATEMENT NØ. 216)
IW2 > LEFT ?
no
yes
1 → STØRFLG
Rewind tape 2.
-1 → STØRFLG
I3 → I28
0 → I5(G+1)
I5 → LCØN
1 → J
B
IX(J) + I2 → I15
Read the J-th group-constants record from the library tape into (RCDS(IX(J),..., RCDS(I15)).
0 → I5(J)
LCØN ≥ 47 ?
no
yes
Call NEWPAGE.
3 → LCØN
IX(J) → I20
I20+3 → I21
I20 + 1 → I22
I20 + 2 → I23
FFLAG = 0 ?
yes
no
I20 + 4 → I24
I20 + 5 → I25
Print the quantities $\sigma_{n,J}$, $\sigma_{c,J}$, D_J, (and, for FFLAG ≠ 0), $\sigma_{f,J}$, and ν_J.
LCØN + 1 → LCØN
C
2.
3.
4.
5.
6.

PRINT1 (Contd.)

7. Subroutine GETSIZE returns a value for I4(L1); this value gives the number of words in the scattering-data record for group J and ℓ-number L1 - 1.

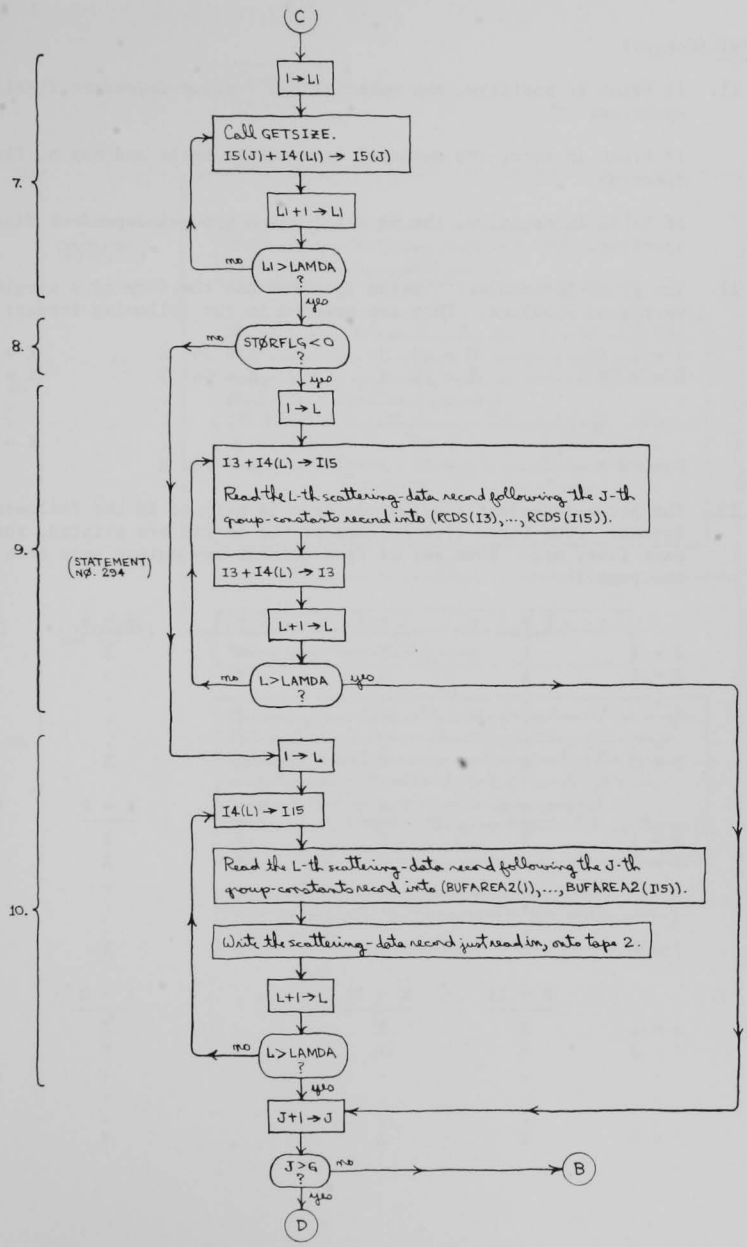
   Note that GETSIZE is called for all ℓ-numbers each time this section of coding is executed.

   See paragraph 4 above regarding I5(J).

8. If STORFLG is positive (negative), the scattering-data records will be stored on tape unit 2 (in core).

9. In this section a single scattering-data record (for group J and ℓ-number L - 1) is read from the library tape and stored in core.

10. In this section a single scattering-data record is read and then written onto tape unit 2.


C
1 → L1
Call GETSIZE.
I5(J) + I4(L1) → I5(J)
L1 + 1 → L1
L1 > LAMDA ?
no
yes
7.
STØRFLG < 0 ?
no
yes
8.
1 → L
I3 + I4(L) → I15
Read the L-th scattering-data record following the J-th group-constants record into (RCDS(I3),..., RCDS(I15)).
I3 + I4(L) → I3
L + 1 → L
L > LAMDA ?
no
yes
9.
(STATEMENT NØ. 294)
1 → L
I4(L) → I15
Read the L-th scattering-data record following the J-th group-constants record into (BUFAREA2(1),..., BUFAREA2(I15)).
Write the scattering-data record just read in, onto tape 2.
L + 1 → L
L > LAMDA ?
no
yes
10.
J + 1 → J
J > G ?
no
yes
B
D

PRINT1 (Contd.)

11. If FFLAG is positive, the material has a group-dependent fission spectrum.

    If FFLAG is zero, the material is nonfissionable and has no fission spectrum.

    If FFLAG is negative, the material has a group-independent fission spectrum.

12. The group-independent fission spectrum has the form of a single vector of G values. They are printed in the following format:

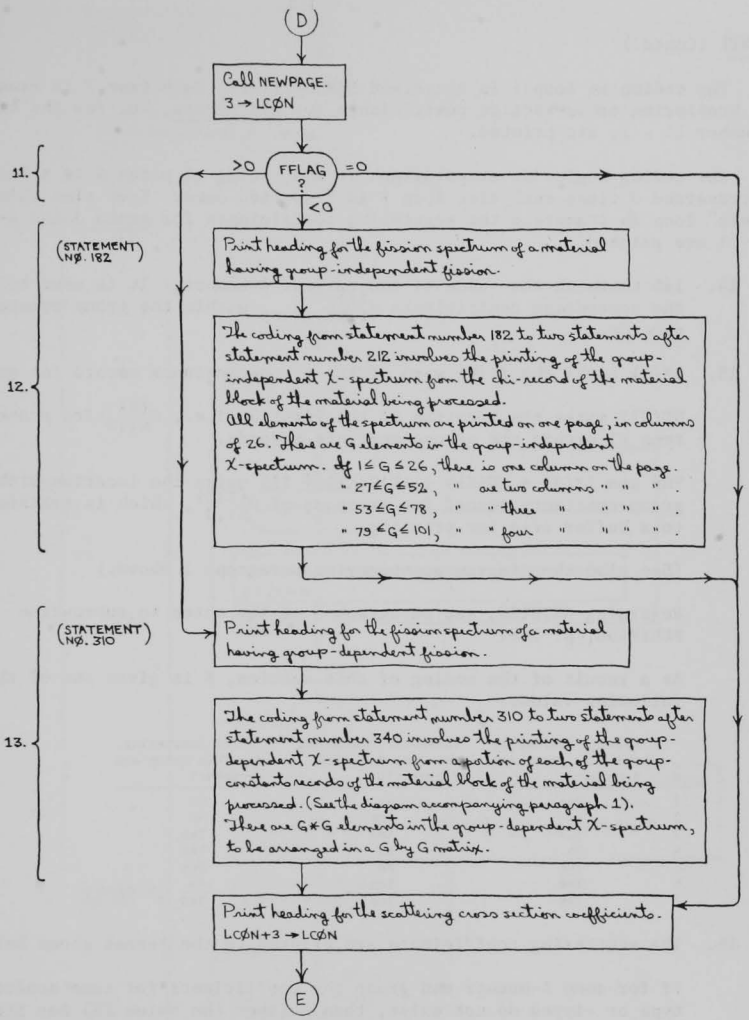
K = 1 X	K = 27 X	K = 35 X	K = 79 X
K = 2 X	K = 28 X	K = 54 X	K = 80 X
. .	. .	. .	. .
. .	. .	. .	. .
. .	. .	. .	K = G X
K = 26 X	K = 52 X	K = 78 X	

13. The group-dependent fission spectrum is printed in the following format: (The first five columns of the matrix are printed, then the next five, etc. Each set of five columns may extend over more than one page.)

	K = 1	K = 2	K = 3	K = 4	K = 5
J = 1	X	X	X	X	X
J = 2	X	X	X	X	X
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
J = G	X	X	X	X	X

	K = 6	K = 7	K = 8	K = 9	K = 10
J = 1	X	X	X	X	X
J = 2	X	X	X	X	X
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
J = G	X	X	X	X	X

	K = 11	K = 12	. . .	K = G
J = 1	X	X		X
J = 2	X	X		X
.	.	.		.
.	.	.		.
.	.	.		.
J = G	X	X		X


D
Call NEWPAGE.
3 → LCØN
11.
FFLAG ?
>0
=0
<0
(STATEMENT NØ. 182)
Print heading for the fission spectrum of a material having group-independent fission.
12.
The coding from statement number 182 to two statements after statement number 212 involves the printing of the group-independent χ-spectrum from the chi-record of the material block of the material being processed.
All elements of the spectrum are printed on one page, in columns of 26. There are G elements in the group-independent χ-spectrum. If 1 ≤ G ≤ 26, there is one column on the page.
" 27 ≤ G ≤ 52, " are two columns " " " .
" 53 ≤ G ≤ 78, " " three " " " " .
" 79 ≤ G ≤ 101, " " four " " " " .
(STATEMENT NØ. 310)
Print heading for the fission spectrum of a material having group-dependent fission.
13.
The coding from statement number 310 to two statements after statement number 340 involves the printing of the group-dependent χ-spectrum from a portion of each of the group-constants records of the material block of the material being processed. (See the diagram accompanying paragraph 1).
There are G*G elements in the group-dependent χ-spectrum, to be arranged in a G by G matrix.
Print heading for the scattering cross section coefficients.
LCØN+3 → LCØN
E

PRINT1 (Contd.)

The coding in loop F is traversed LAMDA times. Each time F is traversed, the scattering cross-section coefficients for all groups, but for the single ℓ-number L1 - 1, are printed.

The coding beginning at paragraph 14 and ending at point G is a loop that is traversed G times each time loop F is traversed once. Each time this "inner" loop is traversed the scattering coefficients for group J and ℓ-number L1 - 1 are printed.

14. I15 contains the value of the current ℓ-number. It is used to locate the scattering coefficients $\sigma^{(\ell)}_{type,group}$ within the group-constants records.

15. IX(J) marks the first word of the group-constants record for group J.

    NG(K1) marks the location of the first G (i.e., $G^{(0)}_{K1,J}$) for scattering-type K1 within the group-constants records.

    The sum IX(J) + NG(K1) + LMAX(K1) + I15 marks the location within the group-constants record for group J, of $\sigma^{(I15)}_{K1,J}$, which is transferred to a buffer area for printing.

    (See also the diagram accompanying paragraph 1 above.)

    Regarding GETSIZE, see paragraph 3 of the notes to subroutine PCHTP468, p. 148.

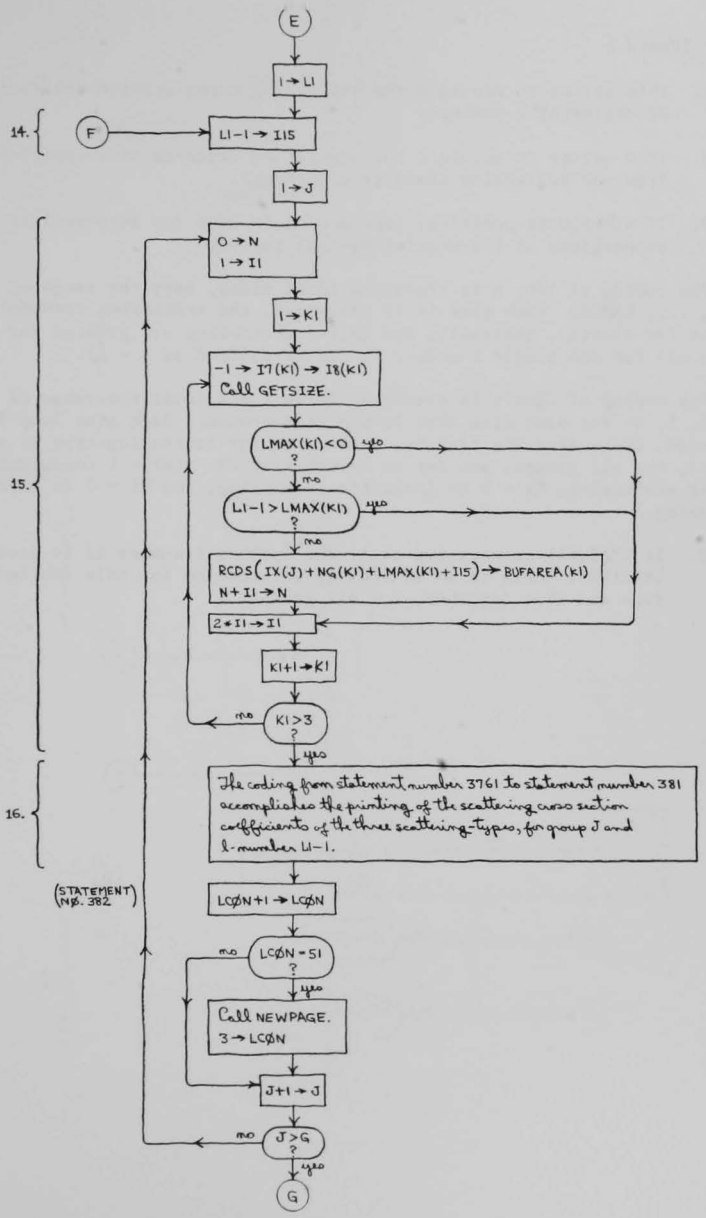
    As a result of the coding of this section, N is given one of the following values:

N	Elastic Scattering for This Group and ℓ-number?	Inelastic Scattering for This Group and ℓ-number?	(n,2n) Scattering for This Group and ℓ-number?
1	Yes	No	No
2	No	Yes	No
3	Yes	Yes	Yes
4	No	No	Yes
5	Yes	No	Yes
6	Yes	Yes	No
7	Yes	Yes	Yes

16. The scattering coefficients are printed in the format shown below.

    If for some ℓ-number and group the coefficients for some scattering-type or -types do not exist, then neither the value (X) nor its accompanying index (J = n) is printed for that type.

	Elastic		Inelastic		(n,2n)	
L = 0	J = 1	X	J = 1	X	J = 1	X
⋮	⋮	⋮	⋮	⋮	⋮	⋮
L = 0	J = G	X	J = G	X	J = G	X
L = 1	J = 1	X	J = 1	X	J = 1	X
⋮	⋮	⋮	⋮	⋮	⋮	⋮
L = 1	J = G	X	J = G	X	J = G	X
⋮	⋮		⋮		⋮	
L = LAMDA	J = 1	X	J = 1	X	J = 1	X
⋮	⋮	⋮	⋮	⋮	⋮	⋮
L = LAMDA	J = G	X	J = G	X		


E
1 → LI
14.
F
LI−1 → I15
1 → J
0 → N
1 → I1
1 → K1
−1 → I7(K1) → I8(K1)
Call GETSIZE.
LMAX(K1)<0 ?
yes
no
15.
LI−1 > LMAX(K1) ?
yes
no
RCDS(IX(J)+NG(K1)+LMAX(K1)+I15) → BUFAREA(K1)
N+I1 → N
2*I1 → I1
K1+1 → K1
no
K1>3 ?
yes
16.
The coding from statement number 3761 to statement number 381 accomplishes the printing of the scattering cross section coefficients of the three scattering-types, for group J and l-number LI−1.
(STATEMENT NØ. 382)
LCØN+1 → LCØN
no
LCØN = 51 ?
yes
Call NEWPAGE.
3 → LCØN
J+1 → J
no
J>G ?
yes
G

PRINT1 (Contd.)

17. This serves to separate the scattering cross-section coefficients of different ℓ-numbers.

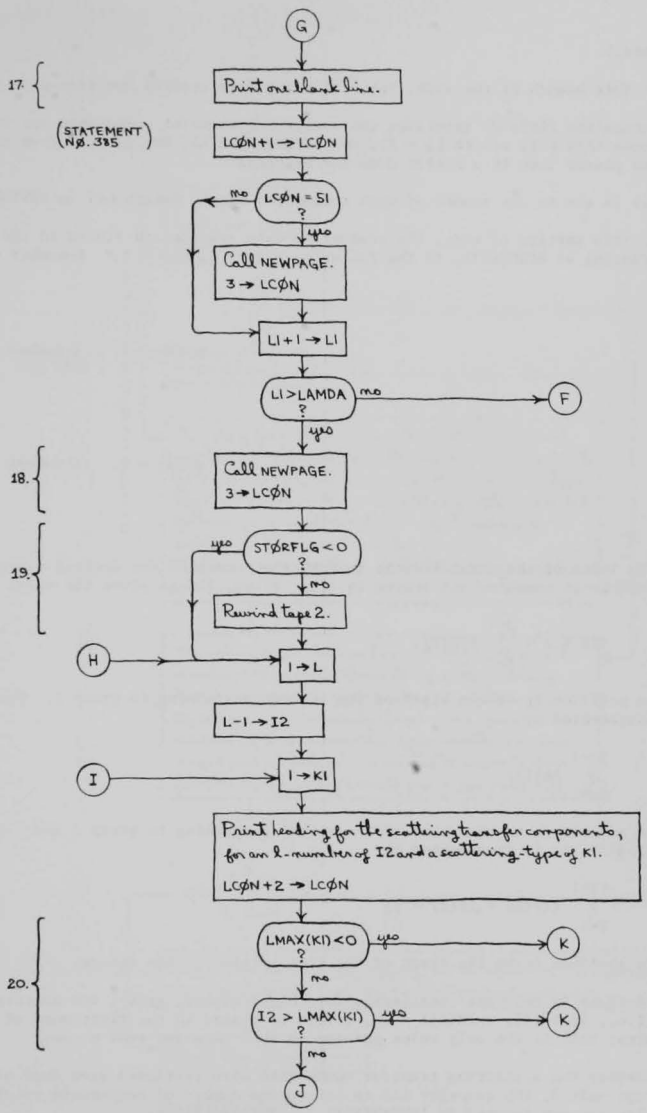
18. This serves to separate the scattering cross-section coefficients from the scattering transfer components.

19. If STORFLG is positive, tape 2 will be used for intermediate storage of portions of the scattering-data records.

The coding of loop H is traversed LAMDA times, over the range of values L = 1, ..., LAMDA. Each time it is traversed, the scattering transfer components for elastic, inelastic, and (n,2n) scattering are printed for all groups and for the single ℓ-number I2 (I2 is defined as L - 1).

The coding of loop I is traversed three times (over the range of values K1 = 1, 2, 3) for each time that loop H is executed. Each time loop I is traversed, the scattering transfer components for scattering-type K1 are printed, for all groups, and for an ℓ-number of I2. (K1 = 1 corresponds to elastic scattering, K1 = 2 to inelastic scattering, and K1 = 3 to (n,2n) scattering.)

20. If LMAX(K1) is negative or if the current ℓ-number I2 is greater than LMAX(K1), there is no scattering information for this scattering-type and this ℓ-number, for all groups.


G
17.
Print one blank line.
(STATEMENT NØ. 335)
LCØN+1 → LCØN
LCØN = 51 ?
no
yes
Call NEWPAGE.
3 → LCØN
L1+1 → L1
L1 > LAMDA ?
no
F
yes
18.
Call NEWPAGE.
3 → LCØN
19.
STØRFLG < 0 ?
yes
no
Rewind tape 2.
H
1 → L
L-1 → I2
I
1 → K1
Print heading for the scattering transfer components, for an l-number of I2 and a scattering type of K1.
LCØN+2 → LCØN
20.
LMAX(K1) < 0 ?
yes
K
no
I2 > LMAX(K1) ?
yes
K
no
J

PRINT1 (Contd.)

21. In this branch of the code, the scattering-data records are stored on tape unit 2.

    Subroutine PRNTSCRH retrieves the scattering transfer components for ℓ-number L (note that L is set to L1 - 1), scattering-type K1, and group J from tape unit 2 and places them in a buffer area for printing.

    I15 is set to the number of such components and is determined in PRNTSCRH.

22. In this section of code, the scattering-data records are stored in the array RCDS, starting at RCDS(I28), in the following order:

group = 1,	ℓ-number = 0	
1	1	
⋮	⋮	
1	LAMDA	
group = 2,	ℓ-number = 0	
2	1	
⋮	⋮	
2	LAMDA	
⋮	⋮	
group = G,	ℓ-number = 0	
G	1	
⋮	⋮	
G	LAMDA	

    The index of the first element in RCDS that contains the desired transfer components is computed and stored in I26. First, I26 is given the value

$$I28 + 1 + \sum_{I=1}^{J-1} I15(I),$$

    to position it to the start of the records pertaining to group J. Second, it is incremented by

$$\sum_{L1=1}^{I2-1} I4(L1),$$

    to advance it to the start of the record pertaining to group J and ℓ-number I2. Finally, it is incremented by

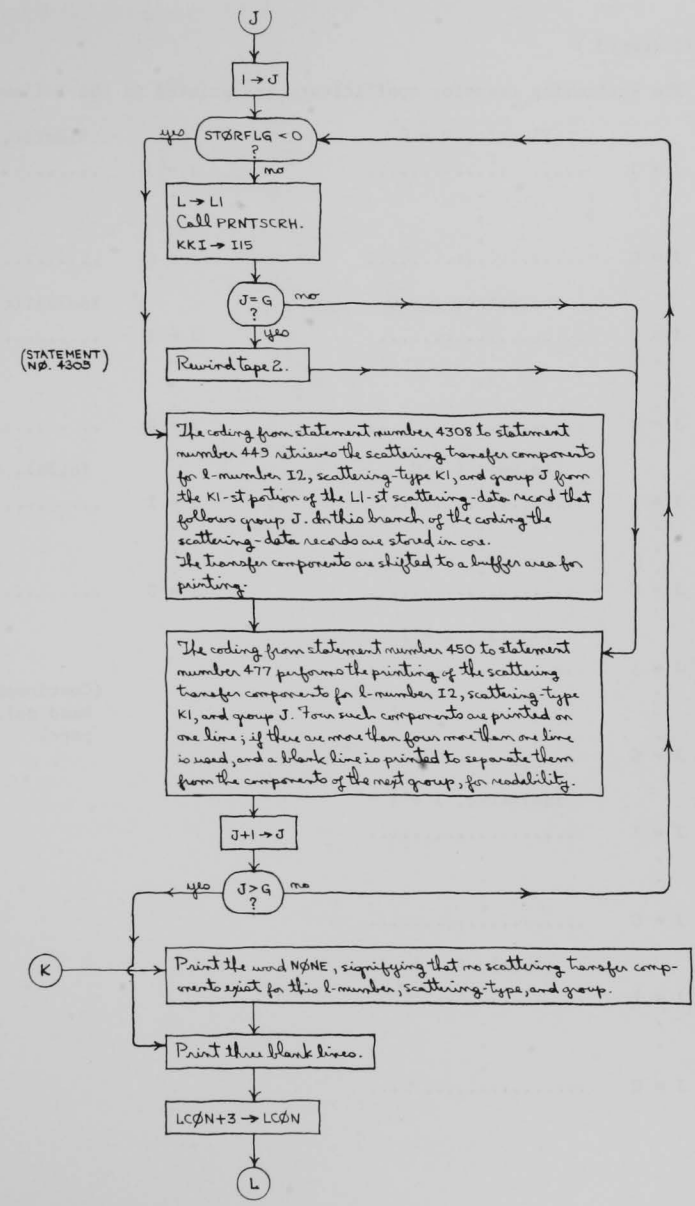
$$\sum_{I=1}^{K1-1} (I7(I) + I8(I) + 1)$$

    to position it to the start of the K1st portion of the record.

    If there is only self-scattering for this ℓ-number, group, and scattering-type (i.e., if I7(K1) = I8(K1) = 0), a zero is placed in the first word of the buffer area; this is the only value printed on that line for such a case.

23. Whether the scattering transfer components were retrieved from core or from tape unit 2, the quantity I15 is set to the number of components retrieved; they are now located in [BUFAREA(1),...,BUFAREA(I15)].

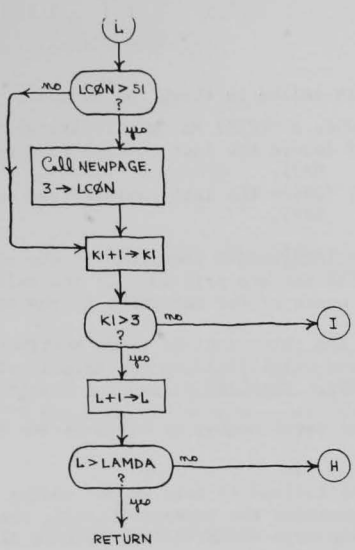
*24. This serves to separate blocks of transfer components.


J
1 → J
STØRFLG < 0 ?
yes
no
L → L1
Call PRNTSCRH.
KKI → I15
21.
J = G ?
no
yes
(STATEMENT NØ. 4308)
Rewind tape 2.
22.
The coding from statement number 4308 to statement number 449 retrieves the scattering transfer components for l-number I2, scattering-type K1, and group J from the K1-st portion of the L1-st scattering-data record that follows group J. In this branch of the coding the scattering-data records are stored in core.
The transfer components are shifted to a buffer area for printing.
23.
The coding from statement number 450 to statement number 477 performs the printing of the scattering transfer components for l-number I2, scattering-type K1, and group J. Four such components are printed on one line; if there are more than four more than one line is used, and a blank line is printed to separate them from the components of the next group, for readability.
J+1 → J
J > G ?
yes
no
K
Print the word NØNE, signifying that no scattering transfer components exist for this l-number, scattering-type, and group.
24.
Print three blank lines.
LCØN+3 → LCØN
L

PRINT1 (Contd.)

The scattering transfer coefficients are printed in the following format:

```
                Elastic, ℓ = 0                          Elastic, ℓ = LAMDA

J = 1     ....................                  J = 1     ....................
  .                .                            .                .
  .                .                            .                .
  .                .                            .                .
J = G     ....................                  J = G     ....................

               Inelastic, ℓ = 0                        Inelastic, ℓ = LAMDA

J = 1     ....................                  J = 1     ....................
  .                .                            .                .
  .                .                            .                .
  .                .                            .                .
J = G     ....................                  J = G     ....................

                (n,2n), ℓ = 0                           (n,2n), ℓ = LAMDA

J = 1     ....................                  J = 1     ....................
  .                .                            .                .
  .                .                            .                .
  .                .                            .                .
J = G     ....................                  J = G     ....................
                                                                 .
                Elastic, ℓ = 1                                   .
J = 1     ....................                                   .
  .                .                            (Continued in right-
  .                .                             hand column of this
  .                .                             page)
J = G     ....................

               Inelastic, ℓ = 1

J = 1     ....................
  .                .
  .                .
  .                .
J = G     ....................

                (n,2n), ℓ = 1

J = 1     ....................
  .                .
  .                .
  .                .
J = G     ....................
```


L
LCØN > 51
?
no
yes
Call NEWPAGE.
3 → LCØN
K1+1 → K1
K1 > 3
?
no
I
yes
L+1 → L
L > LAMDA
?
no
H
yes
RETURN

PRMATDIR

This subroutine is called in three instances:

(1) From SCAN2 (when a *PRINT MAT DIRECTORY card is encountered).
(2) From PRINTMAT (where the last control card encountered was *PRINT MAT).
(3) From PRINTSET (where the last control card encountered was *PRINT SET).

In each case, the information contained in the specifications and group-structure records of the set are printed. If the call is from SCAN2 or PRINTSET, the list of names of the materials in the set is also printed.

1. Whichever of the three control cards mentioned above was encountered, the second (and only) field on the card is either a set name or the message ALLSETS. SCANCARD places the quantity in FIELD(2).

   NUMSETS is the total number of files on the library tape being edited.

2. ALLSETS is initialized to zero in the coding of paragraph 1 above. If FIELD(2) contains the message ALLSETS, then the last control card was of the type *PRINT MAT DIRECTORY, ALLSETS. In this case, it is necessary to read in each cross-section set on the tape (files 2 through NUMSETS) in the order that they appear. This is done by loop A, which is traversed NUMSETS - 1 times, each time printing the data for one set.

   Tape LIB is the library tape (either new or old) that is being edited. It is rewound and spaced forward to the first of the cross-section sets (file 2).

3. If FIELD(2) does not contain the message ALLSETS, it contains a single set name; information concerning this set alone is to be printed. The set names in the directory of sets are compared to the set name from the control card; an equality must be found or the control card is in error. The set name is a five-character Hollerith quantity; therefore NSCAN(2) is set to 5.

   The tape is to be positioned so as to be at the start of file number TEMP1, the file corresponding to the set name on the control card.

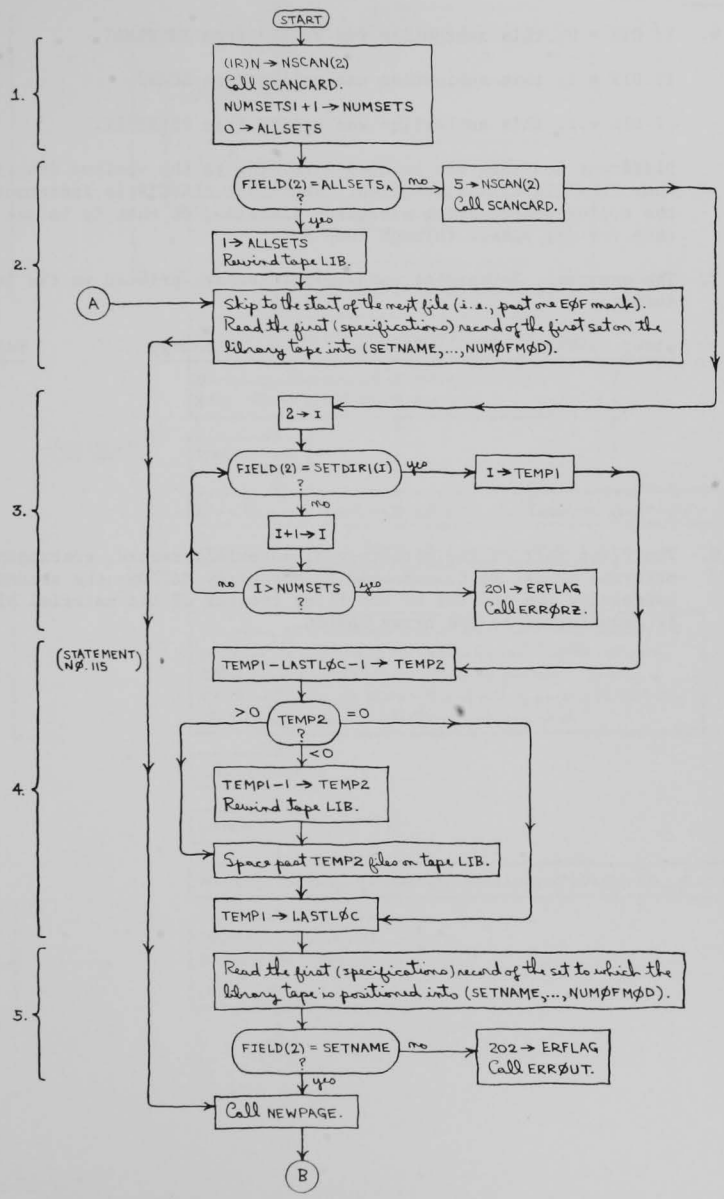
4. LASTLOC designates the present position of the tape; the tape is currently positioned at the start of file number LASTLOC + 1.

   TEMP2 represents the necessary change in tape placement necessary to position it to the start of file number TEMP1. If TEMP2 is zero, the tape is already correctly positioned; if TEMP2 is +N (-N), it must be spaced forward (backward) N files. To accomplish a backspace, the tape is rewound and then spaced forward TEMP1 - 1 files.

   The new position TEMP1 will be the quantity LASTLOC for the next repositioning of the tape.

5. The specifications record of the set just located is read. The set name in this record must conform to the set name given on the control card, or there has been a tape or hardware failure.

PRMATDIR


START
1.
(IR)N → NSCAN(2)
Call SCANCARD.
NUMSETS1 + 1 → NUMSETS
0 → ALLSETS
2.
FIELD(2) = ALLSETS ?
no
5 → NSCAN(2)
Call SCANCARD.
yes
1 → ALLSETS
Rewind tape LIB.
A
Skip to the start of the next file (i.e., past one EØF mark).
Read the first (specifications) record of the first set on the library tape into (SETNAME,...,NUMØFMØD).
3.
2 → I
FIELD(2) = SETDIR1(I) ?
yes
I → TEMP1
no
I+1 → I
no
I > NUMSETS ?
yes
201 → ERFLAG
Call ERRØRZ.
(STATEMENT NØ. 115)
4.
TEMP1 − LASTLØC − 1 → TEMP2
>0
TEMP2 ?
=0
<0
TEMP1 − 1 → TEMP2
Rewind tape LIB.
Space past TEMP2 files on tape LIB.
TEMP1 → LASTLØC
5.
Read the first (specifications) record of the set to which the library tape is positioned into (SETNAME,...,NUMØFMØD).
FIELD(2) = SETNAME ?
no
202 → ERFLAG
Call ERRØUT.
yes
Call NEWPAGE.
B

PRMATDIR (Contd.)

6. If DIR = 0, this subroutine was called from PRINTMAT.

   If DIR = 1, this subroutine was called from SCAN2.

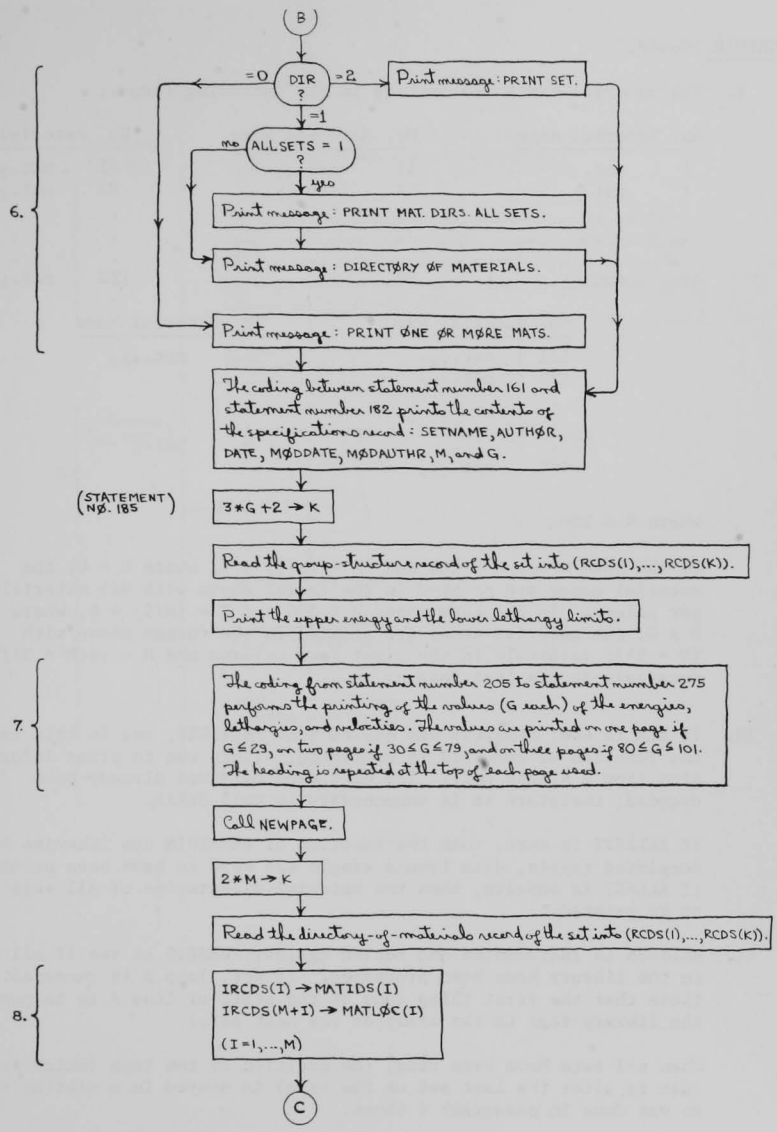
   If DIR = 2, this subroutine was called from PRINTSET.

   Different headings are printed according to the various functions that PRMATDIR will accomplish. Note that ALLSETS is incremented in the coding accompanying paragraph 11 below, so that it is one only once for all passes through loop A.

7. The energies, lethargies, and velocities are printed in the following format:

Group Number	Energy	Lethargy	Velocity
1	X	X	X
2	X	X	X
3	X	X	X
.	.	.	.
.	.	.	.
.	.	.	.
G	X	X	X

8. The first half of the directory-of-materials record, containing the material names, is transferred to the array MATIDS; the second half, containing the indices of the first records of the material blocks, is transferred to the array MATLOC.


B
DIR ?
=0
=2
=1
Print message: PRINT SET.
ALLSETS = 1 ?
no
yes
Print message: PRINT MAT. DIRS. ALL SETS.
Print message: DIRECTØRY ØF MATERIALS.
Print message: PRINT ØNE ØR MØRE MATS.
6.
The coding between statement number 161 and statement number 182 prints the contents of the specifications record: SETNAME, AUTHØR, DATE, MØDDATE, MØDAUTHR, M, and G.
(STATEMENT NØ. 185)
3*G+2 → K
Read the group-structure record of the set into (RCDS(1),...,RCDS(K)).
Print the upper energy and the lower lethargy limits.
7.
The coding from statement number 205 to statement number 275 performs the printing of the values (G each) of the energies, lethargies, and velocities. The values are printed on one page if G ≤ 29, on two pages if 30 ≤ G ≤ 79, and on three pages if 80 ≤ G ≤ 101. The heading is repeated at the top of each page used.
Call NEWPAGE.
2*M → K
Read the directory-of-materials record of the set into (RCDS(1),...,RCDS(K)).
8.
IRCDS(I) → MATIDS(I)
IRCDS(M+I) → MATLØC(I)
(I=1,...,M)
C

PRMATDIR (Contd.)

9. The material names are printed in the following format:

No.	Material Name	No.	Material Name	No.	Material Name
1	$mat._1$	41	$mat._{41}$	81	$mat._{81}$
2	$mat._2$	42	$mat._{42}$	82	$mat._{82}$
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
40	$mat._{40}$	80	$mat._{80}$	120	$mat._{120}$

No.	Material Name	No.	Material Name
121	$mat._{121}$	161	$mat._{161}$
122	$mat._{122}$	.	
.	.	.	
.	.	.	
.	.	N	$mat._N$
160	$mat._{160}$		

where $N \leq 200$.

In the case where $N > 200$ and $X = (M/5) + R$, where $R = 0$, the material names are printed in the format above with M/5 materials per column. In the case where $N > 200$ and $X = (M/5) + R$, where $R \neq 0$, the material names are printed in the format above with $(M + 5)/5$ materials in the first four columns and $M - [4(M + 5)/5]$ materials in the last column.

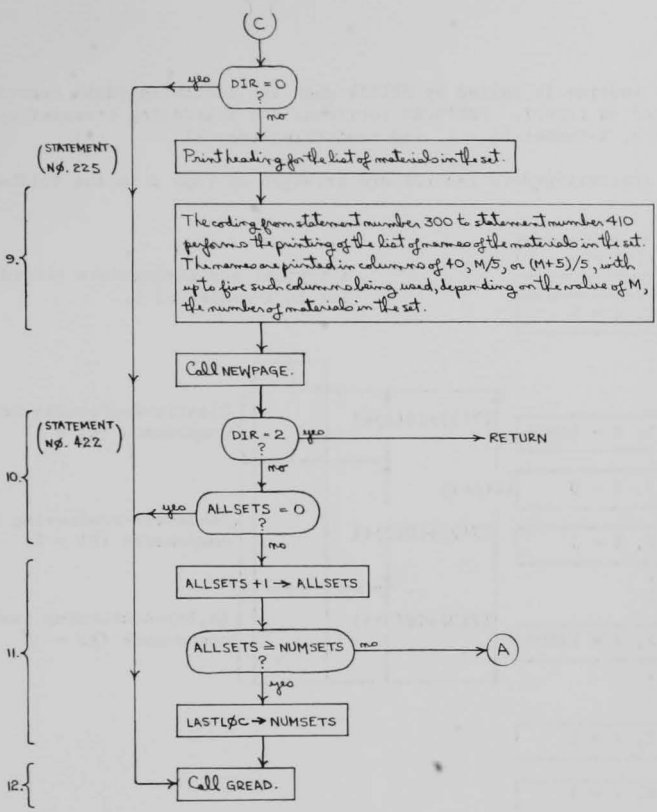
10. If DIR is two, PRMATDIR was called from PRINTSET, and in this case the function of PRMATDIR is completed. (This was to print information from a single set.) The next data card has already been decoded; therefore it is unnecessary to call GREAD.

    If ALLSETS is zero, then the function of PRMATDIR has likewise been completed (again, data from a single set were to have been printed). If ALLSET is nonzero, then the material directories of all sets are to be printed.

11. ALLSETS is incremented and tested against NUMSETS to see if all sets in the library have been processed. If not, loop A is repeated. (Note that the first thing done at the start of loop A is to position the library tape to the start of the next set.)

    When all sets have been read, the position of the tape (which in this case is after the last set on the tape) is stored in a similar manner as was done in paragraph 4 above.

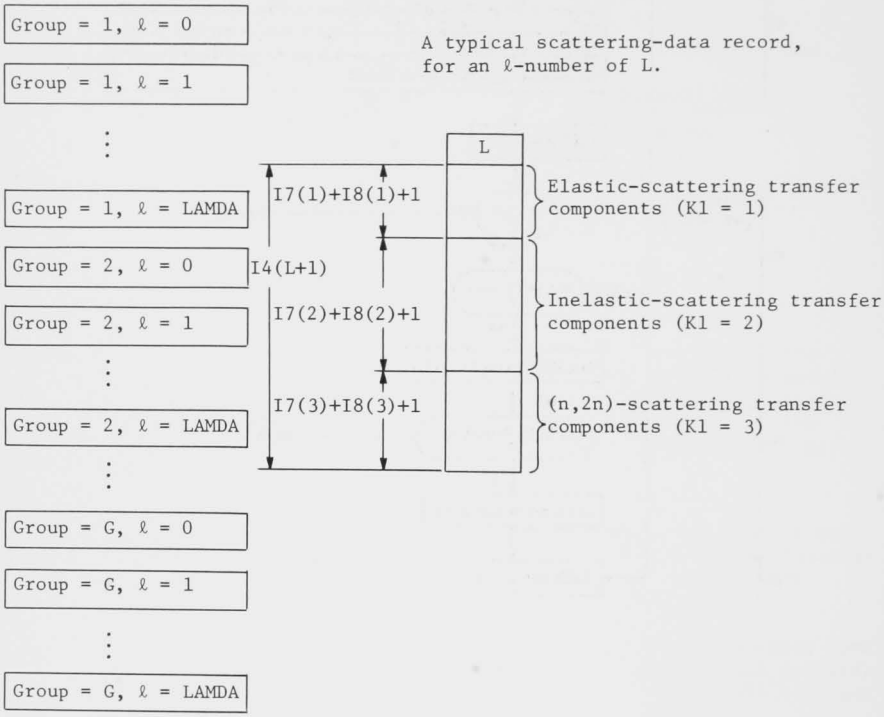
12. When the call to PRMATDIR was from SCAN2 or PRINTSET, it is necessary to decode the next data card.


C
DIR = 0 ?
yes
no
STATEMENT No. 225
Print heading for the list of materials in the set.
9.
The coding from statement number 300 to statement number 410 performs the printing of the list of names of the materials in the set. The names are printed in columns of 40, M/5, or (M+5)/5, with up to five such columns being used, depending on the value of M, the number of materials in the set.
Call NEWPAGE.
STATEMENT No. 422
10.
DIR = 2 ?
yes
RETURN
no
ALLSETS = 0 ?
yes
no
11.
ALLSETS + 1 → ALLSETS
ALLSETS ≥ NUMSETS ?
no
A
yes
LASTLOC → NUMSETS
12.
Call GREAD.

PRNTSCRH

This routine is called by PRINT1 when the scattering-data records have been stored on tape 2. PRNTSCRH retrieves the scattering transfer components for group J, ℓ-number L1 - 1, and scattering-type K1.

The scattering-data records are arranged on tape 2 in the following manner:



PRNTSCRH (Contd.)

1. M2 and M3 are used in the coding of paragraph 5 below. Regarding K1, see the notes to paragraph 21 of PRINT1 p. 184, where PRNTSCRH is called.

2. The scattering components are retrieved under control of subroutine PRINT1 in the following order:

	Group	ℓ-number	Scattering-type
Tape 2 rewound (in PRINT1)	1	0	1
	⋮	⋮	⋮
	G	⋮	1
Tape 2 rewound	1	⋮	2
	⋮	⋮	⋮
	G	⋮	2
Tape 2 rewound	1	⋮	3
	⋮	⋮	⋮
	G	0	3
Tape 2 rewound	1	1	1
	⋮	⋮	⋮
	G	etc.	1

Therefore it is necessary to space past a number of records in order to position tape unit 2 to the start of the record for group J and ℓ-number L1 - 1. (J = 1 is a special case since the tape will have just been rewound.) MORERECS contains the number of records to be spaced by on tape 2.

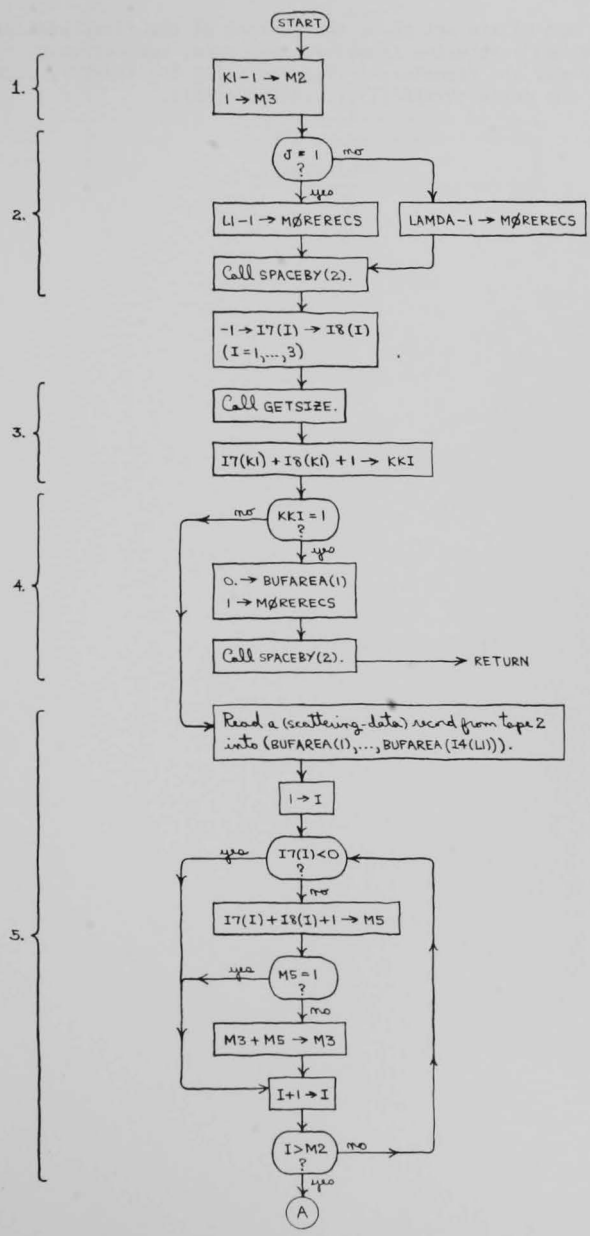
3. GETSIZE returns the length of the desired record, I4(L1), and the lengths of the three portions of this record, I7(I) and I8(I), I = 1, 2, 3. (See the diagram on the preceding page.)

4. If KKI is one, then I7(K1) = I8(K1) = 0; i.e., there are no transfer components of type K1 in this record. Tape unit 2 is spaced to the end of this record and a return is made.

5. If KKI is not one, there are transfer components of type K1 in the record currently being considered. The record is read into core.

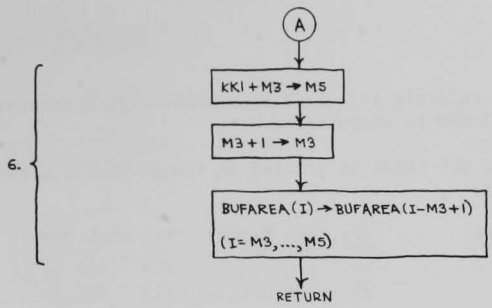
   At the end of this section, M3 is set so as to be an index to the word just before the first transfer component of type K1 within the scattering-data record just read.

PRNTSCRH


START
1.
KI-1 → M2
1 → M3
2.
J = 1 ?
no
yes
LI-1 → MØRERECS
LAMDA-1 → MØRERECS
Call SPACEBY(2).
-1 → I7(I) → I8(I)
(I = 1,...,3)
3.
Call GETSIZE.
I7(KI) + I8(KI) + 1 → KKI
4.
KKI = 1 ?
no
yes
0. → BUFAREA(1)
1 → MØRERECS
Call SPACEBY(2).
RETURN
5.
Read a (scattering-data) record from tape 2
into (BUFAREA(1),..., BUFAREA(I4(LI))).
1 → I
I7(I)<0 ?
yes
no
I7(I) + I8(I) + 1 → M5
M5 = 1 ?
yes
no
M3 + M5 → M3
I+1 → I
I>M2 ?
no
yes
A

PRNTSCRH (Contd.)

6. M3 and M5 are set to be the indices of the first and last of the desired scattering transfer components, respectively. These components are transferred from the cells BUFAREA(M3),...,BUFAREA(M5) to the cells BUFAREA(1),...,BUFAREA(KKI).


A
6.
KKI + M3 → M5
M3 + 1 → M3
BUFAREA(I) → BUFAREA(I-M3+1)
(I= M3,...,M5)
RETURN

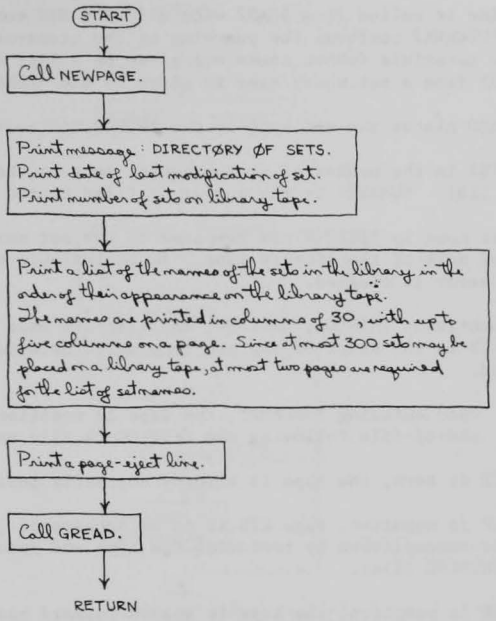
PRSETDIR

Subroutine PRSETDIR is called from SCAN2 when a control card of the type *PRINT SET DIRECTORY is encountered.

The list of set names is printed in the following format, using a maximum of seven columns:

No.	Set Name	No.	Set Name		No.	Set Name	No.	Set Name
1	set_1	44	set_{44}	...	216	set_{216}	259	set_{259}
2	set_2	45	set_{45}		217	set_{217}	260	set_{260}
⋮	⋮	⋮	⋮		⋮	⋮	⋮	⋮
42	set_{42}	85	set_{85}		257	set_{257}	300	set_{300}
43	set_{43}	86	set_{86}		258	set_{258}		

PRSETDIR


START
Call NEWPAGE.
Print message: DIRECTØRY ØF SETS.
Print date of last modification of set.
Print number of sets on library tape.
Print a list of the names of the sets in the library in the order of their appearance on the library tape.
The names are printed in columns of 30, with up to five columns on a page. Since at most 300 sets may be placed on a library tape, at most two pages are required for the list of set names.
Print a page-eject line.
Call GREAD.
RETURN

PUNCHMAT

This routine is called from SCAN2 when a *PUNCH MAT control card is encountered. PUNCHMAT performs the punching of the cross-section data cards for one or more materials (whose names are given on a list card following the *PUNCH MAT card) from a set whose name is given on the *PUNCH MAT card.

1. SCANCARD places the set name on the *PUNCH MAT card in FIELD(2).

   NUMSETS1 is the number of cross-section sets on the library tape (tape LIB). NUMSETS is the number of files on the library tape.

2. The set name in FIELD(2) is compared to the set names in the directory of sets of the library tape. An equivalence must be found, or an error is flagged.

   The quantities THISLOC, LASTLOC, and ITEP are used to position tape LIB to the start of the set from which materials are to be punched.

   Note: Upon entering PUNCHMAT, the tape is positioned after the end-of-file following the LASTLOC-th file on the tape.

3. If ITEP is zero, the tape is already correctly positioned.

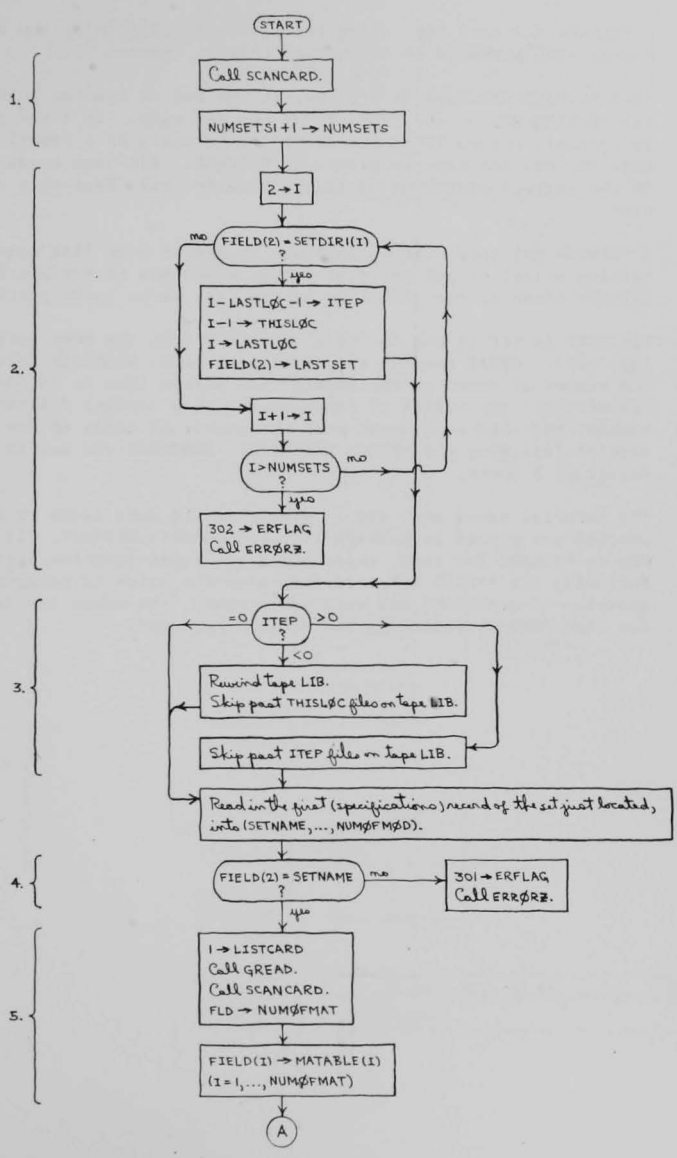
   If ITEP is negative, tape LIB is to be backspaced |ITEP| + 1 files. This is accomplished by rewinding the tape and spacing forward past THISLOC files.

   If ITEP is positive, the tape is spaced forward past ITEP files.

4. A check is made to see if the set just located is the set specified on the *PUNCH MAT card.

5. LISTCARD is set to one to "tell" subroutine SCANCARD that one or more list-cards (containing the names of the materials to be punched) is expected. GREAD decodes the first list card, and SCANCARD places the FLD material names on the card(s) in FIELD(1),...,FIELD(FLD). These names are transferred to the first FLD elements of the array MATABLE.

PUNCHMAT


START
1.
Call SCANCARD.
NUMSETS1+1 → NUMSETS
2.
2 → I
FIELD(2) = SETDIR1(I) ?
no
yes
I − LASTLØC − 1 → ITEP
I − 1 → THISLØC
I → LASTLØC
FIELD(2) → LASTSET
I + 1 → I
I > NUMSETS ?
no
yes
302 → ERFLAG
Call ERRØRZ.
3.
ITEP ?
=0
>0
<0
Rewind tape LIB.
Skip past THISLØC files on tape LIB.
Skip past ITEP files on tape LIB.
Read in the first (specifications) record of the set just located, into (SETNAME, ..., NUMØFMØD).
4.
FIELD(2) = SETNAME ?
no
301 → ERFLAG
Call ERRØRZ.
yes
5.
1 → LISTCARD
Call GREAD.
Call SCANCARD.
FLD → NUMØFMAT
FIELD(I) → MATABLE(I)
(I = 1, ..., NUMØFMAT)
A

PUNCHMAT (Contd.)

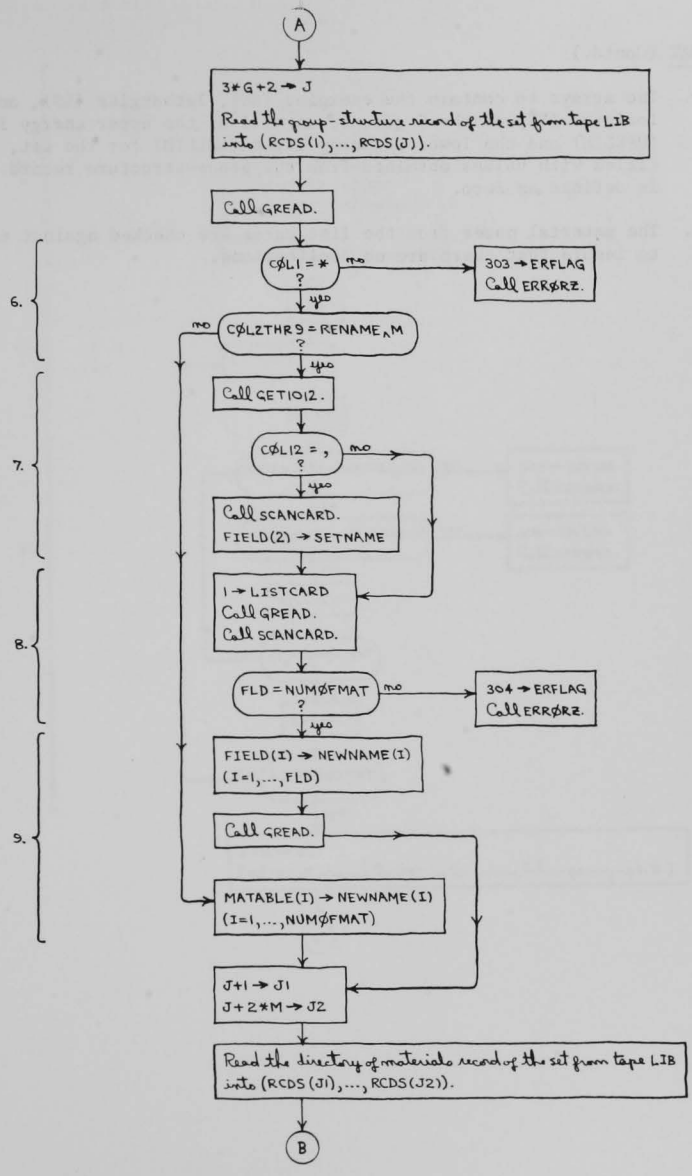
6. A *RENAME MAT card may follow the list card containing the material names. The presence or absence of this is checked for.

7. If a *RENAME MAT card is present, it has one of the two forms (a) *RENAME MAT or (b) *RENAME MAT,new set name. If a new set name is present, column 12 of the control card contains a comma; in this case the new set name is placed in SETNAME. All data cards punched in the current execution of this subroutine will bear this new set name.

8. A *RENAME MAT card must be followed by one or more list cards containing a list of new material names, which are to replace the present names of the materials on the data cards to be punched.

   LISTCARD is set to one to "tell" SCANCARD that the next card is a list card. GREAD decodes the first list card; SCANCARD determines the number of names on the card(s) and places them in FIELD(1),..., FIELD(FLD). The number of names on the list card(s) following the *RENAME MAT card must agree with the number of names on the list card(s) following the *PUNCH MAT card. NUMOFMAT was set in paragraph 5 above.

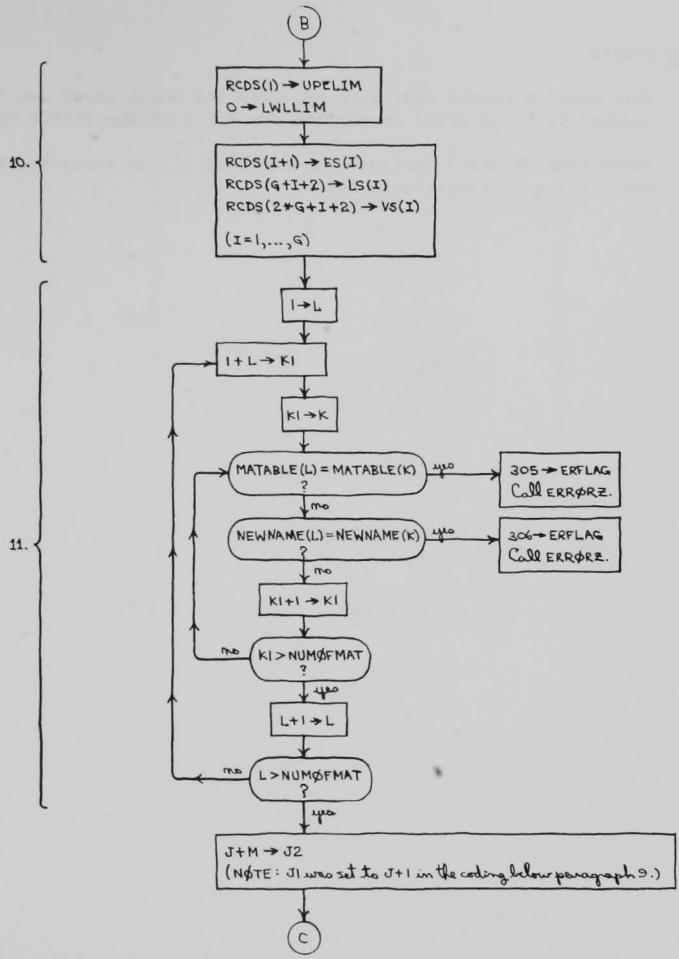
9. The material names that are to appear on the data cards to be punched are placed in NEWNAME(1),...,NEWNAME(NUMOFMAT). If there was no *RENAME MAT card, these names are taken from the list card(s) following the *PUNCH MAT card (see also the notes to paragraph 5 above). If a *RENAME MAT card was present, the names are taken from the list card(s) following the *RENAME MAT card.


A
3*G+2 → J
Read the group-structure record of the set from tape LIB into (RCDS(1),..., RCDS(J)).
Call GREAD.
6.
CØL1 = * ?
no
303 → ERFLAG
Call ERRØRZ.
yes
no
CØL2THR9 = RENAME∧M ?
yes
7.
Call GET1012.
CØL12 = , ?
no
yes
Call SCANCARD.
FIELD(2) → SETNAME
8.
1 → LISTCARD
Call GREAD.
Call SCANCARD.
FLD = NUMØFMAT ?
no
304 → ERFLAG
Call ERRØRZ.
yes
9.
FIELD(I) → NEWNAME(I)
(I=1,..., FLD)
Call GREAD.
MATABLE(I) → NEWNAME(I)
(I=1,..., NUMØFMAT)
J+1 → J1
J+2*M → J2
Read the directory of materials record of the set from tape LIB into (RCDS(J1),..., RCDS(J2)).
B

PUNCHMAT (Contd.)

10. The arrays to contain the energies (ES), lethargies (LS), and velocities (VS) of the G groups, as well as the upper energy limit (UPELIM) and the lower lethargy limit (LWLLIM) for the set, are filled with values obtained from the group-structure record. LWLLIM is defined as zero.

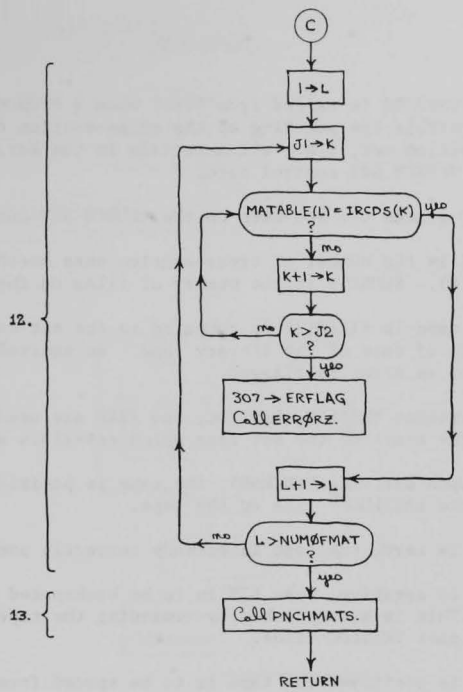
11. The material names from the list cards are checked against each other to insure that there are no duplications.


B
RCDS(1) → UPELIM
0 → LWLLIM
10.
RCDS(I+1) → ES(I)
RCDS(G+I+2) → LS(I)
RCDS(2*G+I+2) → VS(I)
(I=1,...,G)
1 → L
11.
1+L → K1
K1 → K
MATABLE(L) = MATABLE(K) ?
yes
305 → ERFLAG
Call ERRØRZ.
no
NEWNAME(L) = NEWNAME(K) ?
yes
306 → ERFLAG
Call ERRØRZ.
no
K1+1 → K1
K1 > NUMØFMAT ?
no
yes
L+1 → L
L > NUMØFMAT ?
no
yes
J+M → J2
(NØTE: J1 was set to J+1 in the coding below paragraph 9.)
C

PUNCHMAT (Contd.)

12. This section checks that each material for which cards are to be punched is to be found in the set specified on the *PUNCH MAT card.

13. Subroutine PNCHMATS controls the punching of the cross-section data cards for one or more materials.


C
1 → L
J1 → K
MATABLE(L) = IRCDS(K) ?
yes
no
K+1 → K
no
K > J2 ?
yes
307 → ERFLAG
Call ERRØRZ.
L+1 → L
no
L > NUMØFMAT ?
yes
Call PNCHMATS.
RETURN
12.
13.

PUNCHSET

Subroutine PUNCHSET is called from SCAN2 when a *PUNCH SET card is encountered. It controls the punching of the cross-section data cards for a complete cross-section set, i.e., all materials in the set. The set is specified on the *PUNCH SET control card.

1. SCANCARD places the set name on the *PUNCH SET card in FIELD(2).

   NUMSETS1 is the number of cross-section sets on the library tape (tape LIB). NUMSETS is the number of files on the library tape.

2. The set name in FIELD(2) is compared to the set names in the directory of sets of the library tape. An equivalence must be found, or an error is flagged.

   The quantities THISLOC, LASTLOC, and ITEP are used to position tape LIB to the start of the set from which materials are to be punched.

   Note: Upon entering PUNCHSET, the tape is positioned at the end of the LASTLOCth file on the tape.

3. If ITEP is zero, the tape is already correctly positioned.

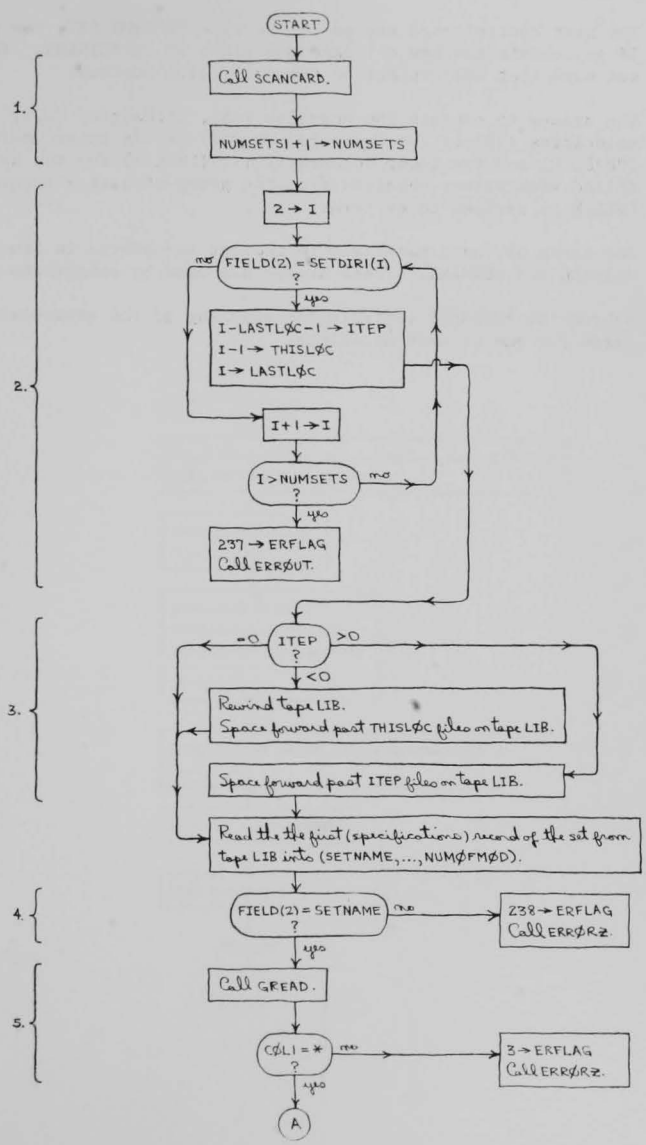
   If ITEP is negative, tape LIB is to be backspaced past LASTLOC files. This is accomplished by rewinding the tape and spacing forward past THISLOC files.

   If ITEP is positive, the tape is to be spaced forward past ITEP files.

4. A check is made to see if the set just located is the set specified on the *PUNCH SET card.

5. Decode the next control card.

PUNCHSET


START
Call SCANCARD.
NUMSETS1 + 1 → NUMSETS
2 → I
FIELD(2) = SETDIR1(I) ?
no
yes
I-LASTLØC-1 → ITEP
I-1 → THISLØC
I → LASTLØC
I+1 → I
I > NUMSETS ?
no
yes
237 → ERFLAG
Call ERRØUT.
ITEP ?
=0
>0
<0
Rewind tape LIB.
Space forward past THISLØC files on tape LIB.
Space forward past ITEP files on tape LIB.
Read the the first (specifications) record of the set from tape LIB into (SETNAME,...,NUMØFMØD).
FIELD(2) = SETNAME ?
no
yes
238 → ERFLAG
Call ERRØR2.
Call GREAD.
CØL1 = * ?
no
yes
3 → ERFLAG
Call ERRØR2.
A
1.
2.
3.
4.
5.

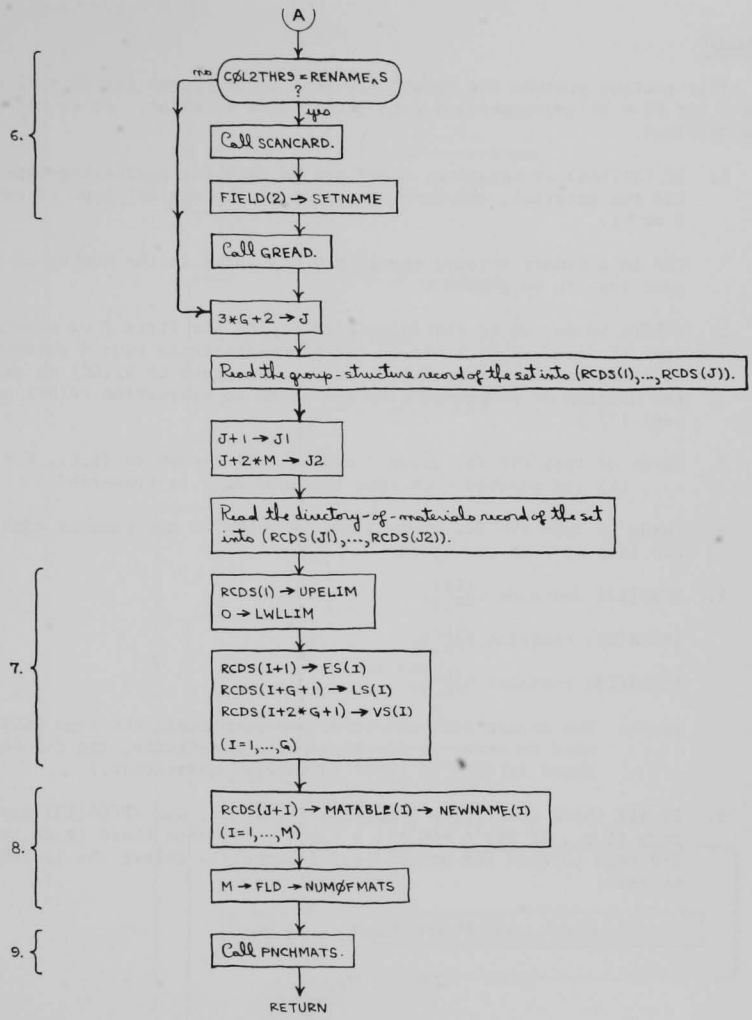
PUNCHSET (Contd.)

6. The next control card may be of the type *RENAME SET, new set name. If so, obtain the new set name and place it in SETNAME. This is the set name that will appear on the data cards punched.

7. The arrays to contain the energies (ES), lethargies (LS), and velocities (VS) of the G groups, as well as the upper energy limit (UPPELIM) and the lower lethargy limit (LWLLIM) for the set, are filled with values obtained from the group-structure record. LWLLIM is defined to be zero.

8. The names of the M materials in the set are placed in the arrays MATABLE and NEWNAME. These arrays are used by subroutine PNCH1MAT.

9. Subroutine PNCHMATS controls the punching of the cross-section data cards for one or more materials.


A
CØL2THRS = RENAME∧S ?
no
yes
6.
Call SCANCARD.
FIELD(2) → SETNAME
Call GREAD.
3*G+2 → J
Read the group-structure record of the set into (RCDS(1),...,RCDS(J)).
J+1 → J1
J+2*M → J2
Read the directory-of-materials record of the set into (RCDS(J1),...,RCDS(J2)).
7.
RCDS(1) → UPELIM
0 → LWLLIM
RCDS(I+1) → ES(I)
RCDS(I+G+1) → LS(I)
RCDS(I+2*G+1) → VS(I)
(I=1,...,G)
8.
RCDS(J+I) → MATABLE(I) → NEWNAME(I)
(I=1,...,M)
M → FLD → NUMØFMATS
9.
Call PNCHMATS.
RETURN

PUNCH357

This routine punches the type-3 (if KK = 1) or type-5 (if KK = 2) or type-7 (if KK = 3) cross-section data cards for a material. It is called from PNCH1MAT.

1. If LMAX(KK) is negative, there are no data for scattering-type KK for the material, and hence no cards of the current type (3 or 5 or 7).

   TYP is a binary integer quantity whose value is the number of the card type to be punched.

2. ITEMP1 is set to be the index pointing to the first F of scattering-type KK (i.e., $F^{(0)}_{KK,J}$) within the group-constants record currently in core, in the arrays RCDS. (It is analogous to NF(KK) as used in the diagram of paragraph 1 in the notes to subroutine PRINT1 on page 172.)

3. Cards of type TYP for group J and for all ℓ-numbers (i.e., ℓ = 1, ..., L1) are punched each time the loop on J is traversed.

4. Cards of type TYP for group J and ℓ-number L2 are punched each time the loop on L is traversed.

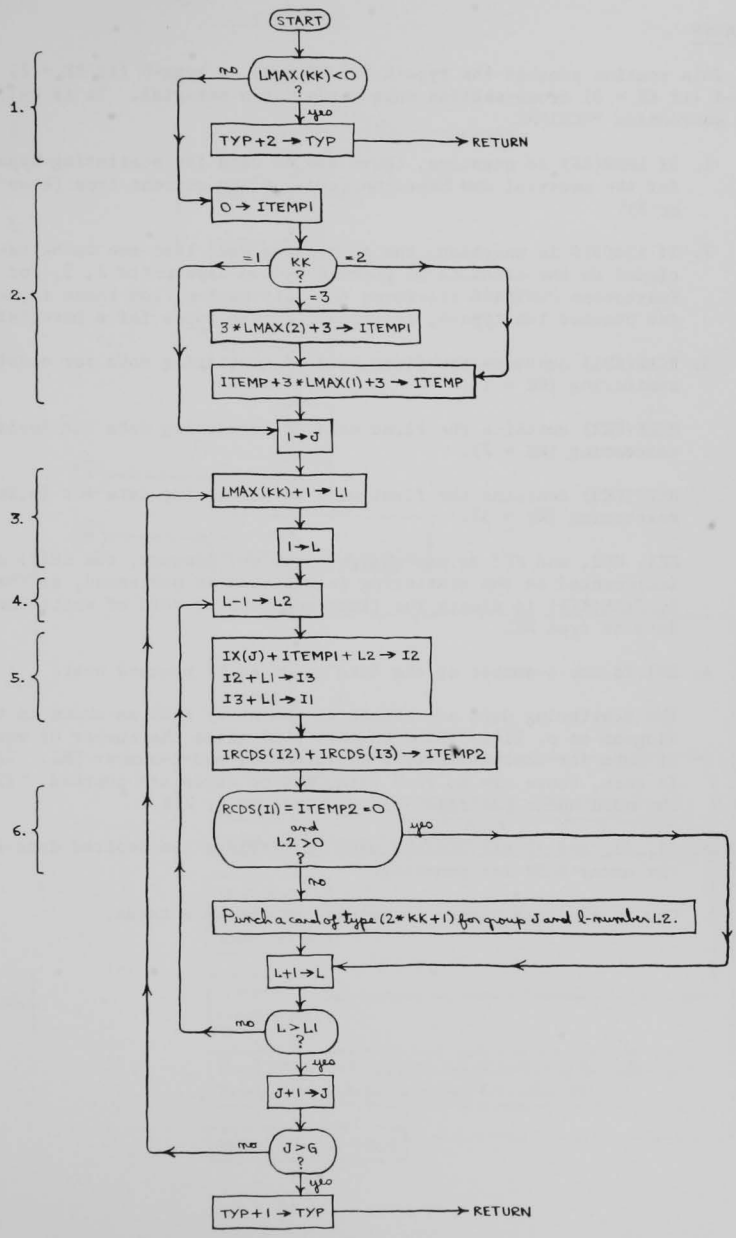
5. RCDS(I1) contains $\sigma^{(L2)}_{KK,J}$.

   IRCDS(I2) contains $F^{(L2)}_{KK,J}$.

   IRCDS(I3) contains $G^{(L2)}_{KK,J}$.

   (Note: The arrays RCDS and IRCDS are identical; the name RCDS is used to refer to floating-point quantities, and the name IRCDS is used to refer to integer quantities.)

6. If all three quantities RCDS(I1), IRCDS(I2), and IRCDS(I3) are zero (i.e., if $F^{(L2)}_{KK,J} = G^{(L2)}_{KK,J} = \sigma^{(L2)}_{KK,J} = 0$), then there is no type-TYP card punched for group J and ℓ-number L2 unless the ℓ-number L2 is zero.


START
1.
LMAX(KK)<0 ?
no
yes
TYP+2 → TYP
RETURN
2.
0 → ITEMP1
KK ?
=1
=2
=3
3*LMAX(2)+3 → ITEMP1
ITEMP+3*LMAX(1)+3 → ITEMP
1 → J
3.
LMAX(KK)+1 → L1
1 → L
4.
L-1 → L2
5.
IX(J)+ITEMP1+L2 → I2
I2+L1 → I3
I3+L1 → I1
IRCDS(I2)+IRCDS(I3) → ITEMP2
6.
RCDS(I1) = ITEMP2 = 0 and L2 > 0 ?
yes
no
Punch a card of type (2*KK+1) for group J and l-number L2.
L+1 → L
L > L1 ?
no
yes
J+1 → J
J > G ?
no
yes
TYP+1 → TYP
RETURN

PUNCH468

This routine punches the type-4 (if KK = 1) or type-6 (if KK = 2) or type-8 (if KK = 3) cross-section data cards for a material. It is called from subroutine PNCH1MAT.

1. If LMAX(KK) is negative, there are no data for scattering-type KK for the material and hence no cards of the current type (4 or 6 or 8).

2. If STORFLG is negative, the scattering data that are to be included on the cards to be punched are on tape units 2, 3, and 4. Subroutine PCHTP468 retrieves the information from these tapes and punches the type-4, type-6, or type-8 cards for a material.

3. RCDS(KK1) contains the first word of scattering data for elastic scattering (KK = 1).

   RCDS(KK2) contains the first word of scattering data for inelastic scattering (KK = 2).

   RCDS(KK3) contains the first word of scattering data for (n,2n) scattering (KK = 3).

   KK1, KK2, and KK3 do not change in value; however, the KM(I) are incremented as the scattering information is processed, so that RCDS(KM(KK)) is always the first unprocessed word of scattering data of type KK.

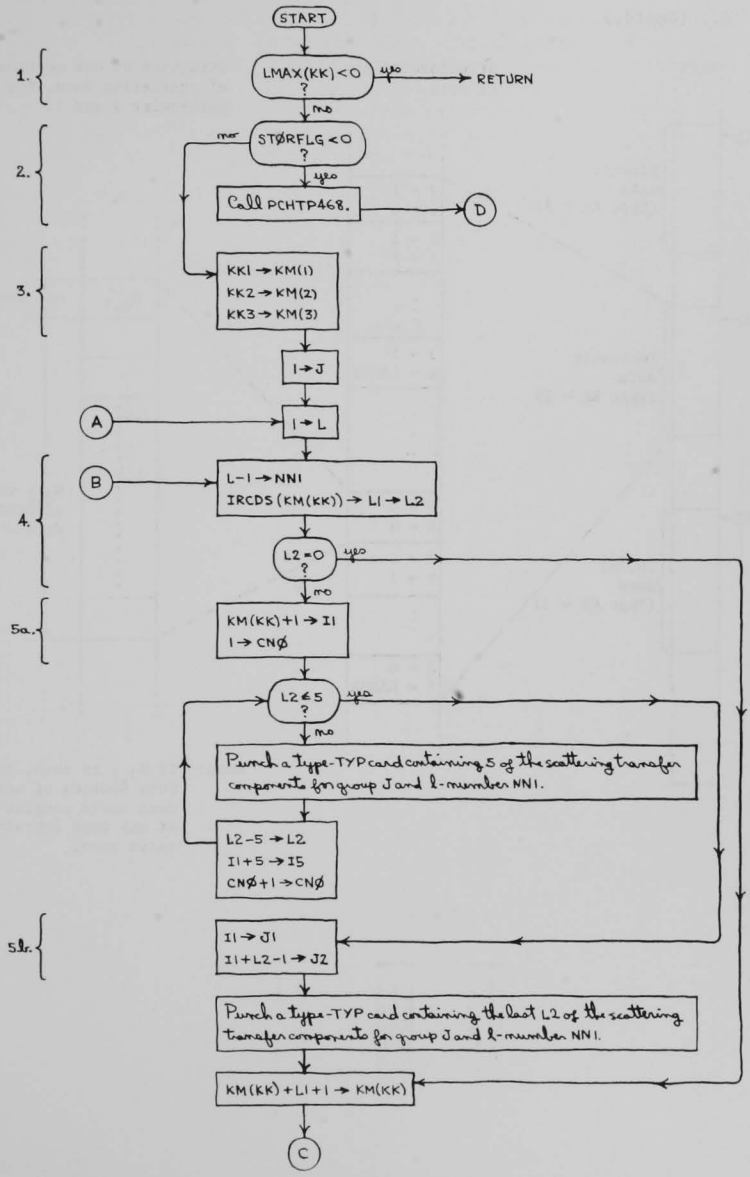
4. NN1 is the ℓ-number of the data cards to be punched next.

   The scattering data are stored in the array RCDS as shown in the diagram on p. 218. IRCDS(KM(KK)) = L2 gives the number of words of data for scattering-type KK, group J, and ℓ-number NN1. If L2 is zero, there are no such data, and no cards are punched. (See the note under the right-hand diagram on p. 218.)

5a,b. I1, J1, and J2 are indices used to retrieve the desired data from the array RCDS for punching.

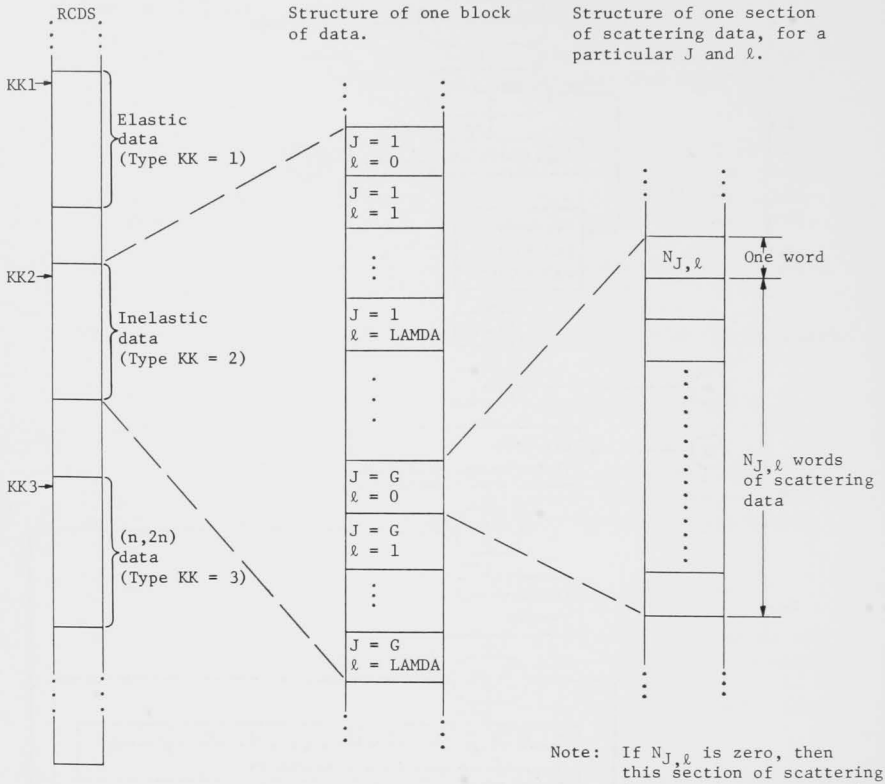
   CNO is the card number as punched on the data cards.

PUNCH468

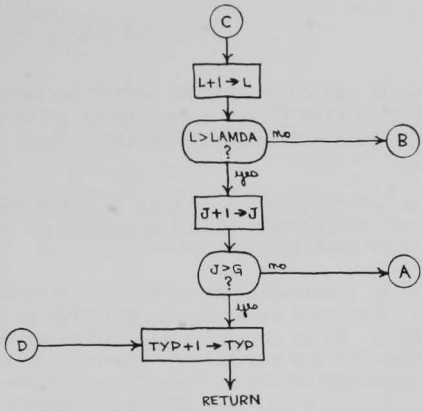

START
1.
LMAX(KK) < 0 ?
yes
RETURN
no
2.
STØRFLG < 0 ?
no
yes
Call PCHTP468.
D
3.
KK1 → KM(1)
KK2 → KM(2)
KK3 → KM(3)
1 → J
A
1 → L
4.
B
L-1 → NN1
IRCDS (KM(KK)) → L1 → L2
L2 = 0 ?
yes
no
5a.
KM(KK)+1 → I1
1 → CNØ
L2 ≤ 5 ?
yes
no
Punch a Type-TYP card containing 5 of the scattering transfer components for group J and l-number NN1.
L2-5 → L2
I1+5 → I5
CNØ+1 → CNØ
5b.
I1 → J1
I1+L2-1 → J2
Punch a Type-TYP card containing the last L2 of the scattering transfer components for group J and l-number NN1.
KM(KK)+L1+1 → KM(KK)
C

PUNCH468 (Contd.)

4. (Contd.)



Note: If $N_{J,\ell}$ is zero, then this section of scattering data would consist merely of one word containing the value zero.


C
L+1 → L
L > LAMDA ?
no
B
yes
J+1 → J
J > G ?
no
A
yes
D
TYP+1 → TYP
RETURN

PUTAWAY

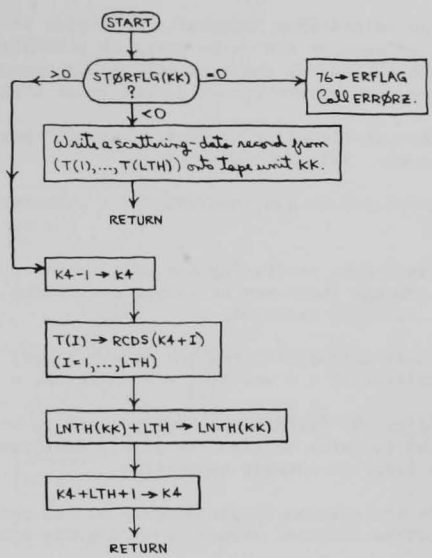
This subroutine is called by READ468 each time the scattering transfer components of scattering-type KK for a particular group and ℓ-number have been obtained from the type-4 (KK = 1) or type-6 (KK = 2) or type-8 (KK = 3) cross-section data cards.

This scattering information, consisting of LTH words, is to be stored either on tape unit KK [if STORFLG(KK) is negative] or in core in the array RCDS [if STORFLG(KK) is positive].

The quantity K4 is initialized in subroutine TESTSTOR for each of the three values of KK. (For each value of KK, TESTSTOR is called before the first call to PUTAWAY.) K4 is the index within the array RCDS of the first unused word of storage for the scattering transfer components of scattering-type KK. K4 is decremented by one to facilitate its use as an index, and later is incremented by LTH + 1, where LTH is the number of words in scattering information just stored.

The quantity LNTH(KK) is the total number of words of scattering information of scattering-type KK that have been stored. It will be used in subroutine WRITSCRH to retrieve this information when the material block for the material currently being processed is placed on tape unit 6.

PUTAWAY


START
STØRFLG(KK) ?
>0
=0
<0
76 → ERFLAG
Call ERRØRZ.
Write a scattering-data record from (T(1),...,T(LTH)) onto tape unit KK.
RETURN
K4-1 → K4
T(I) → RCDS(K4+I)
(I=1,...,LTH)
LNTH(KK)+LTH → LNTH(KK)
K4+LTH+1 → K4
RETURN

RCDLTH(L)

This routine is called from PNCH1MAT. Each time it is called, it calculates the length of each of the three portions [LENTH(I), I = 1, 2, 3] and the total length (RLTH), of the scattering-data record for ℓ-number L, that follows the group-constants record currently in BUFAREA.

LENTH(I) is zero if there are no data in the Ith portion of the scattering-data record. This can occur if:

(1) I-type scattering is not considered for this material [i.e., LMAX(I) negative].

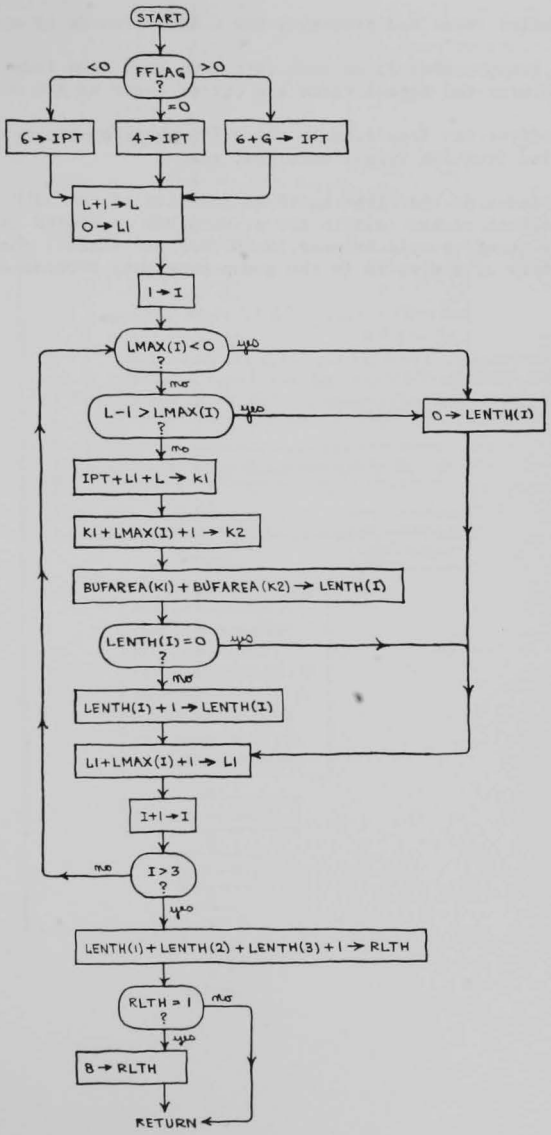
(2) There is no I-type scattering for this ℓ-number [i.e., L - 1 > LMAX(I)], though there may be I-type scattering data for lower ℓ-numbers for this material.

(3) There is only self-scattering for this ℓ-number and group [i.e., $F^{(L)}_{I,J}$ = BUFAREA(K1) = 0 and $G^{(L)}_{I,J}$ = BUFAREA(K2) = 0].

Note: The arrays BUFAREA and IBUFAREA are identical. BUFAREA is used to refer to floating-point quantities; IBUFAREA is used to refer to integer quantities.

Note: See the diagram in paragraph 1 of the notes to PRINT1 (p. 172) for the internal structure of a group-constants record.

RCDLTH(L)


START
FFLAG ?
<0
=0
>0
6 → IPT
4 → IPT
6+G → IPT
L+1 → L
0 → L1
1 → I
LMAX(I) < 0 ?
yes
no
L-1 > LMAX(I) ?
yes
no
0 → LENTH(I)
IPT+L1+L → K1
K1+LMAX(I)+1 → K2
BUFAREA(K1) + BUFAREA(K2) → LENTH(I)
LENTH(I) = 0 ?
yes
no
LENTH(I) + 1 → LENTH(I)
L1+LMAX(I)+1 → L1
I+1 → I
I > 3 ?
no
yes
LENTH(1) + LENTH(2) + LENTH(3) + 1 → RLTH
RLTH = 1 ?
no
yes
8 → RLTH
RETURN

RDSTOR1

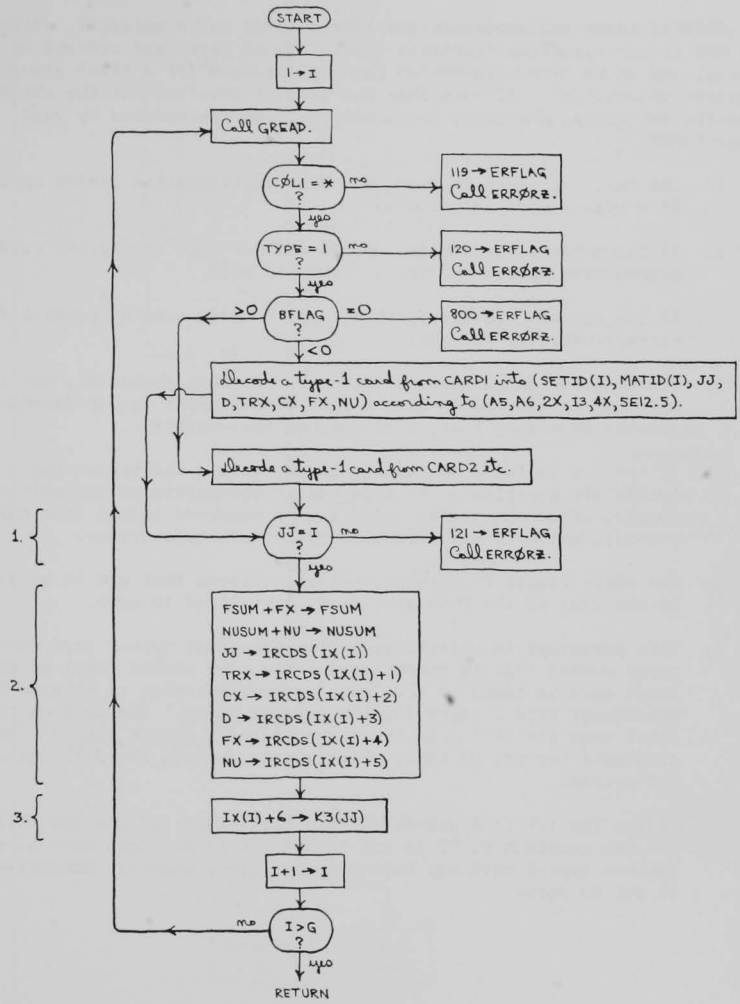
This routine reads and processes the G type-1 cards of a material.

1. The group number JJ on each card must equal the loop index I; otherwise the type-1 cards are out of order or are mispunched.

2. The first six locations in the JJ^{th} group-constants record are filled from the type-1 card just read.

3. The index of the first unfilled location in the JJ^{th} group-constants record (within the array RCDS) is stored in K3(JJ). This quantity will be used in RDSTOR2 and READ357 when the remainder of the cells in the group-constants records are filled.

RDSTØR1


START
1 → I
Call GREAD.
CØL1 = * ?
no
119 → ERFLAG
Call ERRØRZ.
yes
TYPE = 1 ?
no
120 → ERFLAG
Call ERRØRZ.
yes
BFLAG ?
>0
=0
800 → ERFLAG
Call ERRØRZ.
<0
Decode a type-1 card from CARD1 into (SETID(I), MATID(I), JJ, D, TRX, CX, FX, NU) according to (A5, A6, 2X, I3, 4X, SE12.5).
Decode a type-1 card from CARD2 etc.
1.
JJ = I ?
no
121 → ERFLAG
Call ERRØRZ.
yes
2.
FSUM + FX → FSUM
NUSUM + NU → NUSUM
JJ → IRCDS (IX(I))
TRX → IRCDS (IX(I)+1)
CX → IRCDS (IX(I)+2)
D → IRCDS (IX(I)+3)
FX → IRCDS (IX(I)+4)
NU → IRCDS (IX(I)+5)
3.
IX(I)+6 → K3(JJ)
I+1 → I
no
I > G ?
yes
RETURN

RDSTOR2

RDSTOR2 reads and processes the type-2 cards for a material, which contain the fission-spectrum fractions (chi). These cards are ordered by group-numbers, and up to five elements of the chi-spectrum for a given group can be placed on one card. If more than one card is required for the chi fractions for one group, the cards for a single group are ordered by card number (CNO).

1. The next data card is read, and the identification fields applicable to a type-2 card are decoded.

2. If the material is nonfissionable (FFLAG = 0), any type-2 cards present are spaced by, and a return is made.

   If the material is fissionable, type-2 cards must be present for the current material and set.

3. If the group number on the first card is zero (nonzero), the current material is assumed to have a group-independent (group-dependent) fission spectrum, i.e., a chi-vector (chi-matrix).

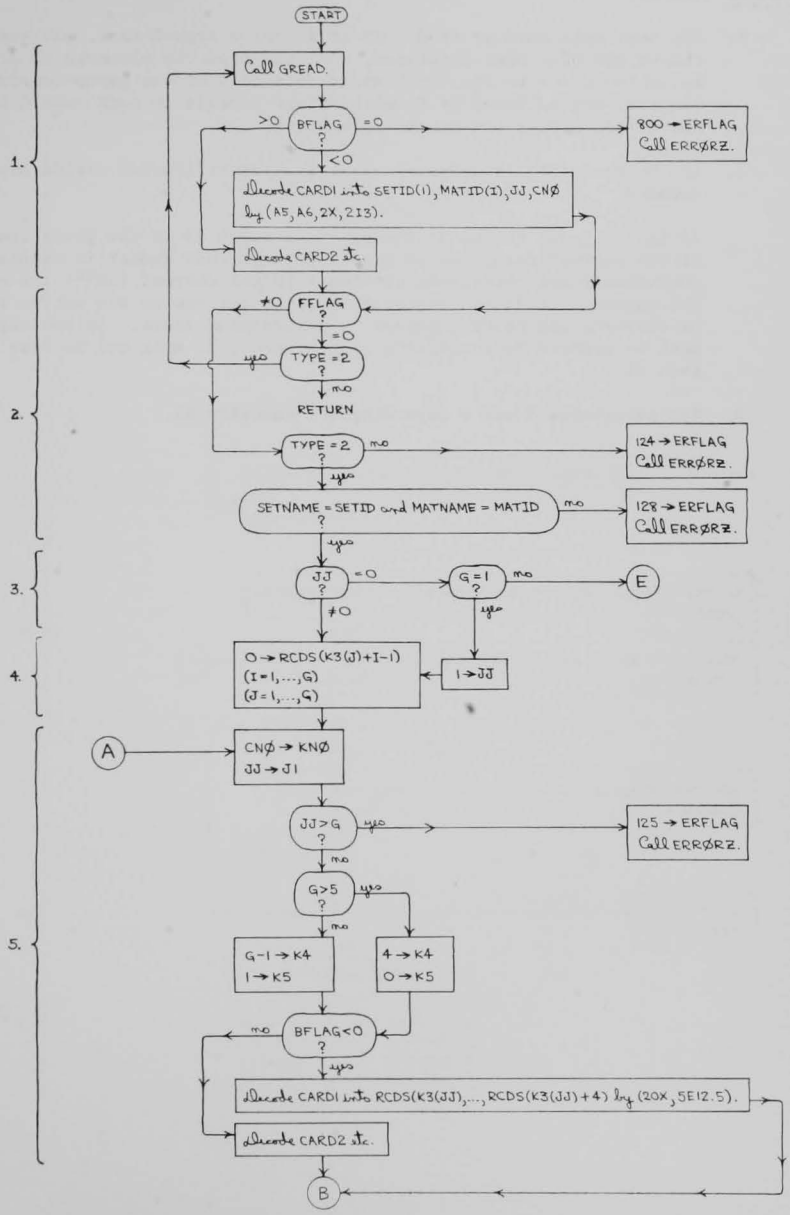
   If the set has only one energy group, then a chi-vector and a chi-matrix are identical. In this case a chi-matrix is unconditionally formed, since the vector would form a one-word record (the chi record), which is not allowable on certain tape drives.

4. The cells within the group-constants records that are to be occupied by the rows of the chi-matrix are initialized to zero.

5. This paragraph is entered each time the first type-2 card with a new group number (JJ) is encountered. The card number (CNO) on this first card is taken as the "base" sequence number to be used for all subsequent type-2 cards for the current group. The data on this first card are stored in the first K4 cells within the JJ^{th} group-constants record, of those which are to contain the JJ^{th} row of the chi-matrix.

   If the (up to) five values on this first card for the row will fill the row completely, K5 is set to nonzero to indicate that no subsequent type-2 card may have the same group number. Otherwise K5 is set to zero.

RDSTØR2


START
Call GREAD.
BFLAG ?
>0
=0
<0
800 → ERFLAG
Call ERRØRZ.
Decode CARD1 into SETID(1), MATID(1), JJ, CNØ by (A5, A6, 2X, 2I3).
Decode CARD2 etc.
FFLAG ?
≠0
=0
TYPE = 2 ?
yes
no
RETURN
TYPE = 2 ?
no
yes
124 → ERFLAG
Call ERRØRZ.
SETNAME = SETID and MATNAME = MATID ?
no
yes
128 → ERFLAG
Call ERRØRZ.
JJ ?
=0
≠0
G = 1 ?
no
yes
E
1 → JJ
0 → RCDS(K3(J)+I-1)
(I = 1, ..., G)
(J = 1, ..., G)
A
CNØ → KNØ
JJ → J1
JJ > G ?
yes
no
125 → ERFLAG
Call ERRØRZ.
G > 5 ?
yes
no
G-1 → K4
1 → K5
4 → K4
0 → K5
BFLAG < 0 ?
no
yes
Decode CARD1 into RCDS(K3(JJ), ..., RCDS(K3(JJ)+4) by (20X, 5E12.5).
Decode CARD2 etc.
B
1.
2.
3.
4.
5.

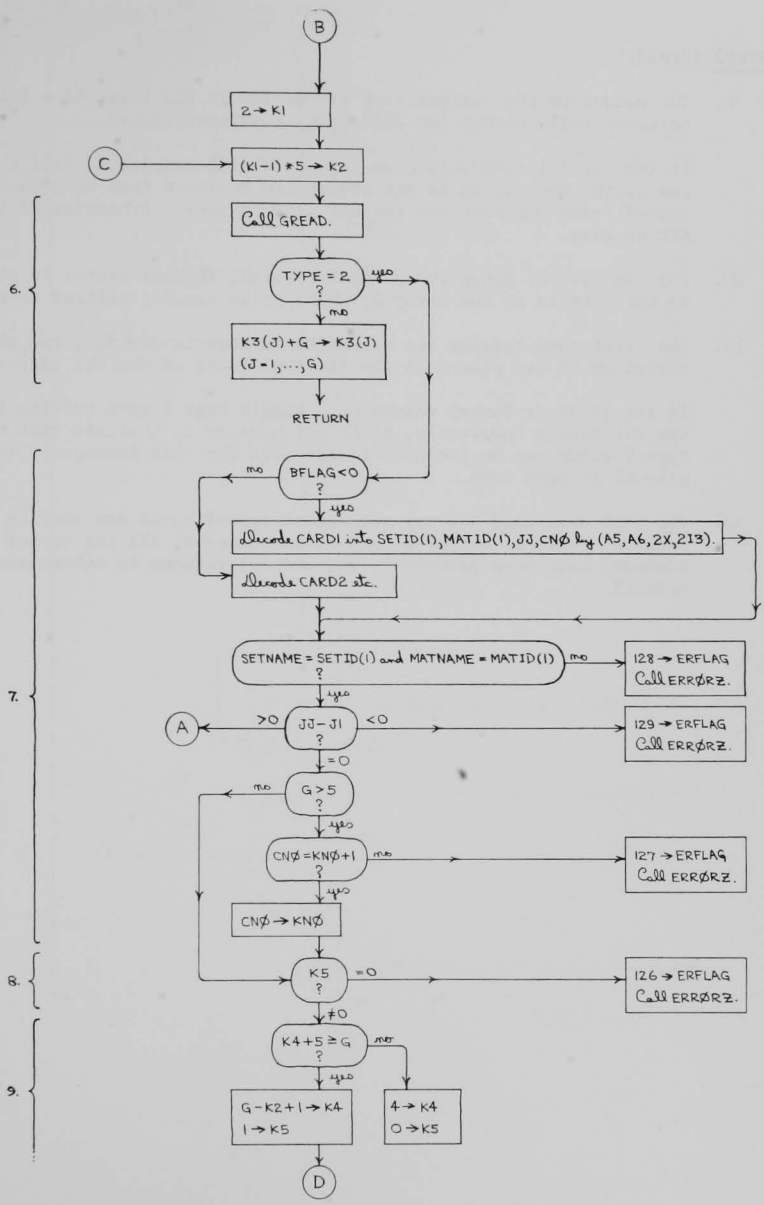
RDSTOR2 (Contd.)

6. The next data card is read. If it is not a type-2 card, all supplied chi-values have been processed. In this case the elements of array K3, which point to the first unoccupied cell of the group-constants records, are advanced by G cells, since G cells in each record are now filled with a row of the chi-matrix.

7. If the next card is a type-2 card, its identification fields are decoded.

   J1 is the group number of the previous card; JJ is the group number of the current card. If JJ equals J1, the card number is checked for consistency and processing continues in the current ($J1^{th}$) row of the chi-matrix. If JJ is greater than J1, a new row of the matrix is to be started, and control passes to paragraph 5 above. As the cards must be ordered by increasing group number, JJ must not be less than J1.

8. See paragraphs 5 and 9 regarding the quantity K5.


B
2 → K1
C
(K1−1) * 5 → K2
Call GREAD.
6.
TYPE = 2 ?
yes
no
K3(J)+G → K3(J)
(J = 1, ..., G)
RETURN
7.
BFLAG<0 ?
no
yes
Decode CARD1 into SETID(1), MATID(1), JJ, CNØ by (A5, A6, 2X, 2I3).
Decode CARD2 etc.
SETNAME = SETID(1) and MATNAME = MATID(1) ?
no
128 → ERFLAG
Call ERRØRZ.
yes
A
>0
JJ − J1 ?
<0
129 → ERFLAG
Call ERRØRZ.
=0
G > 5 ?
no
yes
CNØ = KNØ+1 ?
no
127 → ERFLAG
Call ERRØRZ.
yes
CNØ → KNØ
8.
K5 ?
=0
126 → ERFLAG
Call ERRØRZ.
≠0
9.
K4+5 ≥ G ?
no
yes
G − K2 + 1 → K4
1 → K5
4 → K4
0 → K5
D

RDSTOR2 (Contd.)

9. The values on the current card are stored in the first K4 + 1 unoccupied cells within the JJ^{th} group-constants record.

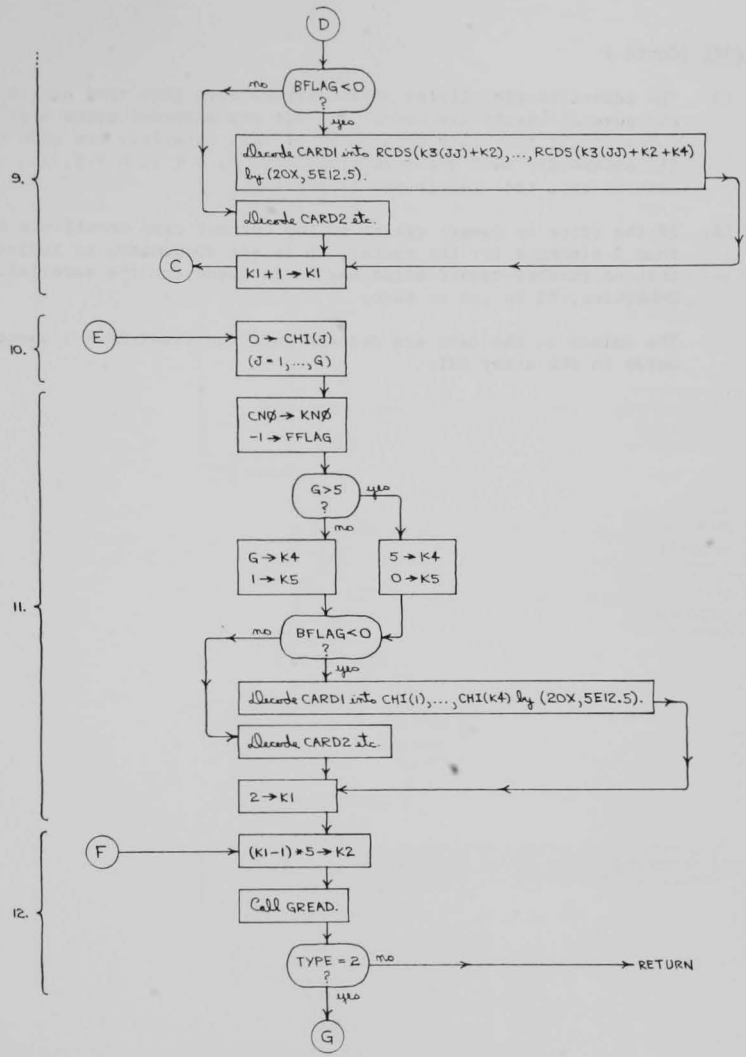
   If the (up to) five values on this card will completely fill this row of the matrix, K5 is set nonzero to indicate that no subsequent type-2 cards may have the current group number. Otherwise K5 is set to zero.

10. For the case of group-independent fission, the chi vector is stored in the G cells of the array G; these cells are initialized to zero.

11. The first card bearing the chi-vector values is decoded, and the values on it are placed in the first K4 cells of the chi vector.

    If the (five or fewer) values on a single type-2 card suffice to fill the chi vector completely, K5 is set nonzero to indicate that no more type-2 cards may be included in the data for this isotope. Otherwise K5 is made zero.

12. The next data card and any subsequent type-2 cards are read in this section. When a nontype-2 card is encountered, all the vector elements have been processed, and control returns to subroutine ADDAMAT.

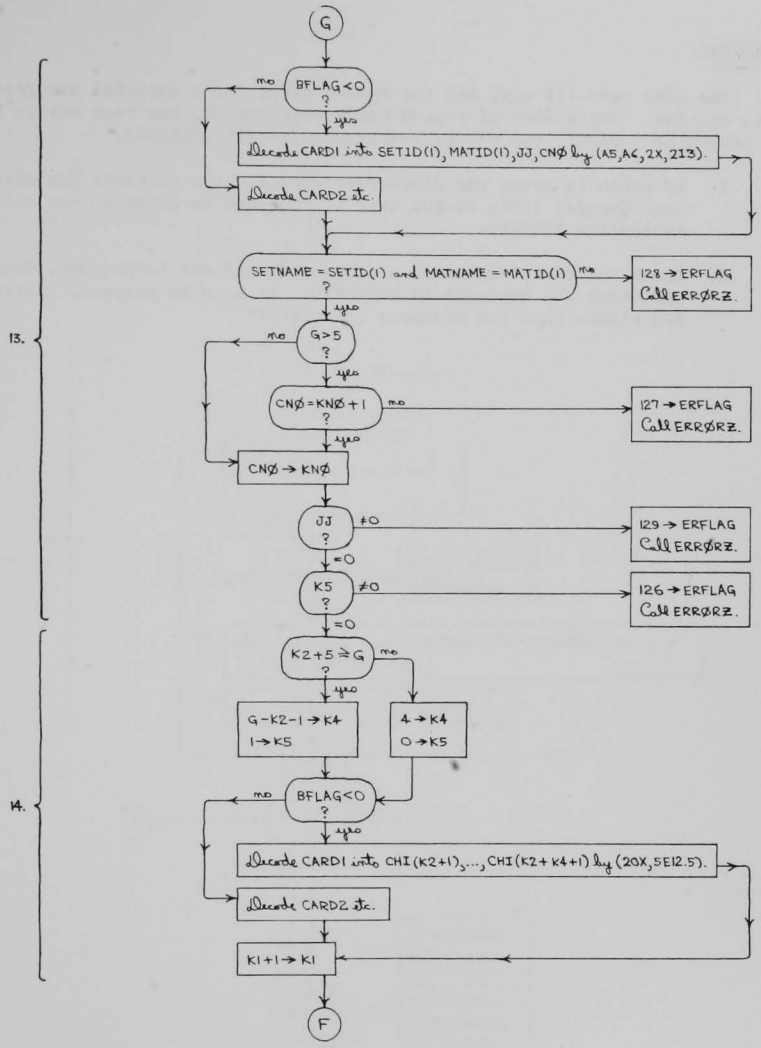

D
BFLAG<0 ?
no
yes
Decode CARD1 into RCDS(K3(JJ)+K2),...,RCDS(K3(JJ)+K2+K4) by (20X,5E12.5).
Decode CARD2 etc.
C
K1+1 → K1
9.
E
0 → CHI(J)
(J = 1,...,G)
10.
CNØ → KNØ
-1 → FFLAG
G>5 ?
yes
no
G → K4
1 → K5
5 → K4
0 → K5
11.
BFLAG<0 ?
no
yes
Decode CARD1 into CHI(1),...,CHI(K4) by (20X,5E12.5).
Decode CARD2 etc.
2 → K1
F
(K1-1)*5 → K2
Call GREAD.
12.
TYPE = 2 ?
no
RETURN
yes
G

RDSTOR2 (Contd.)

13. The identification fields of the type-2 card just read are decoded, and several checks are made: the set and material names must match those of all the other data cards of this material; the card numbers (if necessary) must occur in a sequence N, N + 1, N + 2, ... ; K5 must be zero (see paragraphs 11 and 14).

14. If the (five or fewer) values on the current card constitute more than G elements for the vector, K5 is set to nonzero to indicate e that no further type-2 cards may be included for the material. Otherwise, K5 is set to zero.

    The values on the card are decoded into the first K4 + 1 unoccupied words in the array CHI.


G
BFLAG<0 ?
no
yes
Decode CARD1 into SETID(1), MATID(1), JJ, CNØ by (A5,A6,2X,2I3).
Decode CARD2 etc.
SETNAME = SETID(1) and MATNAME = MATID(1) ?
no
128 → ERFLAG
Call ERRØRZ.
yes
13.
G>5 ?
no
yes
CNØ = KNØ + 1 ?
no
127 → ERFLAG
Call ERRØRZ.
yes
CNØ → KNØ
JJ ?
≠0
129 → ERFLAG
Call ERRØRZ.
=0
K5 ?
≠0
126 → ERFLAG
Call ERRØRZ.
=0
K2+5 ≥ G ?
no
yes
G-K2-1 → K4
1 → K5
4 → K4
0 → K5
14.
BFLAG<0 ?
no
yes
Decode CARD1 into CHI(K2+1),..., CHI(K2+K4+1) by (20X,5E12.5).
Decode CARD2 etc.
K1+1 → K1
F

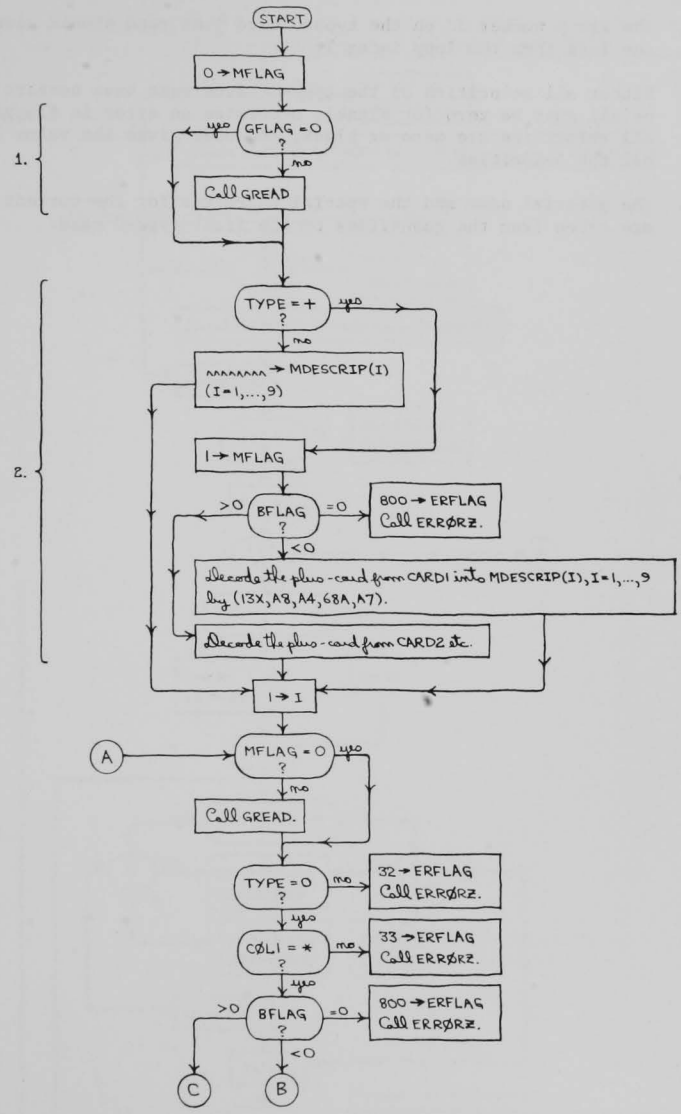
READZERO

The plus card (if any) and the type-0 cards for a material are read by this routine. The number of type-0 cards expected, K, has been set in either of ADDSET or ADDAMAT, the two subroutines that call READZERO.

1. If GFLAG is zero, the first data card for the material has already been decoded (this is the case if the path to READZERO was via subroutine MODSET).

2. Check for the presence of a plus-card. If one is present, decode it and store the contents in MDESCRIP. If none is present, insert BCD blanks into the elements of MDESCRIP.

READZERØ

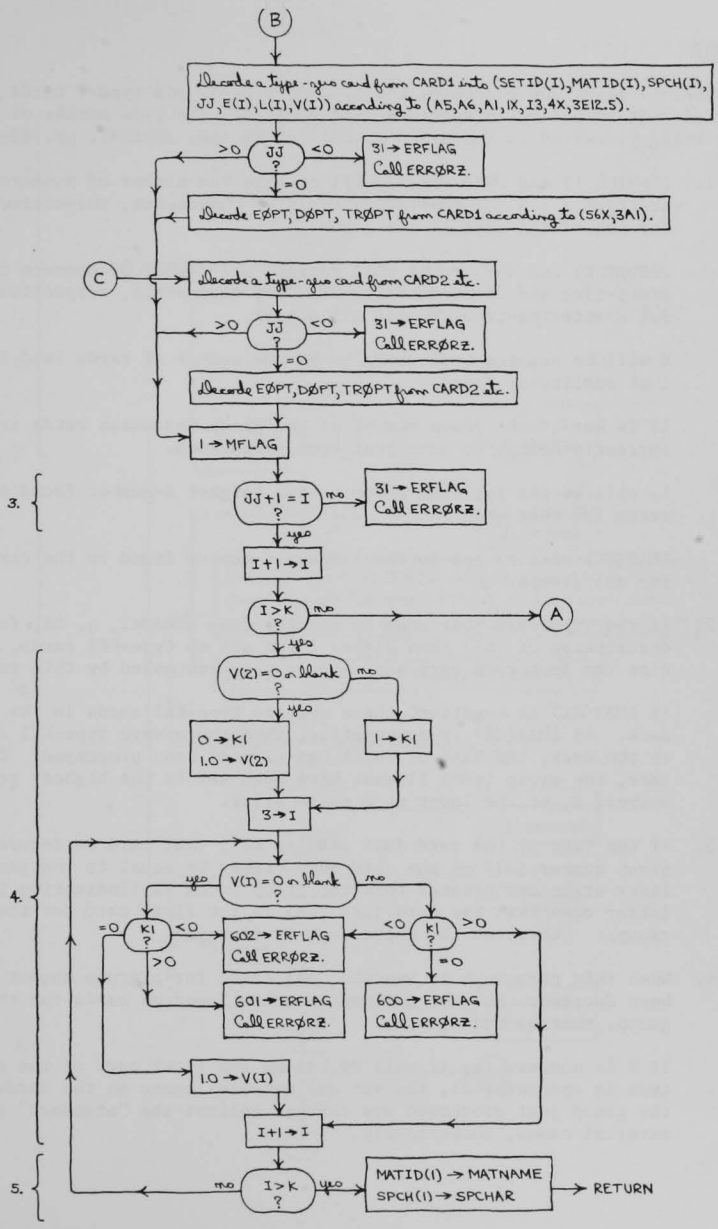

START
0 → MFLAG
GFLAG = 0 ?
yes
no
Call GREAD
1.
TYPE = + ?
yes
no
^^^^^^^^ → MDESCRIP(I)
(I = 1,...,9)
1 → MFLAG
2.
BFLAG ?
>0
=0
<0
800 → ERFLAG
Call ERRØRZ.
Decode the plus-card from CARD1 into MDESCRIP(I), I = 1,...,9
by (13X, A8, A4, 68A, A7).
Decode the plus-card from CARD2 etc.
1 → I
A
MFLAG = 0 ?
yes
no
Call GREAD.
TYPE = 0 ?
no
32 → ERFLAG
Call ERRØRZ.
yes
CØL1 = * ?
no
33 → ERFLAG
Call ERRØRZ.
yes
BFLAG ?
>0
=0
<0
800 → ERFLAG
Call ERRØRZ.
C
B

READZERO (Contd.)

3. The group number JJ on the type-0 card just read should always be one less than the loop index I.

4. Either all velocities on the type-0 cards must have nonzero values, or all must be zero (or blank); otherwise an error is flagged. If all velocities are zero or blank, the code gives the value 1.0 to all the velocities.

5. The material name and the special character for the current material are taken from the quantities on the first type-0 card.


B
Decode a type-gno card from CARD1 into (SETID(I), MATID(I), SPCH(I), JJ, E(I), L(I), V(I)) according to (A5, A6, A1, 1X, I3, 4X, 3E12.5).
JJ ?
>0
<0
=0
31 → ERFLAG
Call ERRØRZ.
Decode EØPT, DØPT, TRØPT from CARD1 according to (56X, 3A1).
C
Decode a type-gno card from CARD2 etc.
JJ ?
>0
<0
=0
31 → ERFLAG
Call ERRØRZ.
Decode EØPT, DØPT, TRØPT from CARD2 etc.
1 → MFLAG
3.
JJ+1 = I ?
no
yes
31 → ERFLAG
Call ERRØRZ.
I+1 → I
I > K ?
no
yes
A
V(2) = 0 or blank ?
no
yes
0 → K1
1.0 → V(2)
1 → K1
3 → I
4.
V(I) = 0 or blank ?
yes
no
K1 ?
=0
<0
>0
602 → ERFLAG
Call ERRØRZ.
601 → ERFLAG
Call ERRØRZ.
K1 ?
<0
>0
=0
600 → ERFLAG
Call ERRØRZ.
1.0 → V(I)
I+1 → I
5.
I > K ?
no
yes
MATID(1) → MATNAME
SPCH(1) → SPCHAR
RETURN

READ357

READ357 reads and processes the type-3, type-5, and type-7 cards for a material. It is called three times from ADDAMAT. The type number of the cards being processed is contained in KK1 and KK (see ADDAMAT, pp. 88-91).

1. UPSM(KK,I) and DWNSM(KK,I) will contain the number of nonzero upscattering and downscattering transfer components, respectively, for scattering-type KK and group I.

   UPSUM(KK) and DWNSUM(KK) will contain the number of nonzero upscattering and downscattering transfer components, respectively, for scattering-type KK, for all groups.

   K will be set for each group to be the number of cards read for that particular group.

   I1 is set to the group number of the group for which cards are currently being, or have just been, processed.

   LL will be set for each group to the largest ℓ-number found on cards for that group.

   LMAX(KK) will be set to the largest ℓ-number found on the cards for all groups.

2. If the type card just read is not KK1 (see ADDAMAT, p. 88, for a description of KK1) then either there are no type-KK1 cards, or else the last such card has already been processed by this routine.

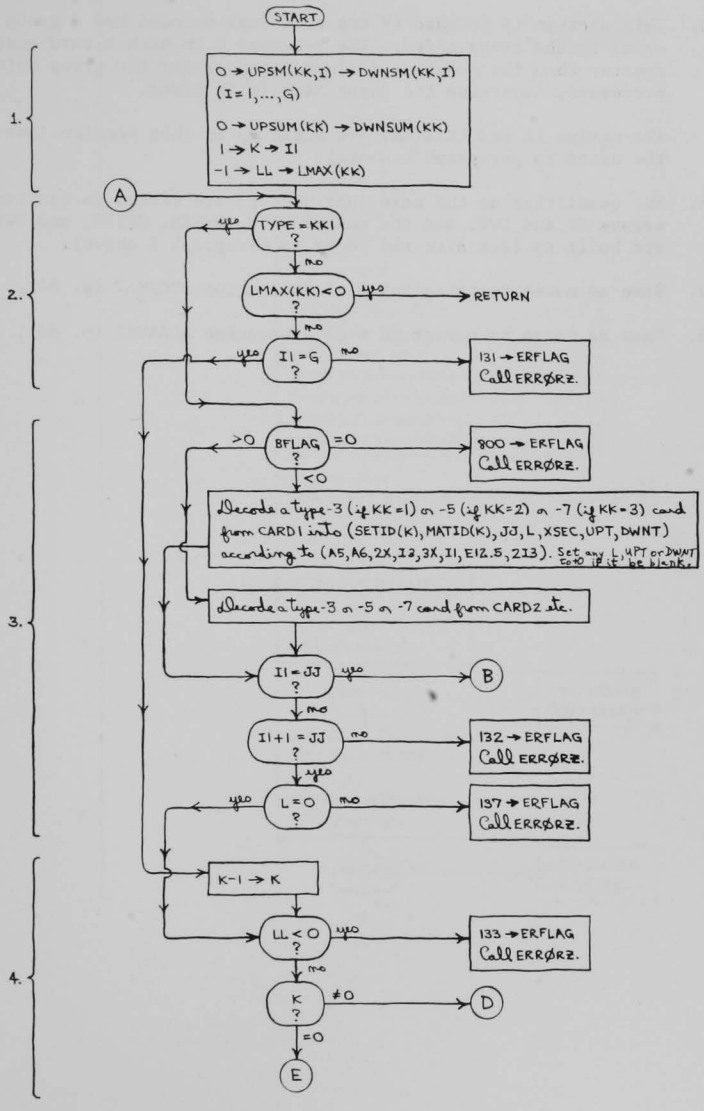
   If LMAX(KK) is negative, there were no type-KK1 cards in the input deck. If LMAX(KK) is nonnegative, then there were type-KK1 cards in the deck, the last of which has already been processed. In this case, the group index I1 must have been set to the highest group number, G, or the input deck is in error.

3. If the type of the card just read is KK1, that card is decoded. The group number (JJ) on the card must either be equal to the group index or be one greater than the group index I1 (indicating in the latter case that the card just read is the first card for the next group). Otherwise the input deck is in error.

4. When this paragraph is reached, all cards for a group number have been decoded. LL, the maximum ℓ-number found on cards for that group, must be nonnegative.

   If K is nonzero (as it will be unless the first card of the next type is encountered), the set and material names on the cards for the group just processed are checked against the "standard" set and material names, respectively.

READ357


START
0 → UPSM(KK,I) → DWNSM(KK,I)
(I = 1,...,G)
0 → UPSUM(KK) → DWNSUM(KK)
1 → K → I1
-1 → LL → LMAX(KK)
1.
A
TYPE = KK1 ?
yes
no
LMAX(KK) < 0 ?
yes
RETURN
no
I1 = G ?
yes
no
131 → ERFLAG
Call ERRØRZ.
2.
BFLAG ?
>0
=0
800 → ERFLAG
Call ERRØRZ.
<0
Decode a type-3 (if KK = 1) or -5 (if KK = 2) or -7 (if KK = 3) card from CARD1 into (SETID(K), MATID(K), JJ, L, XSEC, UPT, DWNT) according to (A5, A6, 2X, I2, 3X, I1, E12.5, 2I3). Set any L, UPT or DWNT to 0 if it be blank.
Decode a type-3 or -5 or -7 card from CARD2 etc.
3.
I1 = JJ ?
yes
B
no
I1 + 1 = JJ ?
no
132 → ERFLAG
Call ERRØRZ.
yes
L = 0 ?
yes
no
137 → ERFLAG
Call ERRØRZ.
K - 1 → K
LL < 0 ?
yes
133 → ERFLAG
Call ERRØRZ.
no
4.
K ?
≠0
D
=0
E

READ357 (Contd.)

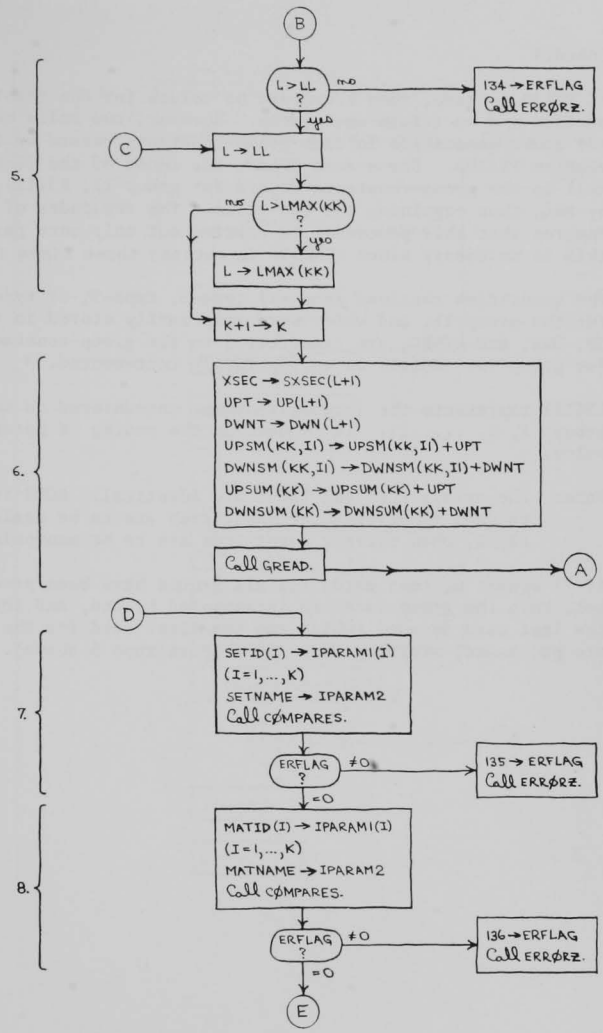
5. This section is reached if the card just decoded had a group number equal to the group index. The ℓ-number L on such a card must be greater than the previous maximum ℓ-number for the group being processed; otherwise the input deck is in error.

   The maxima LL and LMAX(KK) are built up in this section (see also the notes to paragraph 1 above).

6. The quantities on the card just decoded are stored in the temporary arrays UP and DWN, and the maxima UPSM, DWNSM, UPSUM, and DWNSUM are built up (see also the notes to paragraph 1 above).

7. Same as notes to paragraph 5 of subroutine ADDAMAT (p. 84).

8. Same as notes to paragraph 6 of subroutine ADDAMAT (p. 84).


B
L > LL ?
no
134 → ERFLAG
Call ERRØRZ.
yes
C
L → LL
5.
L > LMAX(KK) ?
no
yes
L → LMAX (KK)
K+1 → K
XSEC → SXSEC(L+1)
UPT → UP(L+1)
DWNT → DWN (L+1)
UPSM(KK,II) → UPSM(KK,II) + UPT
DWNSM (KK,II) → DWNSM(KK,II) + DWNT
UPSUM (KK) → UPSUM (KK) + UPT
DWNSUM (KK) → DWNSUM(KK) + DWNT
6.
Call GREAD.
A
D
SETID(I) → IPARAM1(I)
(I = 1, ..., K)
SETNAME → IPARAM2
Call CØMPARES.
7.
ERFLAG ?
≠0
135 → ERFLAG
Call ERRØRZ.
=0
MATID (I) → IPARAM1(I)
(I = 1, ..., K)
MATNAME → IPARAM2
Call CØMPARES.
8.
ERFLAG ?
≠0
136 → ERFLAG
Call ERRØRZ.
=0
E

READ357 (Contd.)

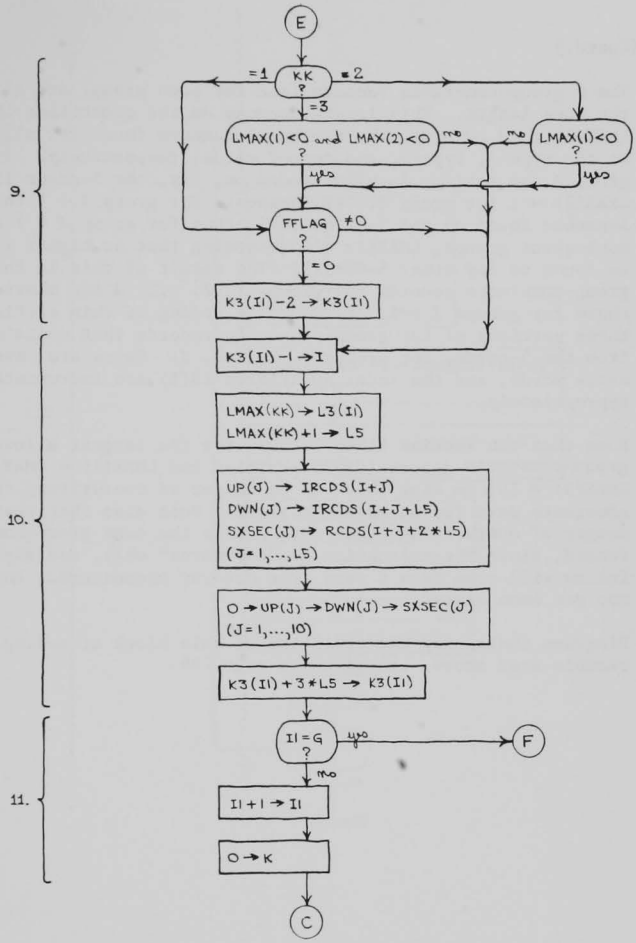
9. If FFLAG is zero, then there are no values for the fission cross section and nu (ν) in each group. However, two cells were reserved for these quantities in each group-constants record by the subroutine FIXADD. For a zero FFLAG, the index of the first unused cell in the group-constants record for group I1, K3(I1), is reduced by two, thus regaining the two cells. The remainder of the coding insures that this procedure is carried out only once per material; this is necessary since READ357 is entered three times from ADDAMAT.

10. The quantities obtained from all type-3, type-5, or type-7 cards for the group I1, and which were temporarily stored in the arrays UP, DWN, and SXSEC, are transferred to the group-constants record for group I1. K3(I1) is appropriately incremented.

    L3(I1) represents the largest ℓ-number encountered on the cards for groups 1, 2, ..., I1. It is used in the coding of paragraph 12 below.

    Note: The arrays RCDS and IRCDS are identical. RCDS is referred to when floating-point quantities are to be dealt with; IRCDS, when integer quantities are to be manipulated.

11. If I1 equals G, then cards for all groups have been processed. If not, then the group index is incremented by one, and the data from the last card decoded (which was the first card for the next group) are processed, starting at point C (paragraph 5 above).

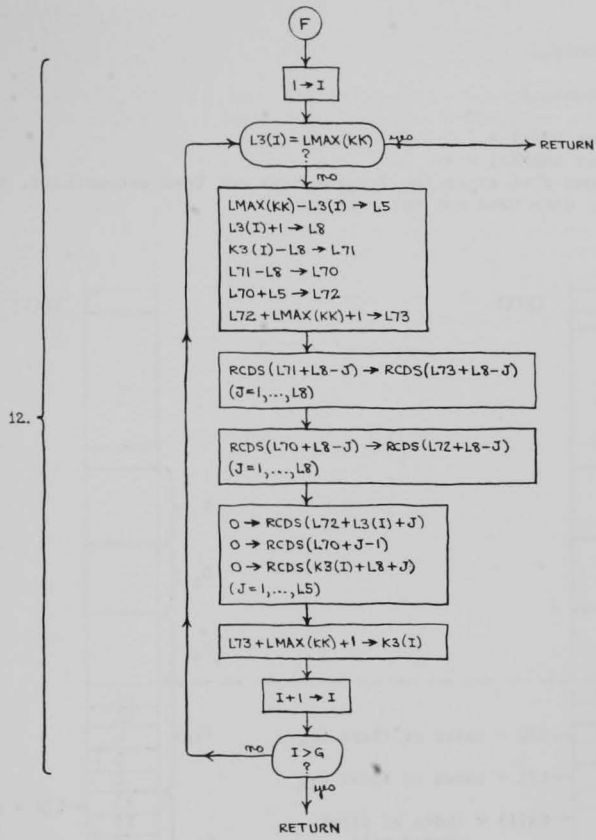

E
KK ?
=1
=2
=3
LMAX(1)<0 and LMAX(2)<0 ?
no
no
LMAX(1)<0 ?
yes
yes
9.
FFLAG ?
≠0
=0
K3(I1)-2 → K3(I1)
K3(I1)-1 → I
LMAX(KK) → L3(I1)
LMAX(KK)+1 → L5
10.
UP(J) → IRCDS(I+J)
DWN(J) → IRCDS(I+J+L5)
SXSEC(J) → RCDS(I+J+2*L5)
(J=1,...,L5)
0 → UP(J) → DWN(J) → SXSEC(J)
(J=1,...,10)
K3(I1)+3*L5 → K3(I1)
I1=G ?
yes
F
no
11.
I1+1 → I1
0 → K
C

READ357 (Contd.)

12. The G group-constants records (one for each group) are all to have the same length. This length depends on the quantities LMAX(1), LMAX(2), and LMAX(3), the maximum ℓ-numbers found for all groups on the type-3, type-5, and type-7 cards, respectively. If for group J the maximum ℓ-number found on, say, the 5-cards is 1, then LMAX(2) = 1 for group J. If, however, for group J + 1 the maximum ℓ-number found on the 5-cards is 4, then for group J + 1 and all subsequent groups, LMAX(2) = 4 (assuming that no higher ℓ-number is found on any other 5-cards). The result of this is that the group-constants records for groups 1, 2, ..., J are shorter than those for groups J + 1, ..., G. The coding of this section lengthens those portions of the group-constants records that contain the data from the 5-cards, for groups 1, 2, ..., J. Zeros are inserted in the extra words, and the index quantities K3(I) are incremented appropriately.

    Note that the routine FIXADD allows for the largest allowable group-constants record (FFLAG positive and LMAX(1) = LMAX(2) = LMAX(3) = 10] so that there is no danger of overwriting the group-constants data for an adjacent group. Note also that there is no danger of overwriting other data within the same group-constants record, since the relocation is "downwards" only, and any data to follow will come from a card type not yet encountered, and so have not yet been stored.

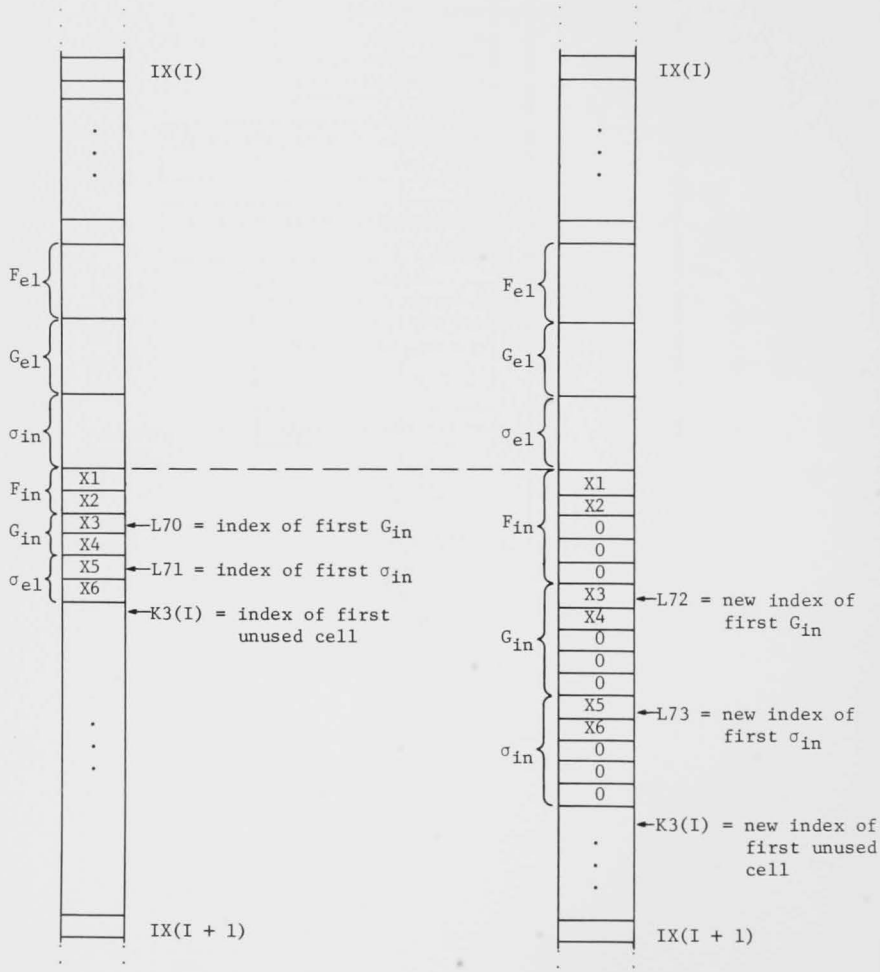
    Diagrams indicating the operation of this block of coding, for the example used above, are given on page 246.


F
1 → I
L3(I) = LMAX(KK) ?
yes
RETURN
no
LMAX(KK) − L3(I) → L5
L3(I) + 1 → L8
K3(I) − L8 → L71
L71 − L8 → L70
L70 + L5 → L72
L72 + LMAX(KK) + 1 → L73
RCDS(L71 + L8 − J) → RCDS(L73 + L8 − J)
(J = 1, ..., L8)
RCDS(L70 + L8 − J) → RCDS(L72 + L8 − J)
(J = 1, ..., L8)
0 → RCDS(L72 + L3(I) + J)
0 → RCDS(L70 + J − 1)
0 → RCDS(K3(I) + L8 + J)
(J = 1, ..., L5)
L73 + LMAX(KK) + 1 → K3(I)
I + 1 → I
I > G ?
no
yes
RETURN
12.

READ357 (Contd.)

12. (Contd.)

Let L3(I) = 1.
Let LMAX(2) = 4.
Note that since the 7-cards have not been encountered, the (n,2n) data have not yet been stored.



The structure of the Ith group-constants record before entering the coding of paragraph 12.

The structure of the Ith group-constants record after leaving the coding of paragraph 12.

READ468

READ468 reads and processes the type-4, type-6, and type-8 cards for a material. It is called three times from ADDAMAT, once for each card type; the type currently being processed is given by KK1 and KK (see ADDAMAT pp. 88-91):

1. Loop A is traversed G times over the range of values I = 1, ..., G. Each time the loop is traversed the type-4 (if KK = 1) or type-6 (if KK = 2) or type-8 (if KK = 3) cards for a single group are read and processed.

2. If UPSM(KK,I1) and DWNSM(KK,I1) are both zero, then for scattering-type* KK there is only self-scattering for group I1, for all ℓ-numbers. This means that any type-4 or type-6 or type-8 cards for group I1 will be spaced past and ignored, and that

$$\mathrm{TSUM(I1,KK)} \equiv \sum_{K} T_{KK}^{(0)}(\mathrm{I1} \rightarrow K) = 1.0.$$

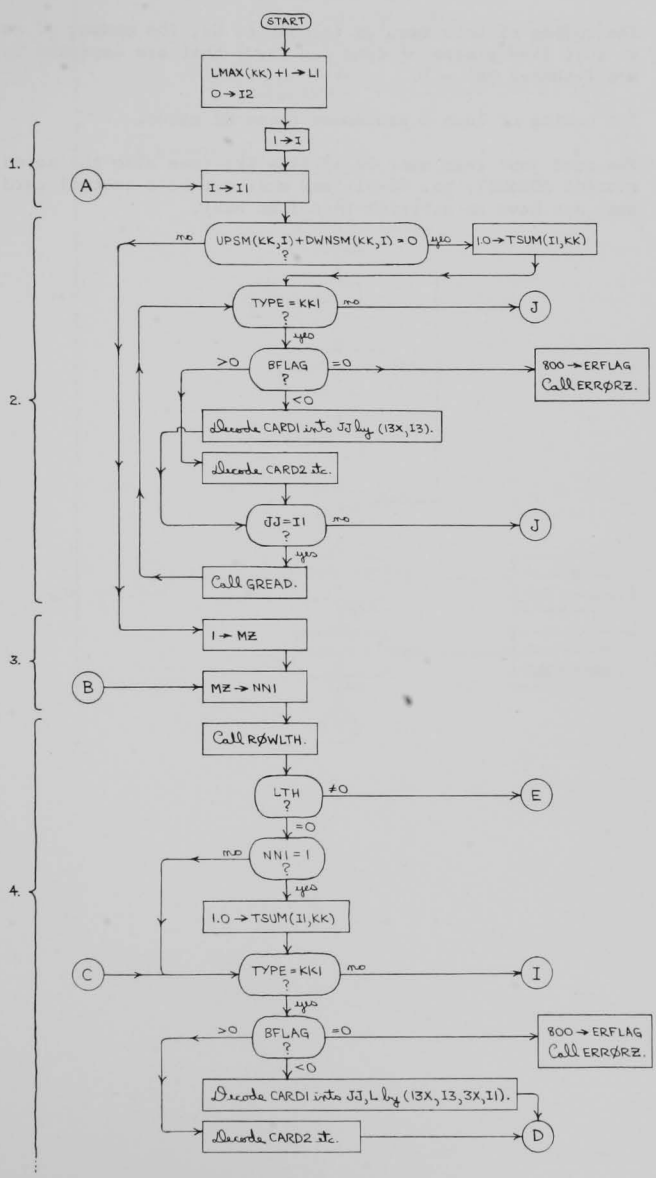
3. Loop B is traversed LMAX(KK) + 1 times each time that loop A is traversed once. When loop B is traversed a single time, the type-4 or type-6 or type-8 cards for a single group and a single ℓ-number are read and processed.

4. ROWLTH calculates LTH, the number of words of scattering information expected for group I1 and ℓ-number NN1 - 1. If LTH is zero, there is only self-scattering for this combination of group and ℓ-number, and hence any type-4 or type-6 or type-8 card will be spaced past and ignored. Again, as in paragraph 2 above, TSUM(I1,KK) = 1.0.

---

*KK = 1 corresponds to elastic scattering.
KK = 2 corresponds to inelastic scattering.
KK = 3 corresponds to (n,2n) scattering.

READ468

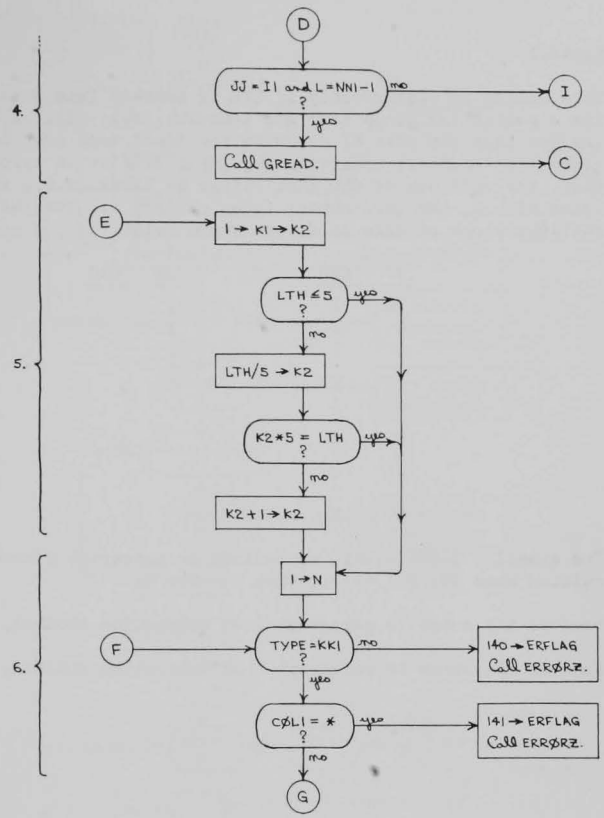

START
LMAX(KK)+1 → L1
0 → I2
1 → I
1.
A
I → I1
UPSM(KK,I)+DWNSM(KK,I) = 0 ?
no
yes
1.0 → TSUM(I1,KK)
TYPE = KK1 ?
no
J
yes
BFLAG ?
>0
=0
800 → ERFLAG
Call ERRØRZ.
<0
2.
Decode CARD1 into JJ by (13X,I3).
Decode CARD2 etc.
JJ=I1 ?
no
J
yes
Call GREAD.
1 → MZ
3.
B
MZ → NN1
Call RØWLTH.
LTH ?
≠0
E
=0
NN1 = 1 ?
no
yes
4.
1.0 → TSUM(I1,KK)
C
TYPE = KK1 ?
no
I
yes
BFLAG ?
>0
=0
800 → ERFLAG
Call ERRØRZ.
<0
Decode CARD1 into JJ,L by (13X,I3,3X,I1).
Decode CARD2 etc.
D

READ468 (Contd.)

5. The coding of this section calculates K2, the number of cards (with at most five pieces of data per card) that are expected for group I1 and ℓ-number NN1 - 1.

   The coding of loop C processes these K2 cards.

6. The card just read must be of type KK1 (see also the notes to subroutine ADDAMAT, pp. 88-91) and must not be a control card (i.e., it must not have an asterisk in column one).


D
JJ = I1 and L = NN1 - 1 ?
no
I
yes
Call GREAD.
C
4.
E
1 → K1 → K2
LTH ≤ 5 ?
yes
no
LTH/5 → K2
K2 * 5 = LTH ?
yes
no
K2 + 1 → K2
5.
1 → N
F
TYPE = KK1 ?
no
140 → ERFLAG
Call ERRØRZ.
yes
CØL1 = * ?
yes
141 → ERFLAG
Call ERRØRZ.
no
G
6.

READ468 (Contd.)

7. This coding is reached only if LTH is nonzero (see paragraph 4 above) for a particular group (I1) and ℓ-number (NN1 - 1). LTH nonzero implies that the next K2 cards in the input deck must bear this group and ℓ-number; otherwise the input deck is in error. In addition, the card number CNO must either be blank (i.e., zero) or must agree with N, the card-number index of loop C. The two allowable configurations of data cards are shown below.

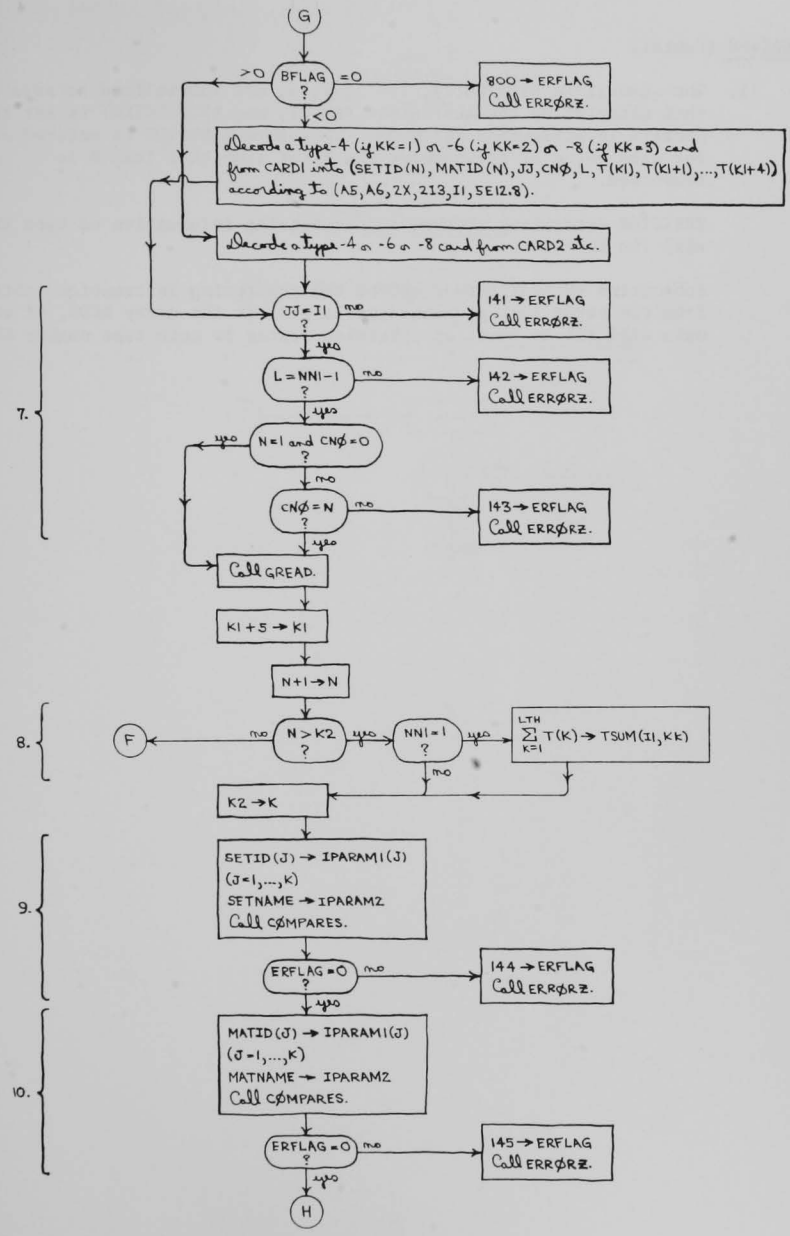
JJ	CNO
1	1
G	2
⋮	⋮
1	K2
⋮	⋮
G	1
G	2
⋮	⋮
G	K2

JJ	CNO
1	
2	
⋮	
G	

8. The quantity TSUM(I1,KK), as defined in paragraph 2 above, is calculated when NN1 = 1 (i.e., when ℓ = 0).

9. Same as the notes to paragraph 5 of subroutine ADDAMAT, p. 84.

10. Same as the notes to paragraph 6 of subroutine ADDAMAT, p. 84.

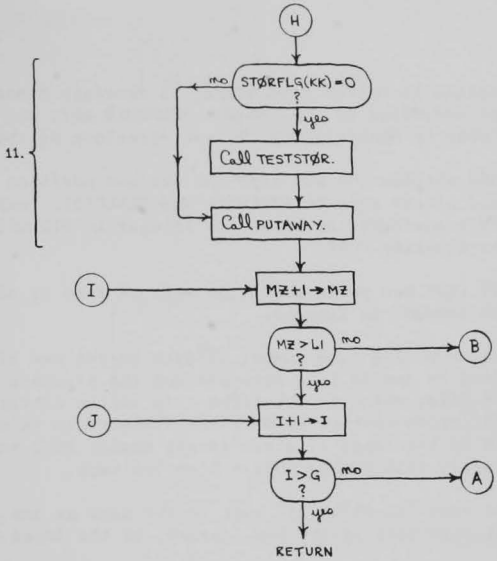

G
BFLAG ?
>0
=0
800 → ERFLAG
Call ERRØRZ.
<0
Decode a type-4 (if KK=1) or -6 (if KK=2) or -8 (if KK=3) card from CARD1 into (SETID(N), MATID(N), JJ, CNØ, L, T(K1), T(K1+1),..., T(K1+4)) according to (A5, A6, 2X, 2I3, I1, 5E12.8).
Decode a type-4 or -6 or -8 card from CARD2 etc.
7.
JJ=I1 ?
no
141 → ERFLAG
Call ERRØRZ.
yes
L=NN1-1 ?
no
142 → ERFLAG
Call ERRØRZ.
yes
N=1 and CNØ=0 ?
yes
no
CNØ=N ?
no
143 → ERFLAG
Call ERRØRZ.
yes
Call GREAD.
K1+5 → K1
N+1 → N
8.
F
no
N > K2 ?
yes
NN1=1 ?
yes
no
LTH
Σ T(K) → TSUM(I1, KK)
K=1
K2 → K
9.
SETID(J) → IPARAM1(J)
(J=1,...,K)
SETNAME → IPARAM2
Call CØMPARES.
ERFLAG=0 ?
no
144 → ERFLAG
Call ERRØRZ.
yes
10.
MATID(J) → IPARAM1(J)
(J=1,...,K)
MATNAME → IPARAM2
Call CØMPARES.
ERFLAG=0 ?
no
145 → ERFLAG
Call ERRØRZ.
yes
H

READ468 (Contd.)

11. The quantities STORFLG(I), I = 1, 2, 3, are initialized to zero for each material by the subroutine PRESET, and STORFLG(KK) is set to nonzero in subroutine TESTSTOR. Therefore TESTSTOR is entered only for each value of KK, namely, the first time that loop B is traversed.

    TESTSTOR determines whether the scattering information of type KK will fit in core.

    Subroutine PUTAWAY either stores the scattering information (obtained from the cards just processed in loop B) in the array RCDS, if the data will fit in core, or otherwise writes it onto tape number KK.


H
STØRFLG(KK) = 0 ?
no
yes
Call TESTSTØR.
Call PUTAWAY.
11.
I
MZ+1 → MZ
MZ > LI ?
no
B
yes
J
I+1 → I
I > G ?
no
A
yes
RETURN

REORDER

This subroutine is called from SCAN2; it reorders cross-section sets on the library tape according to the command *REORDER SET, set name, new position. This is done by manipulating the set directory of the new library tape.

1. SCANCARD obtains the set name and the new position from the control card and places them in FIELD(2) and FIELD(3), respectively. BCD2BIN transforms the Hollerith integer in FIELD(3) to the binary integer quantity POS2.

   If POS2 (the new position of the set) is zero or negative, the REORDER command is ignored.

2. POS2 and NUMSETS2 (the number of sets in the new library) are incremented by one to take into account the presence of the directory-of-sets file, which is the first file on the library tape. The NUMSETS2 cross-section sets in the library are files 2, 3, ..., NUMSETS on the tape; if a set is set number POS2 in the library, it is actually file number POS2 + 1 on the tape.

3. The set name [in FIELD(2)] must be the same as one of the names in the set directory of the new library, or the input card is in error.

   POS1 represents the current position of the set in the library.

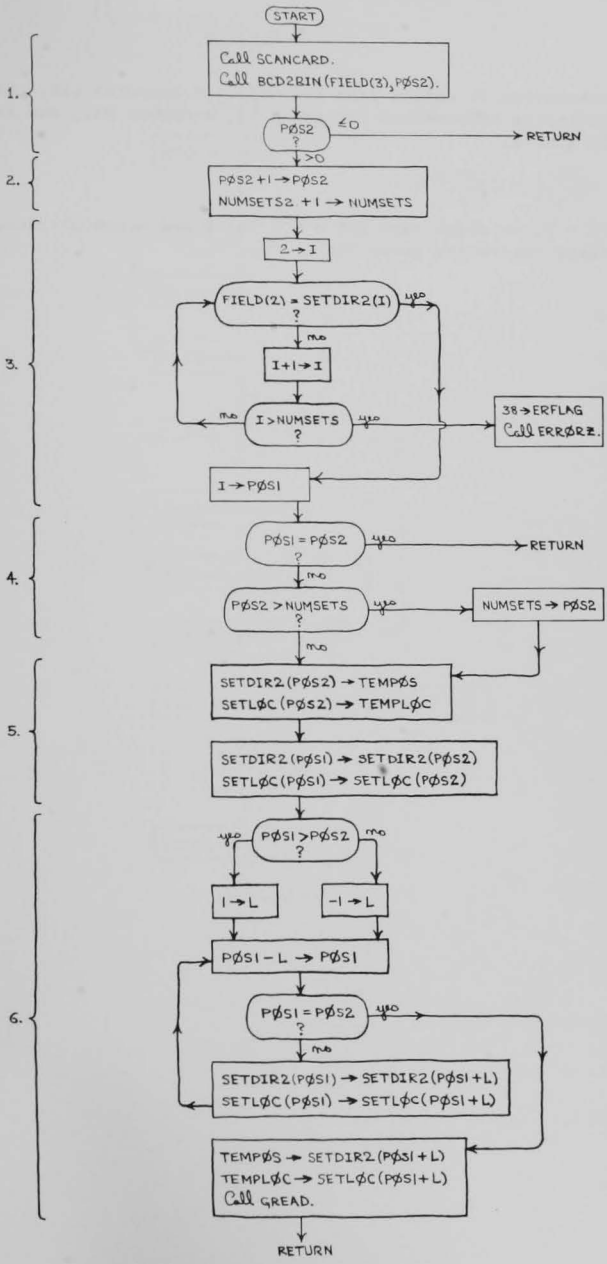
4. If POS1 = POS2, the set is already positioned as requested on the *REORDER card.

   If POS2 is greater than the number of sets in the new library, the set to be reordered is placed last on the library tape.

5. The name and location of the set currently in position POS2 are saved in TEMPOS and TEMPLOC, respectively. The name and location of the set to be repositioned are put in the new, requested location. [Note: The quantity SETLOC(I) refers to the intermediate location of a set while the new library is being constructed, and bears no relation to the set's eventual position on the new library tape.]

6. The names and locations of the sets between the new position (POS2) and present position (POS1) of the set being reordered are shifted by one away from the position POS2. This creates a gap adjacent to position POS2, in the direction of position POS1. The name and location of the set that formerly occupied position POS2 (stored temporarily in the coding of paragraph 5 above) are placed in the gap so created.

   The next input card is decoded. This is necessary in general, so that upon re-entrance to SCAN2 the next card is ready for processing.


START
1.
Call SCANCARD.
Call BCD2RIN(FIELD(3), PØS2).
PØS2 ?
≤0
RETURN
>0
2.
PØS2+1 → PØS2
NUMSETS2 + 1 → NUMSETS
3.
2 → I
FIELD(2) = SETDIR2(I) ?
yes
no
I+1 → I
I > NUMSETS ?
no
yes
38 → ERFLAG
Call ERRØRZ.
I → PØS1
4.
PØS1 = PØS2 ?
yes
RETURN
no
PØS2 > NUMSETS ?
yes
NUMSETS → PØS2
no
5.
SETDIR2(PØS2) → TEMPØS
SETLØC(PØS2) → TEMPLØC
SETDIR2(PØS1) → SETDIR2(PØS2)
SETLØC(PØS1) → SETLØC(PØS2)
6.
PØS1 > PØS2 ?
yes
no
1 → L
-1 → L
PØS1 - L → PØS1
PØS1 = PØS2 ?
yes
no
SETDIR2(PØS1) → SETDIR2(PØS1+L)
SETLØC(PØS1) → SETLØC(PØS1+L)
TEMPØS → SETDIR2(PØS1+L)
TEMPLØC → SETLØC(PØS1+L)
Call GREAD.
RETURN

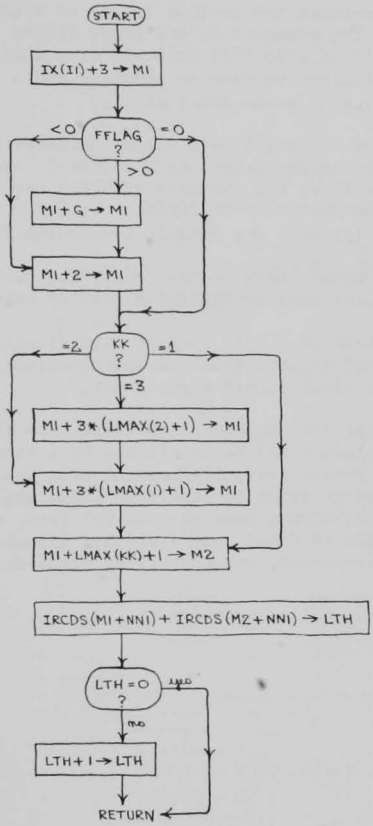
ROWLTH

This subroutine is called from READ468. It computes LTH, the number of words of scattering information for group I1, ℓ-number NN1, and scattering-type KK. In general,

$$LTH = F^{(NN1)}_{KK,I1} + G^{(NN1)}_{KK,I1} + 1$$

unless F = G = 0, in which case LTH = 0. The F and G are obtained from the group-constants record for group I1.

RØWLTH


START
IX(II)+3 → MI
FFLAG ?
<0
=0
>0
MI+G → MI
MI+2 → MI
KK ?
=2
=1
=3
MI+3*(LMAX(2)+1) → MI
MI+3*(LMAX(1)+1) → MI
MI+LMAX(KK)+1 → M2
IRCDS(MI+NNI)+IRCDS(M2+NNI) → LTH
LTH=0 ?
yes
no
LTH+1 → LTH
RETURN

SCANCARD

This routine separates the various fields on a control card, and stores them individually in the elements of the array FIELD. For example, the contents of the Ith field on a control card are stored in FIELD(I). A field is defined as those characters between an asterisk and a comma, an asterisk and a period, two commas, or a comma and a period.

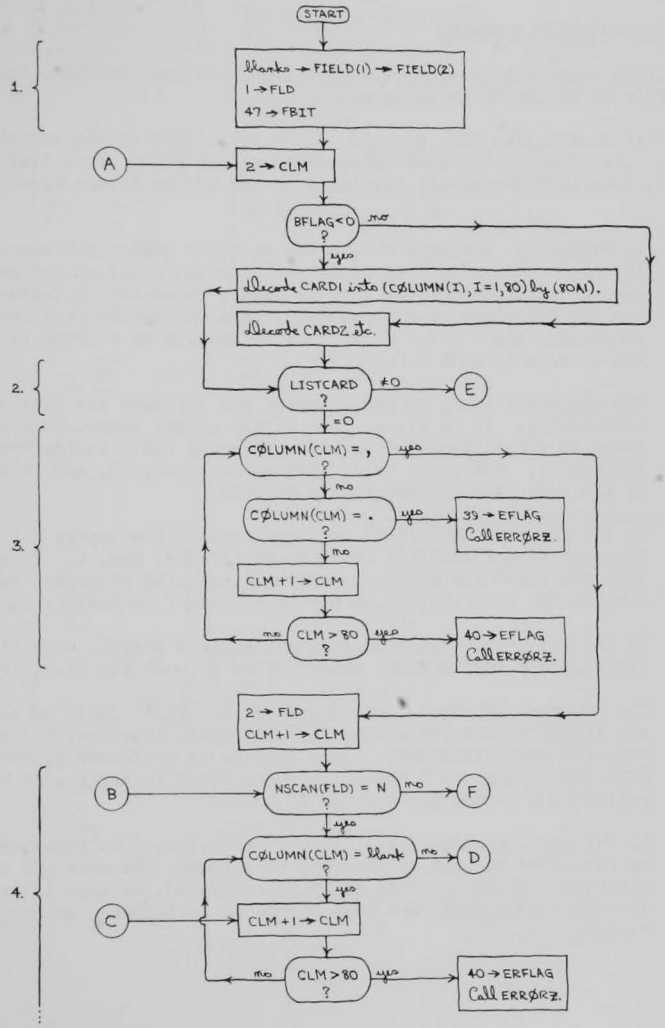
1. Column one of a control card always contains an asterisk (*) and is bypassed; processing starts with column 2. As a housekeeping measure, the first two elements of FIELD are filled with blanks. FLD gives the number of the element of FIELD to be processed. See paragraph 4 (p. 262) for details concerning FBIT.

2. If LISTCARD equals zero (equals one), the card to be processed is not (is) a list card containing a list of material names.

   The first field of a list card is significant. The first field on a nonlist control card (i.e., a modification, extraction, or separation control card) is not significant.

3. If the control card is not a list card, the first field, which starts in column 2 and is terminated by a comma, is bypassed. If a period is encountered, the control card has been incorrectly prepared, and an error is flagged. Likewise, if no terminating comma is found anywhere on the control card, an error is flagged. When the comma is found, the field and column counters (FLD and CLM) are incremented, and processing proceeds.

SCANCARD


START
blanks → FIELD(1) → FIELD(2)
1 → FLD
47 → FBIT
1.
A
2 → CLM
BFLAG < 0 ?
no
yes
decode CARD1 into (CØLUMN(I), I=1,80) by (80A1).
decode CARD2 etc.
2.
LISTCARD ?
≠0
E
=0
3.
CØLUMN(CLM) = , ?
yes
no
CØLUMN(CLM) = . ?
yes
39 → EFLAG
Call ERRØRZ.
no
CLM + 1 → CLM
no
CLM > 80 ?
yes
40 → EFLAG
Call ERRØRZ.
2 → FLD
CLM+1 → CLM
4.
B
NSCAN(FLD) = N ?
no
F
yes
CØLUMN(CLM) = blank ?
no
D
yes
C
CLM + 1 → CLM
no
CLM > 80 ?
yes
40 → ERFLAG
Call ERRØRZ.

SCANCARD (Contd.)

Notes to paragraphs 4 and 6:

A comma signifies that the end of a field has been reached, and that there are more fields to be processed.

A period signifies that the end of the last field on the card has been reached. The quantity LISTCARD is normally set to zero. If a list card is expected, then LISTCARD is set specially to one before this subroutine is called.

4. If NSCAN(FLD) contains the letter N, field number FLD contains an integer. Blanks in this field are ignored; nonblank characters are stored successively from left to right, character by character, in the FLDth element of array FIELD. This storage is performed by the subroutine BITS. If the field is terminated by a comma or a period, the storage is halted.

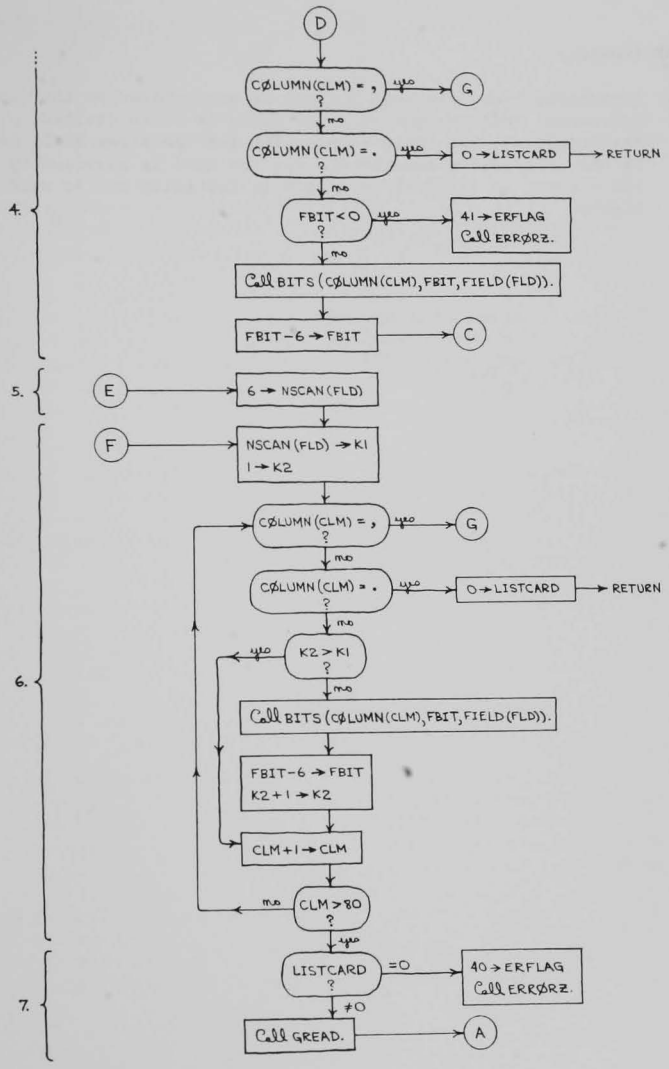
   The character being stored occupies the leftmost six bits in word COLUMN(CLM). It is stored in a string of six consecutive bits (the index of the leftmost of which is given by FBIT) in the word FIELD(FLD). For each field, FBIT is initially 47, and is decreased by six each time a character is stored.

5. If the quantity NSCAN(FLD) does not contain the letter N (for numeral) then NSCAN(FLD) contains an integer, say, I. This indicates that the FLDth field on the control card being processed contains a Hollerith quantity, which may be at most I characters in length.

   If the card is a list card (see paragraph 2 above), each field contains a material name, which may be at most six characters long.

6. The leftmost NSCAN(FLD) characters of the FLDth field on the card are stored successively from left to right, character by character, into the word FIELD(FLD). This storage is performed by subroutine BITS (see paragraph 4 above). If the field is terminated by a period or a comma, the storage is halted.

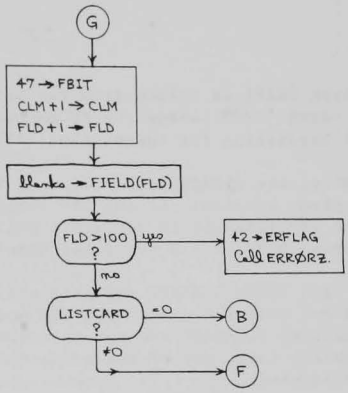
7. If the card is a list card, it may be continued on subsequent cards, so that when the end of one card is reached, the next one is read in. A nonlist control card is never so continued, so that if processing is still being performed when column 80 is reached, an error is flagged.


D
CØLUMN(CLM) = , ?
yes
G
no
CØLUMN(CLM) = . ?
yes
0 → LISTCARD
RETURN
no
FBIT < 0 ?
yes
41 → ERFLAG
Call ERRØRZ.
no
Call BITS (CØLUMN(CLM), FBIT, FIELD(FLD)).
FBIT − 6 → FBIT
C
4.
5.
E
6 → NSCAN(FLD)
F
NSCAN(FLD) → K1
1 → K2
CØLUMN(CLM) = , ?
yes
G
no
CØLUMN(CLM) = . ?
yes
0 → LISTCARD
RETURN
no
K2 > K1 ?
yes
no
Call BITS (CØLUMN(CLM), FBIT, FIELD(FLD)).
FBIT − 6 → FBIT
K2 + 1 → K2
CLM + 1 → CLM
no
CLM > 80 ?
yes
6.
LISTCARD ?
=0
40 → ERFLAG
Call ERRØRZ.
≠0
Call GREAD.
A
7.

SCANCARD (Contd.)

8. Control arrives here when a comma is encountered on the card being processed, indicating that a new field is to be started. FBIT is reinitialized; the index counter FLD for the array FIELD is advanced by one; the column counter CLM for the card is advanced by one; and the element of FIELD to be filled is initially filled with BCD blanks.

8.


G
47 → FBIT
CLM + 1 → CLM
FLD + 1 → FLD
blanks → FIELD(FLD)
FLD > 100 ?
yes
42 → ERFLAG
Call ERRØRZ.
no
LISTCARD ?
= 0
B
≠ 0
F

SCAN1

Subroutine SCAN1 is called from the main program XLIBIT when either of the control cards *FORM LIBRARY or *FORM NEW EDITION is encountered. SCAN1 controls the processing for the section (2) modification control cards.

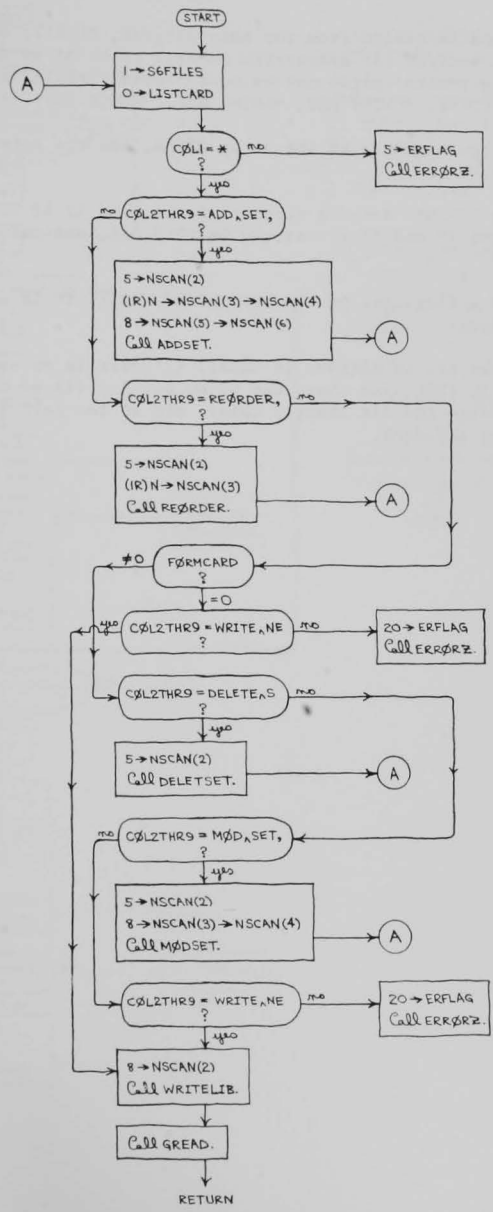
For each of the fields I, I = 2, ..., 6 on a control card, the quantity NSCAN(I) is given a value: if the Ith field contains a numeral, NSCAN(I) = (1R)N; if the Ith field is to contain a Hollerith quantity having a maximum of K characters (e.g., K = 5 for a set name), then NSCAN(I) = K.

If the card *FORM LIBRARY was encountered, there is no existing "old" library tape and only the cards *ADD SET and *REORDER may be used. For this case, the quantity FORMCARD was set to nonzero in XLIBIT. Otherwise, if there is an old library tape, any of the cards *ADD SET, *REORDER, *DELETE SET, or *MOD SET may be used.

The card *WRITE NEW EDITION terminates section (2) of an input deck.

Subroutine WRITELIB retrieves the various portions of the new library from tape units 4, 6, and 44, and writes them onto tape unit 1.

SCAN1


START
A
1 → SGFILES
0 → LISTCARD
CØLI = *
?
no
5 → ERFLAG
Call ERRØRZ.
yes
CØL2THR9 = ADD∧SET,
?
no
yes
5 → NSCAN(2)
(IR)N → NSCAN(3) → NSCAN(4)
8 → NSCAN(5) → NSCAN(6)
Call ADDSET.
A
CØL2THR9 = REØRDER,
?
no
yes
5 → NSCAN(2)
(IR)N → NSCAN(3)
Call REØRDER.
A
≠0
FØRMCARD
?
=0
yes
CØL2THR9 = WRITE∧NE
?
no
20 → ERFLAG
Call ERRØRZ.
CØL2THR9 = DELETE∧S
?
no
yes
5 → NSCAN(2)
Call DELETSET.
A
no
CØL2THR9 = MØD∧SET,
?
yes
5 → NSCAN(2)
8 → NSCAN(3) → NSCAN(4)
Call MØDSET.
A
CØL2THR9 = WRITE∧NE
?
no
20 → ERFLAG
Call ERRØRZ.
yes
8 → NSCAN(2)
Call WRITELIB.
Call GREAD.
RETURN

SCAN2

This routine is called from the main program, XLIBIT, and processes the section (1) and section (3) extraction control cards of an input deck. Any of the following control cards may be encountered: *PRINT SET DIRECTORY, *PRINT MAT DIRECTORY, *PRINT SET, *PRINT MAT, *PUNCH SET, and *PUNCH MAT.

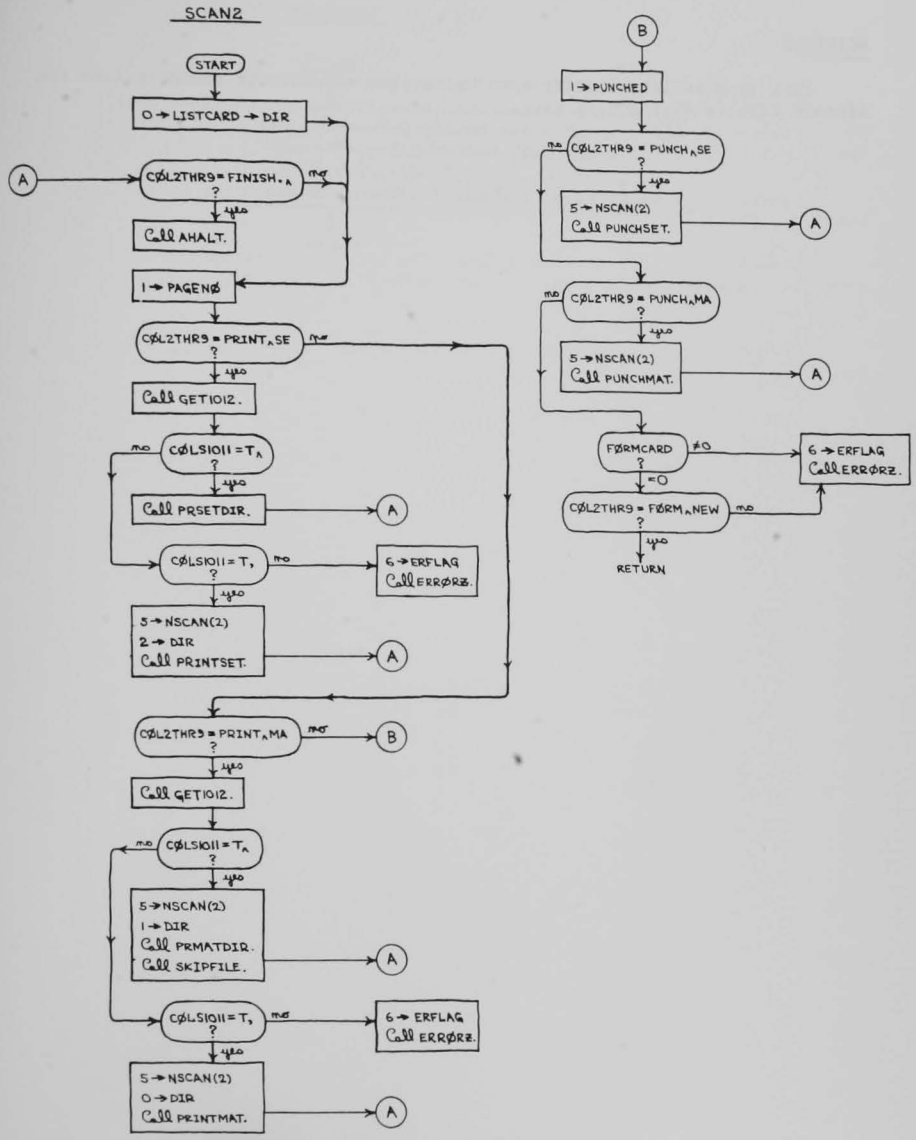
Regarding the elements of the array NSCAN, see the notes to subroutine SCAN1, p. 266.

Subroutine GET1012 decodes columns 10, 11, and 12 of the card being examined; columns 10 and 11 are stored in COLS1011, and column 12 is stored in COL12.

PUNCHED is a flag used by the subroutine AHALT, it is set to nonzero if cards are punched.

FORMCARD was set to nonzero in XLIBIT if there is no existing "old" library tape. In this case there can be no section (1) of an input deck (this section edits the old library tape), and so the card *FORM NEW EDITION may not occur in the deck.

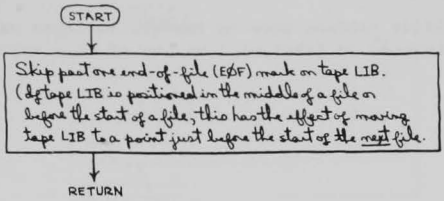
SCAN2


START
0 → LISTCARD → DIR
A
CØL2THR9 = FINISH.A ?
no
yes
Call AHALT.
1 → PAGENØ
CØL2THR9 = PRINTASE ?
no
yes
Call GET1012.
CØLS1011 = TA ?
no
yes
Call PRSETDIR.
A
CØLS1011 = T, ?
no
yes
6 → ERFLAG
Call ERRØRZ.
5 → NSCAN(2)
2 → DIR
Call PRINTSET.
A
CØL2THR9 = PRINTAMA ?
no
yes
B
Call GET1012.
CØLS1011 = TA ?
no
yes
5 → NSCAN(2)
1 → DIR
Call PRMATDIR.
Call SKIPFILE.
A
CØLS1011 = T, ?
no
yes
6 → ERFLAG
Call ERRØRZ.
5 → NSCAN(2)
0 → DIR
Call PRINTMAT.
A
B
1 → PUNCHED
CØL2THR9 = PUNCHASE ?
no
yes
5 → NSCAN(2)
Call PUNCHSET.
A
CØL2THR9 = PUNCHAMA ?
no
yes
5 → NSCAN(2)
Call PUNCHMAT.
A
FØRMCARD ?
≠0
=0
6 → ERFLAG
Call ERRØRZ.
CØL2THR9 = FØRMANEW ?
no
yes
RETURN

SKIPFILE

This is a utility routine used by various subroutines to manipulate the library tape (either old or new).

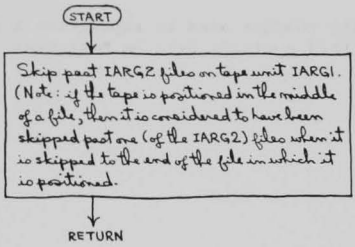
SKIPFILE


START
Skip past one end-of-file (EØF) mark on tape LIB.
(If tape LIB is positioned in the middle of a file or
before the start of a file, this has the effect of moving
tape LIB to a point just before the start of the next file.
RETURN

SKPFILES(IARG1,IARG2)

This is a utility routine used to reposition tapes used by the program. The tape unit designated by IARG1 will be one of tape units 4, 6, or 44.

SKPFILES(IARG1, IARG2)



SPACEBY(TAPE)

This is a utility routine used to reposition a tape forward a certain number of records within a single file on that tape.

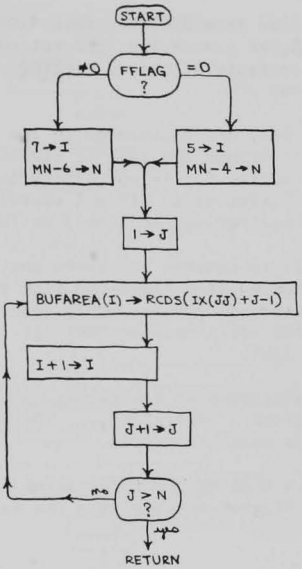
SPACEBY (TAPE)

STORGRP

This routine is called by subroutine PNCH1MAT. It transfers the as-yet-unpunched information in the group-constants record for group JJ from a buffer area into the appropriate cells of the array RCDS.

The quantities MN and IX(JJ) are set in subroutine FIX2ADD.

STØRGRP


START
FFLAG ?
≠0
=0
7 → I
MN-6 → N
5 → I
MN-4 → N
1 → J
BUFAREA(I) → RCDS(IX(JJ)+J-1)
I+1 → I
J+1 → J
J > N ?
no
yes
RETURN

STORSCAT

This routine is called from PNCH1MAT when STORFLG(KK) is negative (signifying that the type-KK scattering information can be contained in core). STORSCAT transfers the contents of the scattering-data record just read from a buffer area to the array RCDS.

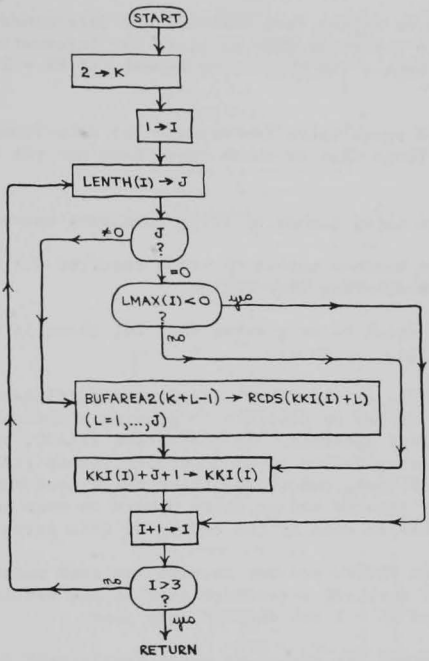
The quantities LENTH(I) are calculated in the routine RCDLTH, which is called from PNCH1MAT just before the call to STORSCAT. LENTH(I) gives the number of words of information for I-type scattering in the scattering-data record currently in the buffer area. (I = 1 corresponds to elastic scattering, I = 2 to inelastic scattering, and I = 3 to (n,2n) scattering.)

Only if J = LENTH(I) is nonzero are there any data to be stored. If J is zero but LMAX(I) is nonnegative (implying that there is I-type scattering information in some scattering-data records), then a single cell in RCDS is left for this record; this cell contains LENTH(I) (see also paragraph 7 of subroutine PNCH1MAT, p. 156).

The indices for the storage of scattering information in RCDS, KKI(I), are initialized in paragraph 3 of FIX2ADD, p. 130 and augmented here, as scattering information is read and stored.

The structure within RCDS of the information from the scattering-data records is shown on the diagram accompanying the notes to subroutine PUNCH468, p. 218.

STØRSCAT


START
2 → K
1 → I
LENTH(I) → J
J ?
≠0
=0
LMAX(I) < 0 ?
ja
nei
BUFAREA2(K+L-1) → RCDS(KKI(I)+L)
(L=1,...,J)
KKI(I)+J+1 → KKI(I)
I+1 → I
I > 3 ?
nei
ja
RETURN

TESTSTOR

This routine is called from READ468. It determines whether there is sufficient room in the array RCDS to store the information for scattering type KK from the type-4 (if KK = 1) or type-6 (if KK = 2) or type-8 (if KK = 3) cards.

1. L7 is the space taken for storage for data from the 4-cards and 6-cards (note that if these cards have not yet been read, L7 is zero).

   L5 is the total number of words remaining unused in the array RCDS.

   L4 is the maximum number of words required for storage of the type-KK scattering information.

2. If L5 is equal to or greater than L4, there is enough room in RCDS for storage.

3. If there is, at the first estimate, insufficient room in RCDS for storage, it may be possible to provide additional words in RCDS to make storage possible. In subroutine FIXADD, space was allowed for the largest possible group-constants record (LAMDA equal to 9 and FELAG positive), and if all these cells were not used, the group-constants records can be moved upward to make room available in RCDS. This is done by the coding of this paragraph.

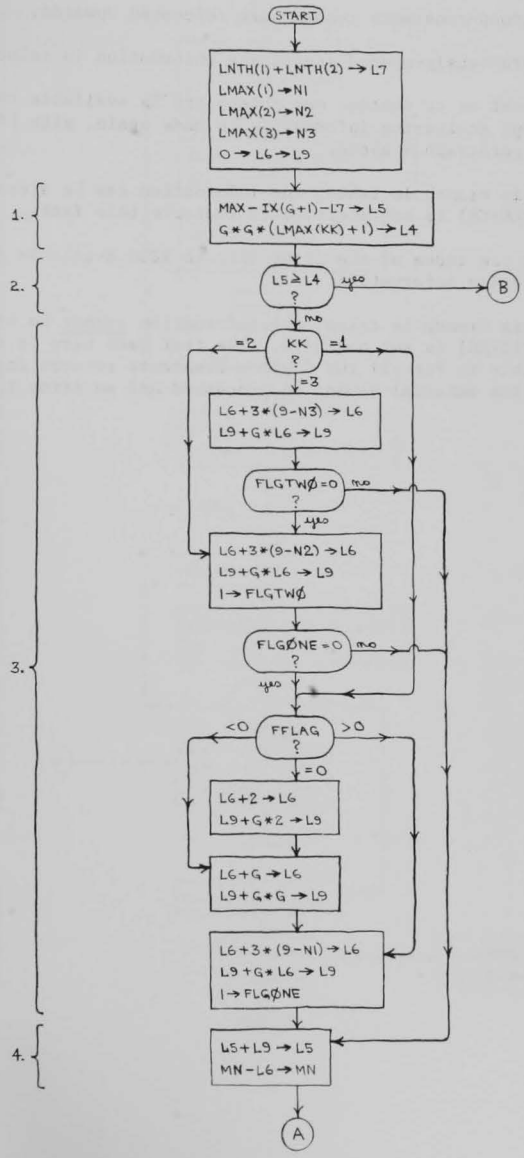
   FLGONE and FLGTWO are set to zero for each material in PRESET. Note that they are reset here so that the sections of coding for KK = 1 and KK = 2 are entered only once.

   L6 represents the number of words "reclaimed" in one group-constants record.

   L9 represents the total number of words so reclaimed for other use from all G group-constants records.

4. MN is the size of a group-constants record; it is decreased appropriately. L5, the number of words available for storage in RCDS, is increased by the total number of words reclaimed.

TESTSTØR


START
LNTH(1)+LNTH(2) → L7
LMAX(1) → N1
LMAX(2) → N2
LMAX(3) → N3
0 → L6 → L9
1.
MAX − IX(G+1) − L7 → L5
G*G*(LMAX(KK)+1) → L4
2.
L5 ≥ L4 ?
yes
B
no
KK ?
=2
=1
=3
L6+3*(9−N3) → L6
L9+G*L6 → L9
FLGTWØ=0 ?
no
yes
L6+3*(9−N2) → L6
L9+G*L6 → L9
1 → FLGTWØ
3.
FLGØNE=0 ?
no
yes
FFLAG ?
<0
>0
=0
L6+2 → L6
L9+G*2 → L9
L6+G → L6
L9+G*G → L9
L6+3*(9−N1) → L6
L9+G*L6 → L9
1 → FLGØNE
4.
L5+L9 → L5
MN−L6 → MN
A

TESTSTOR (Contd.)

5. The group-constants records are relocated upwards.

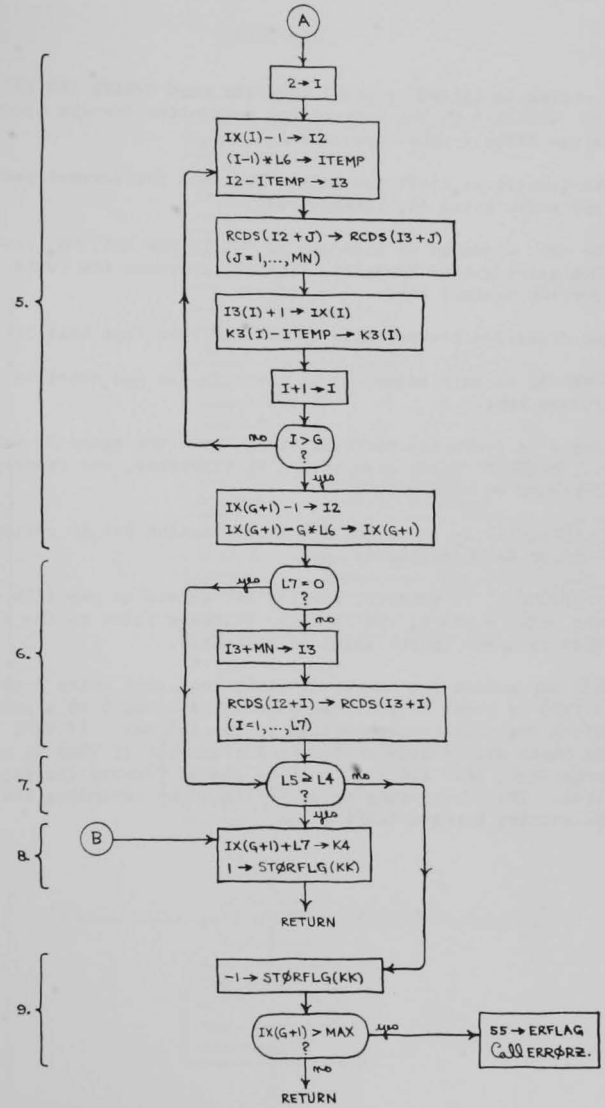
6. Any previously-stored scattering information is relocated upwards.

7. The test as to whether enough storage is available to store the KK-type scattering information is made again, with L5 augmented from paragraph 4 above.

8. If this branch is taken, the information can be stored in core; STORFLG(KK) is set positive to indicate this fact.

   K4 is the index of the first cell in RCDS available for storage of scattering information.

9. If this branch is taken, the information cannot be stored in core; STORFLG(KK) is set negative. The test made here is to see if it is possible to fit all the G group-constants records into RCDS. If not, the material cannot be processed and an error is flagged.


A
2 → I
IX(I) − 1 → I2
(I−1) * L6 → ITEMP
I2 − ITEMP → I3
RCDS(I2 + J) → RCDS(I3 + J)
(J = 1,..., MN)
I3(I) + 1 → IX(I)
K3(I) − ITEMP → K3(I)
I + 1 → I
I > G ?
no
yes
IX(G+1) − 1 → I2
IX(G+1) − G * L6 → IX(G+1)
L7 = 0 ?
yes
no
I3 + MN → I3
RCDS(I2+I) → RCDS(I3+I)
(I = 1,..., L7)
L5 ≥ L4 ?
no
yes
B
IX(G+1) + L7 → K4
1 → STØRFLG(KK)
RETURN
−1 → STØRFLG(KK)
IX(G+1) > MAX ?
yes
no
55 → ERFLAG
Call ERRØRZ.
RETURN
5.
6.
7.
8.
9.

WRITELIB

This routine is called by SCAN1 when the card *WRITE NEW EDITION is encountered. WRITELIB is the controlling subroutine for the process of writing the new library onto tape unit 1.

1. The quantities LAST6 and LAST44 indicate the current positions of tape units 6 and 44, respectively.

2. The card examined by SCANCARD is *WRITE NEW EDITION, date of run. Upon exiting from SCANCARD, FIELD(2) contains the "date of run" from the control card.

3. The directory-of-sets file is written onto tape unit 1.

4. FORMCARD is zero (nonzero) if there is (is no) existing "old" library tape.

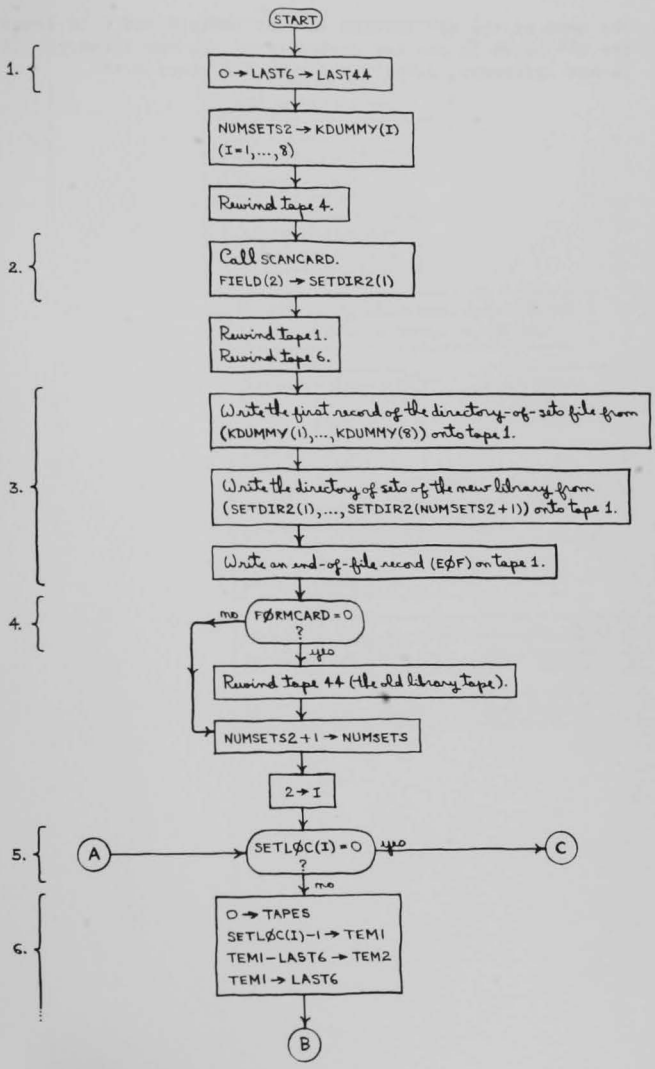
   Loop A is traversed NUMSETS2 times, over the range of values I = 2, ..., NUMSETS. Each time loop A is traversed, one cross-section set is placed on tape unit 1.

5. If SETLOC(I) is zero, the Ith cross-section set is currently to be found on the old library tape.

   If SETLOC(I) is nonzero, the Ith set exists as one file each on tape units 4 and 6; the location of these files on the respective tapes is given by the value of SETLOC(I).

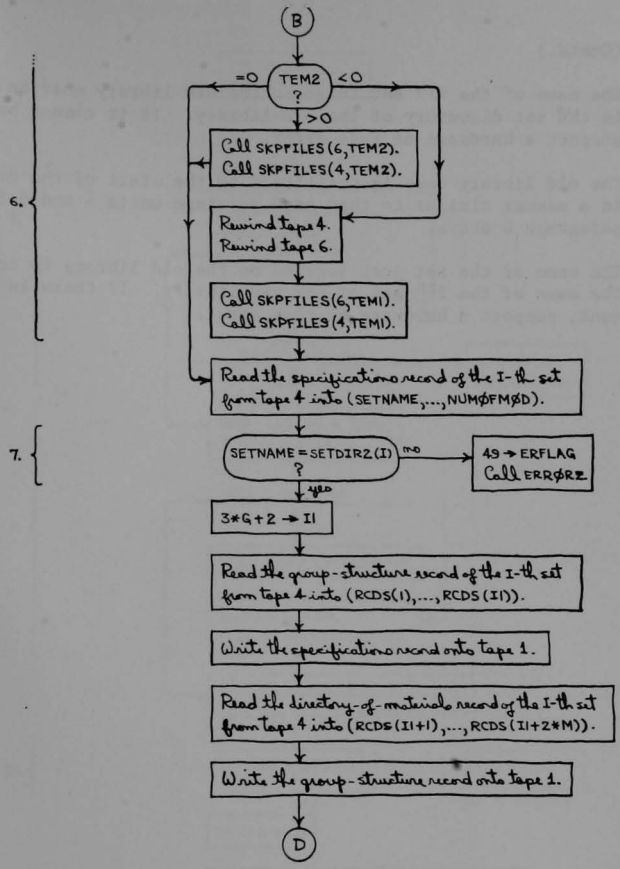
6. TEM2 represents the number of files that tape units 4 and 6 must be shifted in order to position tape units 4 and 6 to a point just before the files corresponding to the Ith set. If TEM2 is zero, the tapes are already correctly positioned; if TEM2 is positive (negative), then the tape must be spaced forward (backspaced) TEM2 files. The backspacing is accomplished by rewinding the tapes and spacing forward TEMP1 files.

WRITELIB


START
1.
0 → LAST6 → LAST44
NUMSETS2 → KDUMMY(I)
(I=1,...,8)
Rewind tape 4.
2.
Call SCANCARD.
FIELD(2) → SETDIR2(1)
Rewind tape 1.
Rewind tape 6.
3.
Write the first record of the directory-of-sets file from
(KDUMMY(1),...,KDUMMY(8)) onto tape 1.
Write the directory of sets of the new library from
(SETDIR2(1),...,SETDIR2(NUMSETS2+1)) onto tape 1.
Write an end-of-file record (EØF) on tape 1.
4.
FØRMCARD = 0
?
no
yes
Rewind tape 44 (the old library tape).
NUMSETS2+1 → NUMSETS
2 → I
5.
A
SETLØC(I) = 0
?
yes
C
no
6.
0 → TAPES
SETLØC(I)−1 → TEM1
TEM1 − LAST6 → TEM2
TEM1 → LAST6
B

WRITELIB (Contd.)

7. The name of the set located on tape units 4 and 6 is compared to the Ith entry in the set directory of the new library. If there is not agreement, suspect a hardware or tape error.

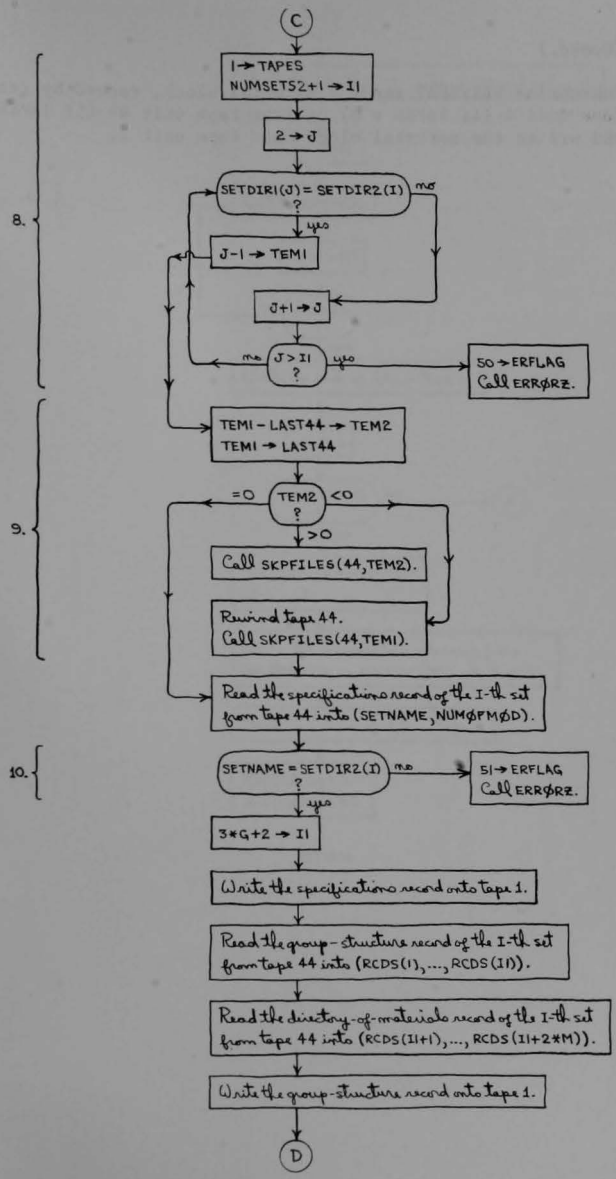

B
TEM2 ?
=0
<0
>0
Call SKPFILES (6,TEM2).
Call SKPFILES(4,TEM2).
6.
Rewind tape 4.
Rewind tape 6.
Call SKPFILES(6,TEM1).
Call SKPFILES (4,TEM1).
Read the specifications record of the I-th set from tape 4 into (SETNAME,...,NUMØFMØD).
7.
SETNAME = SETDIR2(I) ?
no
yes
49 → ERFLAG
Call ERRØR2
3*G+2 → I1
Read the group-structure record of the I-th set from tape 4 into (RCDS(1),...,RCDS(I1)).
Write the specifications record onto tape 1.
Read the directory-of-materials record of the I-th set from tape 4 into (RCDS(I1+1),...,RCDS(I1+2*M)).
Write the group-structure record onto tape 1.
D

WRITELIB (Contd.)

8. The name of the Ith set to go on the new library must be located in the set directory of the old library. If it cannot be found, suspect a hardware or tape error.

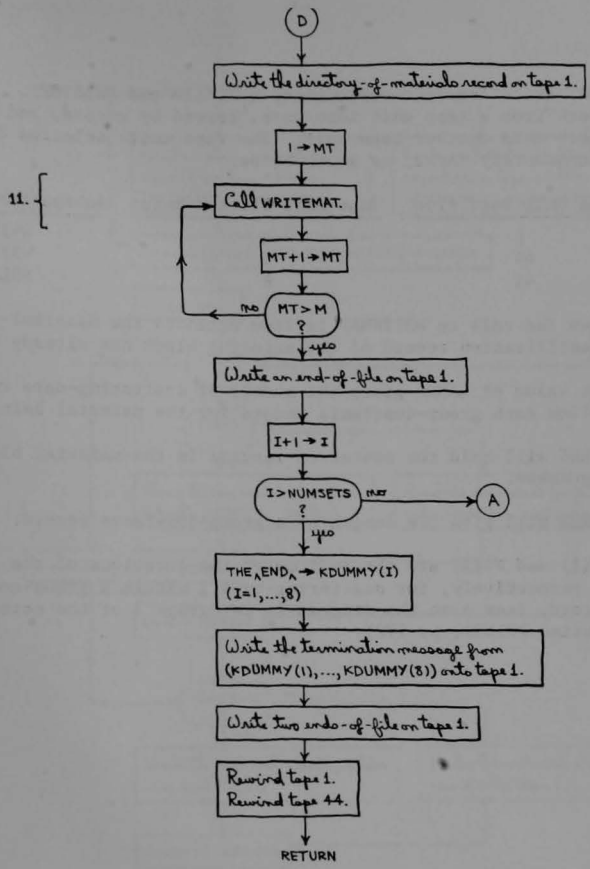
9. The old library tape is positioned to the start of the desired set, in a manner similar to that used for tape units 4 and 6 in paragraph 6 above.

10. The name of the set just located on the old library is compared to the name of the Ith set of the new library. If there is no agreement, suspect a hardware or tape error.


C
1 → TAPES
NUMSETS2+1 → I1
2 → J
8.
SETDIR1(J) = SETDIR2(I) ?
no
yes
J-1 → TEM1
J+1 → J
J > I1 ?
no
yes
50 → ERFLAG
Call ERRØRZ.
9.
TEM1 - LAST44 → TEM2
TEM1 → LAST44
TEM2 ?
=0
<0
>0
Call SKPFILES(44,TEM2).
Rewind tape 44.
Call SKPFILES(44,TEM1).
Read the specification record of the I-th set from tape 44 into (SETNAME, NUMØFMØD).
10.
SETNAME = SETDIR2(I) ?
no
51 → ERFLAG
Call ERRØRZ.
yes
3*G+2 → I1
Write the specification record onto tape 1.
Read the group-structure record of the I-th set from tape 44 into (RCDS(1), ..., RCDS(I1)).
Read the directory-of-materials record of the I-th set from tape 44 into (RCDS(I1+1), ..., RCDS(I1+2*M)).
Write the group-structure record onto tape 1.
D

WRITELIB (Contd.)

11. Subroutine WRITEMAT reads a material block, record by record, from tape unit 6 (if TAPES = 0) or from tape unit 44 (if TAPES = 1), and writes the material block onto tape unit 1.


D
Write the directory-of-materials record on tape 1.
1 → MT
11.
Call WRITEMAT.
MT+1 → MT
MT > M ?
no
yes
Write an end-of-file on tape 1.
I+1 → I
I > NUMSETS ?
no
A
yes
THE_END_ → KDUMMY(I)
(I=1,...,8)
Write the termination message from
(KDUMMY(1),..., KDUMMY(8)) onto tape 1.
Write two end-of-file on tape 1.
Rewind tape 1.
Rewind tape 44.
RETURN

WRITEMAT

WRITEMAT is called from subroutines WRITELIB and DELETSET. It copies a material block from a tape unit into core, record by record, and writes the material block onto another tape unit. The tape units selected depend on the value of the quantity TAPES, as shown below.

TAPES	Tape Unit Read from	Tape Unit Written onto	Subroutine Called from
0	6	1	WRITELIB
1	44	1	WRITELIB
2	44	6	DELETSET

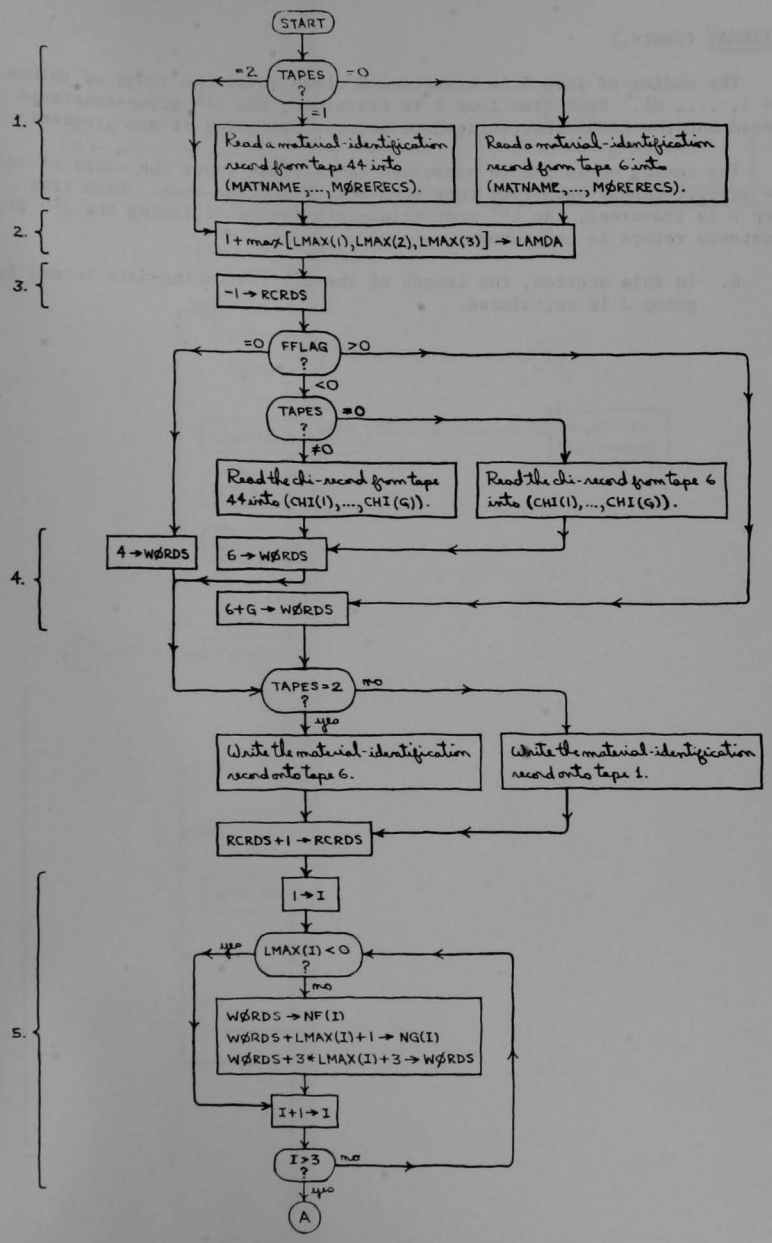
1. When the call to WRITEMAT is from DELETSET the material-identification record of the material block has already been read in.

2. The value of LAMDA gives the number of scattering-data records that follow each group-constants record for the material being processed.

3. RCRDS will hold the number of records in the material block being processed.

4. WORDS will give the length of a group-constants record.

5. NF(I) and NG(I) are the indices of the locations of the first F and G, respectively, for scattering-type I within a group-constants record, (see also the diagram in paragraph 1 of the notes to subroutine PRINT1, p. 172).

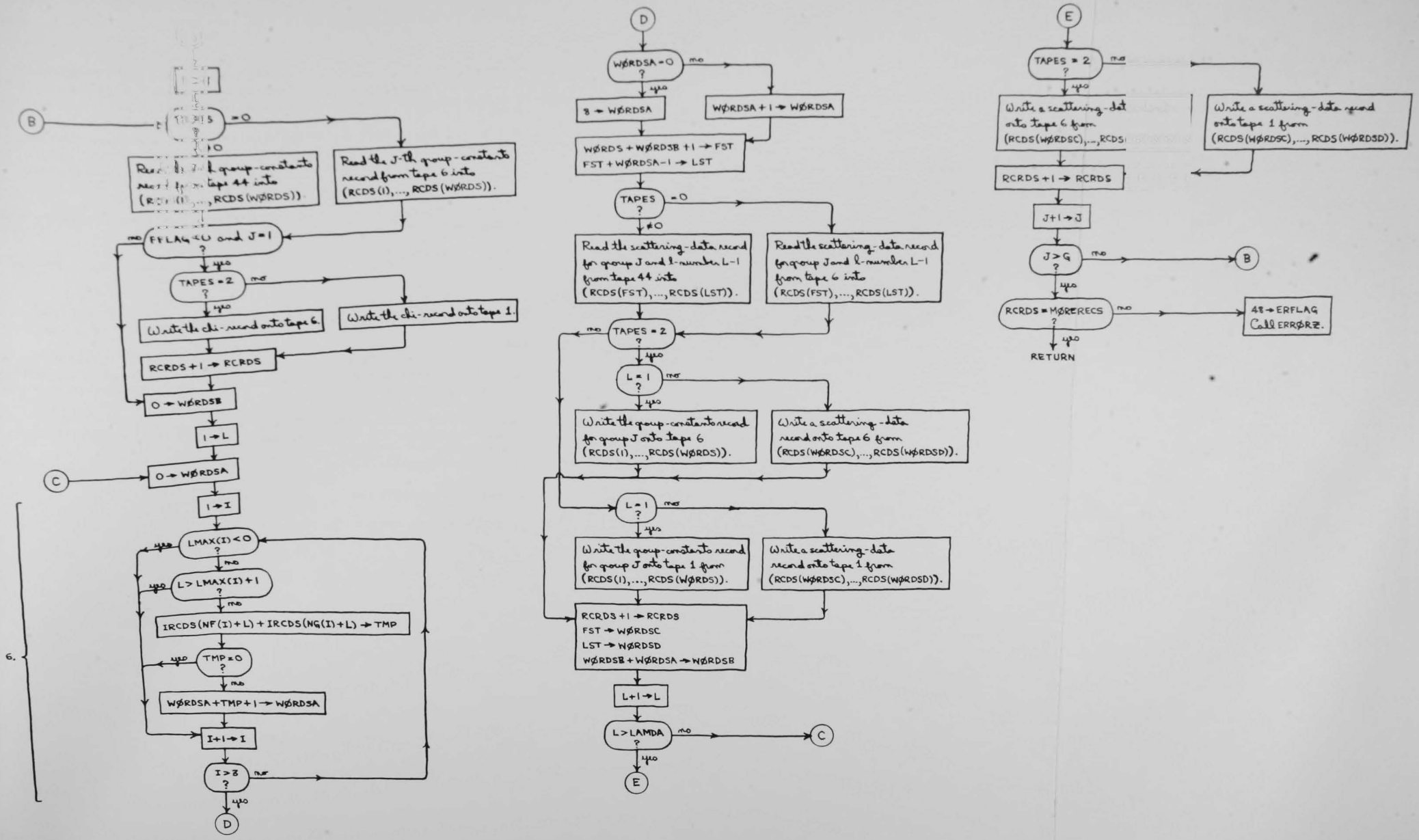

START
TAPES ?
=2
=0
=1
1.
Read a material-identification record from tape 44 into (MATNAME,..., MØREREC5).
Read a material-identification record from tape 6 into (MATNAME,..., MØREREC5).
2.
1 + max [LMAX(1), LMAX(2), LMAX(3)] → LAMDA
3.
-1 → RCRDS
FFLAG ?
=0
>0
<0
TAPES ?
=0
≠0
Read the chi-record from tape 44 into (CHI(1),..., CHI(G)).
Read the chi-record from tape 6 into (CHI(1),..., CHI(G)).
4.
4 → WØRDS
6 → WØRDS
6+G → WØRDS
TAPES=2 ?
no
yes
Write the material-identification record onto tape 6.
Write the material-identification record onto tape 1.
RCRDS+1 → RCRDS
5.
1 → I
LMAX(I) < 0 ?
yes
no
WØRDS → NF(I)
WØRDS + LMAX(I) + 1 → NG(I)
WØRDS + 3*LMAX(I) + 3 → WØRDS
I+1 → I
I > 3 ?
no
yes
A

WRITEMAT (Contd.)

The coding of loop B is traversed G times (over the range of values J = 1, ..., G). Each time loop B is traversed, the Jth group-constants record and the LAMDA scattering-data records following it are processed.

The coding of loop C is traversed LAMDA times (over the range of values L = 1, ..., LAMDA) each time that loop B is traversed once. Each time loop C is traversed, the Lth scattering-data record following the Jth group-constants record is processed.

6. In this section, the length of the Lth scattering-data record for group J is calculated.


B
=0
Read the J-th group-constants record from tape 6 into (RCDS(1),..., RCDS(WØRDS)).
FFLAG = 0 and J = 1
no
yes
TAPES = 2
no
yes
Write the chi-record onto tape 6.
Write the chi-record onto tape 1.
RCRDS + 1 → RCRDS
0 → WØRDSB
1 → L
C
0 → WØRDSA
1 → I
LMAX(I) < 0
L > LMAX(I) + 1
IRCDS(NF(I) + L) + IRCDS(NG(I) + L) → TMP
TMP = 0
WØRDSA + TMP + 1 → WØRDSA
I + 1 → I
I > 8
D
6.
D
WØRDSA = 0
8 → WØRDSA
WØRDSA + 1 → WØRDSA
WØRDS + WØRDSB + 1 → FST
FST + WØRDSA − 1 → LST
TAPES = 0
≠0
Read the scattering-data record for group J and l-number L−1 from tape 44 into (RCDS(FST),..., RCDS(LST)).
Read the scattering-data record for group J and l-number L−1 from tape 6 into (RCDS(FST),..., RCDS(LST)).
TAPES = 2
L = 1
Write the group-constants record for group J onto tape 6 (RCDS(1),..., RCDS(WØRDS)).
Write a scattering-data record onto tape 6 from (RCDS(WØRDSC),..., RCDS(WØRDSD)).
L = 1
Write the group-constants record for group J onto tape 1 from (RCDS(1),..., RCDS(WØRDS)).
Write a scattering-data record onto tape 1 from (RCDS(WØRDSC),..., RCDS(WØRDSD)).
RCRDS + 1 → RCRDS
FST → WØRDSC
LST → WØRDSD
WØRDSB + WØRDSA → WØRDSB
L + 1 → L
L > LAMDA
C
E
E
TAPES = 2
Write a scattering-data record onto tape 6 from (RCDS(WØRDSC),..., RCDS
Write a scattering-data record onto tape 1 from (RCDS(WØRDSC),..., RCDS(WØRDSD)).
RCRDS + 1 → RCRDS
J + 1 → J
J > G
B
RCRDS = MØRERECS
48 → ERFLAG
Call ERRØRZ.
RETURN

WRITSCRH

Subroutine WRITSCRH is called by ADDAMAT. It places the material block for a newly-added or a modified material onto tape unit 6 (from where it will be retrieved when the new library tape is formed).

1. LMAX(I) nonnegative (negative) implies that the data of scattering-type I exist (do not exist) for this material. STORFLG(I) nonnegative (negative) implies that the data for scattering-type I are contained in core (on tape unit I1).

   I = 1 corresponds to elastic scattering, I = 2 to inelastic scattering, and I = 3 to (n,2n) scattering.

2. NN1 is an indicator used by subroutine CALCDTRX. See paragraph 4 below.

   WFLAG is used in the coding of paragraph 10 (p. 300) to read a scattering-data record into the top or bottom half of a buffer area.

3. KK1 was given the value 1 + max [LMAX(1), LMAX(2), LMAX(3)] in subroutine ADDAMAT. The quantity RC therefore represents the number of scattering-data records following all group-constants records of the material block.

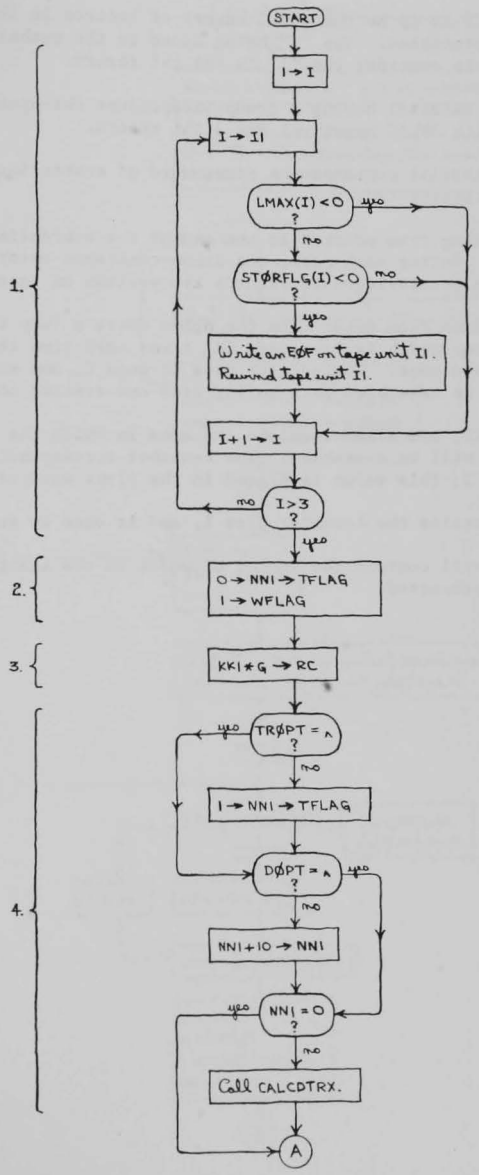
4. CALCDTRX calculates either or both of the diffusion coefficients D_j and the transport cross section σ_{tr}.

   If NN1 = 1, calculate σ_{tr}.

   If NN1 = 10, calculate D_j.

   If NN1 = 11, calculate σ_{tr} and D_j.

WRITSCRH


START
1 → I
I → I1
LMAX(I) < 0 ?
yes
no
STØRFLG(I) < 0 ?
no
yes
Write an EØF on tape unit I1.
Rewind tape unit I1.
I+1 → I
I > 3 ?
no
yes
0 → NN1 → TFLAG
1 → WFLAG
KK1 * G → RC
TRØPT = ^ ?
yes
no
1 → NN1 → TFLAG
DØPT = ^ ?
yes
no
NN1 + 10 → NN1
NN1 = 0 ?
yes
no
Call CALCDTRX.
A
1.
2.
3.
4.

5. MORERECS is to be the total number of records in the material block being processed. The "G" being added is the number of group-constants records; the "1" is the chi record.

6. Only a material having a group-independent chi-spectrum (and therefore with FFLAG negative) has a chi record.

7. RCDS(KADD(I)) contains the first word of scattering information of scattering-type I.

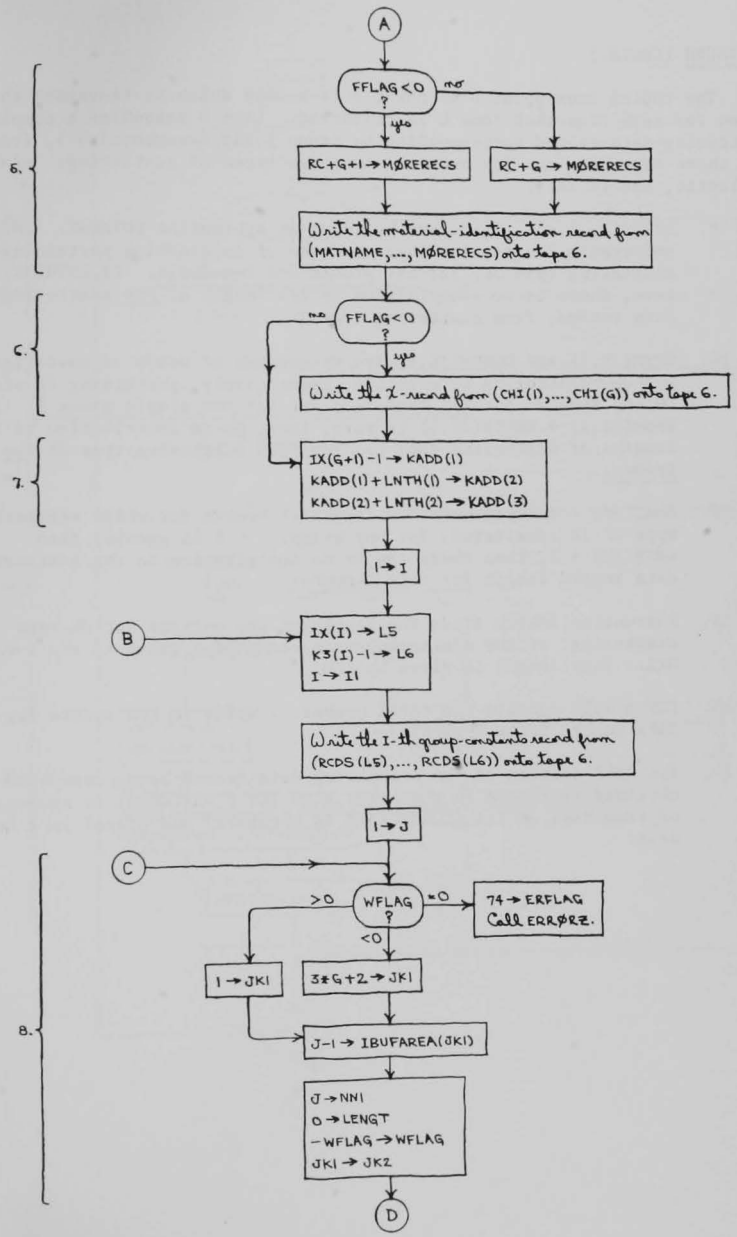
   The coding from point B to the end of the subroutine is traversed G times. During each pass, one group-constants record and its accompanying scattering-data records are written on tape unit 6.

   The coding from point C to the point where a jump to point C occurs is a loop which is traversed KK1 times each time the loop B is traversed once. During each pass through C, one scattering-data record is assembled in a buffer area and written on tape unit 6.

8. JK1, JK2, and WFLAG regulate the area in which the scattering-data record will be assembled. The ℓ-number corresponding to this record is J - 1; this value is placed in the first word of the record.

   NN1 contains the ℓ-number plus 1, and is used by subroutine ROWLTH.

   LENGT will contain the number of words in the scattering-data record being processed.


A
FFLAG < 0 ?
no
yes
RC + G + 1 → MØRERECS
RC + G → MØRERECS
6.
Write the material-identification record from (MATNAME, ..., MØRERECS) onto tape 6.
FFLAG < 0 ?
no
yes
5.
Write the χ-record from (CHI(1), ..., CHI(G)) onto tape 6.
IX(G+1) − 1 → KADD(1)
KADD(1) + LNTH(1) → KADD(2)
KADD(2) + LNTH(2) → KADD(3)
7.
1 → I
B
IX(I) → L5
K3(I) − 1 → L6
I → I1
Write the I-th group-constants record from (RCDS(L5), ..., RCDS(L6)) onto tape 6.
1 → J
C
WFLAG ?
> 0
= 0
< 0
74 → ERFLAG
Call ERRØRZ.
1 → JK1
3 * G + 2 → JK1
J − 1 → IBUFAREA(JK1)
8.
J → NN1
0 → LENGT
− WFLAG → WFLAG
JK1 → JK2
D

The coding from point D to point E is a loop which is traversed three times for each time that loop C is traversed. Loop D assembles a complete scattering-data record corresponding to group I and ℓ-number J - 1, from its three sections [one for each of the three types of scattering: elastic, inelastic, and (n,2n)].

9. LNTH(I), I = 1, 2, 3, are built up in subroutine PUTAWAY. LNTH(JK) represents the total number of words of information pertaining to scattering type JK, for all groups and ℓ-numbers. If LNTH(JK) is zero, there is no contribution to the length of any scattering-data record, from scattering type JK.

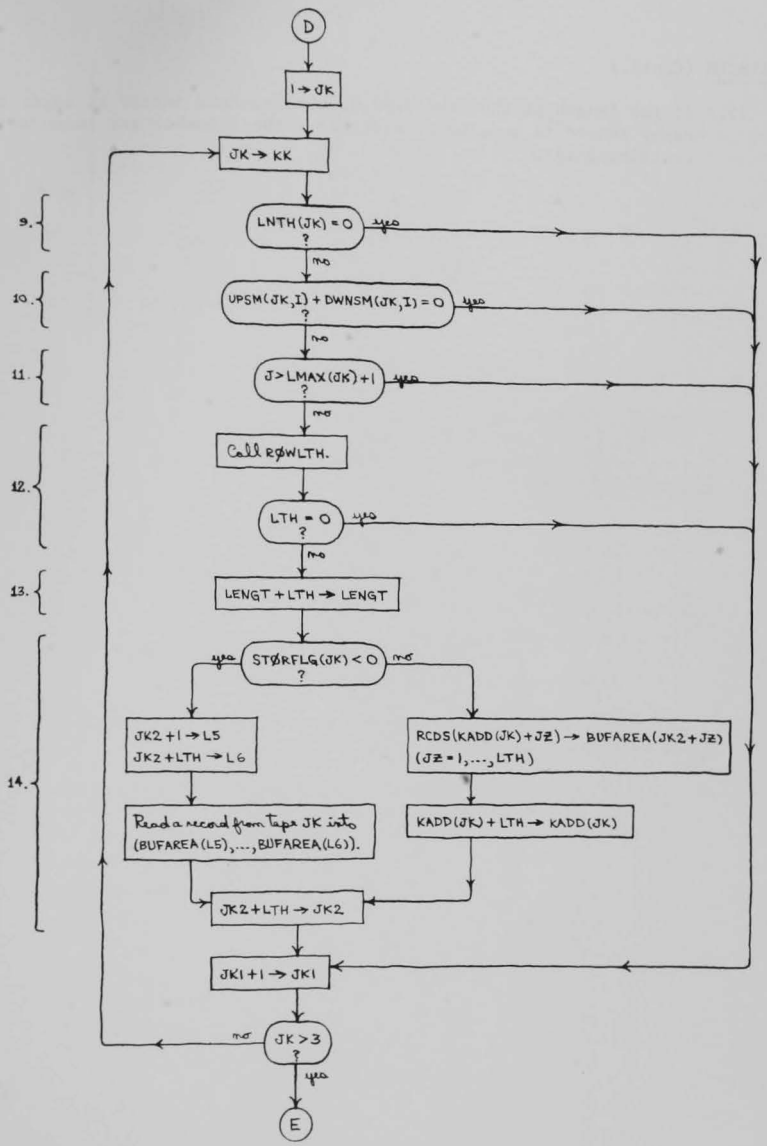
10. UPSM(JK,I) and DWNSM(JK,I) are the number of words of upscattering and downscattering information, respectively, pertaining to scattering type JK for all ℓ-numbers but for the single group I. If UPSM(JK,I) + DWNSM(JK,I) is zero, there is no contribution to the lengths of scattering-data records from scattering type JK for group I.

11. LMAX(JK) + 1 represents the largest ℓ-number for which scattering of type JK is considered, for any group. If L is greater than LMAX(JK) + 1, then there can be no contribution to the scattering-data record length for this ℓ-number.

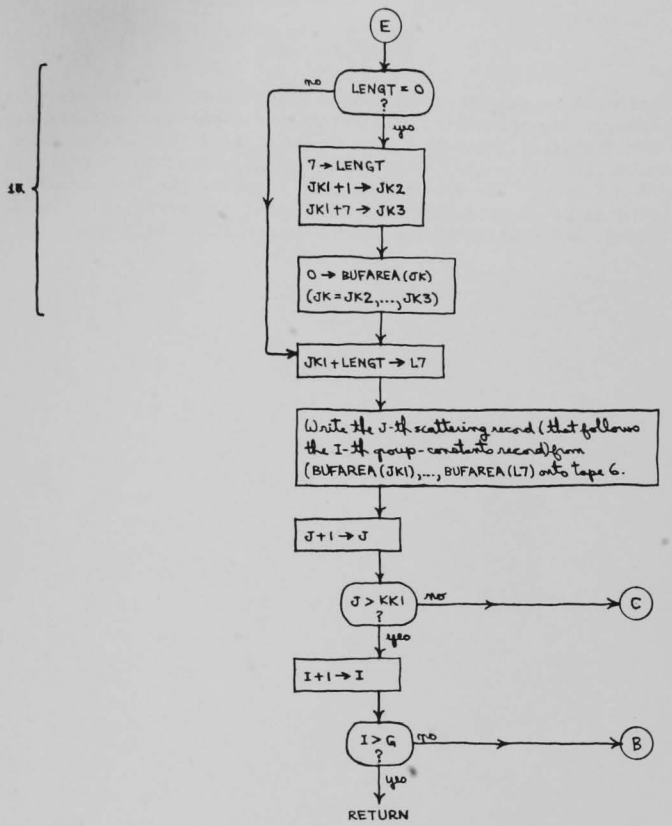
12. Subroutine ROWLTH finds the length of the portion for JK-type scattering, of the scattering-data record for group I1 and ℓ-number NN1. This length is given by LTH.

13. LENGT will contain the total number of words in the scattering-data record being assembled.

14. The JK^{th} portion of the scattering-data record being assembled is obtained from core in the array RCDS [if STORFLG(JK) is nonnegative] or from tape JK [if STORFLG(JK) is negative] and placed in a buffer area.


D
1 → JK
JK → KK
9.
LNTH(JK) = 0 ?
yes
no
10.
UPSM(JK,I) + DWNSM(JK,I) = 0 ?
yes
no
11.
J > LMAX(JK) + 1 ?
yes
no
12.
Call RØWLTH.
LTH = 0 ?
yes
no
13.
LENGT + LTH → LENGT
14.
STØRFLG(JK) < 0 ?
yes
no
JK2 + 1 → L5
JK2 + LTH → L6
RCDS(KADD(JK) + JZ) → BUFAREA(JK2 + JZ)
(JZ = 1, ..., LTH)
Read a record from Tape JK into
(BUFAREA(L5), ..., BUFAREA(L6)).
KADD(JK) + LTH → KADD(JK)
JK2 + LTH → JK2
JK1 + 1 → JK1
JK > 3 ?
no
yes
E

15. If the length of the assembled scattering-data record is zero, a dummy record is made up consisting of the ℓ-number and seven words containing zero.


E
LENGT = 0 ?
no
yes
7 → LENGT
JK1 + 1 → JK2
JK1 + 7 → JK3
0 → BUFAREA (JK)
(JK = JK2, ..., JK3)
JK1 + LENGT → L7
Write the J-th scattering record (that follows the I-th group-constants record) from (BUFAREA (JK1), ..., BUFAREA (L7) onto tape 6.
J + 1 → J
J > KK1 ?
no
C
yes
I + 1 → I
I > G ?
no
B
yes
RETURN

ACKNOWLEDGMENTS

Section II of this report (Cross-section Nomenclature) was written by Robert Avery, Margaret Butler, and Harold Greenspan and appeared originally as ANL-AMD Technical Memorandum 92. Ed Mueller of the Control Data Corporation worked with the author at Argonne National Laboratory on the preparation of the XLIBIT code. Gerry Duffy and John Zapatka, the two remaining members of the project at Argonne, assisted throughout the preparation of XLIBIT by their use of, and constructive comments concerning, the code.

ARGONNE NATIONAL LAB WEST
3 4444 00005905 5