203
10/3/77
LA-6950-MS

Informal Report

1455
UC-32

Issued: September 1977

**I2G: Code for Conversion of
Isotope-Ordered Cross-Section Libraries into
Group-Ordered Cross-Section Libraries**

W. M. Resnik II
G. E. Bosler

MASTER

los alamos
scientific laboratory
of the University of California
LOS ALAMOS, NEW MEXICO 87545

↓ ↓
An Affirmative Action/Equal Opportunity Employer

UNITED STATES
ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION
CONTRACT W-7405-ENG. 36

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

Price: Printed Copy ~~\$4.50~~ Microfiche \$3.00

44.50
This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.

TABLE OF CONTENTS

ABSTRACT	1
I. INTRODUCTION	1
II. STANDARD DATA MANAGEMENT SUBROUTINES	2
A. Sequential I/O Routines.	2
B. Multilevel Random Access Routines.	3
III. DESCRIPTION OF CODE.	4
A. Code Driver.	5
B. Program Initialization Routine	5
C. File Processing Routines	6
D. Long-Word - Short-Word Computer Conversion	6
E. Program-Storage Requirements	6
F. Calculation of Total Scattering from Partial Scattering Matrices.	6
G. User Input	7
H. File Unit Assignments.	7
IV. STORAGE.	7
A. Fast-Core Storage (SCM).	7
1. Basic Fast-Core Requirements	7
2. Single- and Double-Hierarchy Machines.	8
3. Fast-Core Storage Algorithm.	8
B. Extended-Core Storage (LCM).	10
V. MULTILEVEL DATA CONVERSION METHOD.	11
VI. CONDITIONS AND LIMITATIONS	14
VII. TYPICAL EXECUTION TIMES.	14
ACKNOWLEDGMENTS.	14
REFERENCES	15

APPENDIXES

Appendix A - ISOTXS and GRUPXS Standard File Descriptions.	15
Appendix B - Functions of the I2G Subroutines.	25

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

CONTENTS (cont)

FIGURES

Fig. 1. Scheme for multilevel, random access data transfers	3
Fig. 2. I2G code block flow diagram	5
Fig. 3. Contents of I2G code container array A.	8
Fig. 4. General scheme for multilevel, random access data transfers . .	12
Fig. 5. Example SCM-LCM transfers for conversion of principal cross sections.	13
Fig. 6. Example SCM-LCM transfers for conversion of principal cross sections.	13

TABLES

Table I. I2G Array Descriptions	9
Table II. I2G Test Run Times	14
Table A-I. Functions of I2G Subroutines	15

I2G: CODE FOR CONVERSION OF ISOTOPE-ORDERED CROSS-SECTION LIBRARIES
INTO GROUP-ORDERED CROSS-SECTION LIBRARIES

by

W. M. Resnik II
and
G. E. Bosler

ABSTRACT

Many current reactor physics codes accept cross-section libraries in an isotope-ordered form, convert them with internal preprocessing routines to a group-ordered form, and then perform calculations using these group-ordered data. Occasionally, because of storage and time limitations, the preprocessing routines in these codes cannot convert very large multigroup isotope-ordered libraries. For this reason, the I2G code, i.e., ISOTXS to GRUPXS, has been written to externally convert isotope-ordered cross section libraries in the standard file format called ISOTXS to group-ordered libraries in the standard format called GRUPXS. This code uses standardized multilevel data management routines which establish a strategy for the efficient conversion of large libraries. The I2G code is exportable contingent on access to, and an intimate familiarization with, the multilevel routines. These routines are machine dependent and therefore must be provided by the importing facility.

I. INTRODUCTION

Historically, reactor physics codes such as DTF-IV,¹ ANISN,² ONETRAN,³ TWOTRAN,⁴ etc. have been structured to accept multigroup cross sections in an isotope- (material-) ordered form. These codes contain preprocessing routines which read the cross sections in isotope-ordered matrices, each of which is dimensioned IGM by IHM (IGM is the number of energy groups and IHM is the table length for each group), and convert these matrices into an internal group-ordered structure. This conversion is performed because group-ordered formats are generally more amenable for use in actual reactor physics calculations than the isotope-ordered format originally read as input. Over the past several years, however, the number of groups in cross-section libraries has tended to increase rather dramatically. A 240-group library generated at Los Alamos Scientific Laboratory⁵ (LASL) and a 171-group library produced at Oak Ridge National Laboratory⁶ (ORNL) are typical examples. Generally, fast core storage allocated within the preprocessing routines which are attached to the

reactor physics codes cannot meet the storage requirements of the isotope-ordered tables which are constructed from these large multigroup libraries. While the very fine group structures in these large libraries may be collapsed into coarser group structures to ease the storage problem, often users prefer to process the entire uncollapsed library at once. A solution to the storage problem which eliminates the need to collapse large libraries before preprocessing is to externally convert these large isotope-ordered libraries into group-ordered libraries and then to input the converted library to the reactor physics codes. To perform this conversion, the exportable I2G code has been written to efficiently convert cross-section data from an isotope-ordered format to a group-ordered format.

Guidelines^{7,8} for code standardization established by the Committee on Computer Code Coordination (CCCC) were followed whenever possible during the development of the I2G code. The CCCC is a group of representatives from various reactor community facilities

assembled by the Energy Research and Development Administration's (ERDA) Division of Reactor Development and Demonstration (RDD) to provide a set of standards, procedures, and recommendations for the development of fast reactor physics codes. The resulting specifications and guidelines from the CCCC were formulated to be compatible with several different large-scale computing environments and, when implemented, to facilitate the interchange of codes between computing facilities.

As part of the CCCC efforts, specifications for a set of standard interface files were developed.^{9,10} Included in this set are files for cross-section data in an isotope-ordered format called ISOTXS and also for cross sections in a group-ordered format called GRUPXS. These files, particularly the ISOTXS file, have now been generally accepted as the standards for exchanging applicable fast reactor data between codes and when necessary, between facilities. In keeping with these standards the I2G code was created to read isotope-ordered data in the ISOTXS format and convert it to group-ordered data in the GRUPXS format. See Appendix A for file descriptions.

As previously mentioned, storage is a major problem when internally converting larger libraries. For example, internal code processing for a 171-group library which has matrices with IGM=171 and IHM=174, requires nearly 30 000 words of fast (small) core memory (SCM) for each matrix before a group-ordered library can be written. Such storage requirements are excessive for some computer systems. In the I2G external processing scheme, ISOTXS data are efficiently converted to group-ordered formats without forming IGM*IHM matrices, thereby greatly reducing the fast core storage requirements of large libraries. A multilevel storage strategy is used in I2G for transferring data between fast core (SCM), extended core (large core memory, i.e., LCM), and disk. Standard routines, which have been adopted by the CCCC for trial use, are employed in I2G for the multilevel transfers. Specifications for these routines are described in Sec. II.B. and Ref. 9. The multilevel storage schemes used in I2G for processing the principal and scattering cross sections are described in detail in Sec. V. The multilevel storage routines used by the I2G code are installation dependent and are therefore not included in the I2G code package.

II. STANDARD DATA MANAGEMENT SUBROUTINES

In order to achieve compatibility between various computers for sequential file data transfers and for multilevel random access data management, a set of standardized data management routines has been established by the CCCC. While the functions, call names, and argument lists of these routines are standardized, coding for the routines is left entirely to the installation using them. Any systems-dependent coding appears in the installation-dependent routine which is isolated from the main code body. All of the routines described in this section are more thoroughly discussed in Ref. 9.

A. Sequential I/O Routines

In adhering to standards established by the CCCC, three subroutines, i.e., SEEK, REED, and RITE, are used in I2G for all sequential file management and for data transfers between central memory and in-line storage devices.

In subroutine SEEK, a catalog of data files and their current status is maintained. In I2G, SEEK is used to manage three sequential files; namely, the ISOTXS file, the GRUPXS file, and a scratch file. The scratch file is used as temporary storage for isotope chi data. The simplified, free-standing version of SEEK included with the I2G code package assigns logical unit number 19 to GRUPXS, unit 20 to ISOTXS, and unit 21 to the scratch file.

The argument list for SEEK is as follows:

SEEK(HNAME, IVERS, NREF, NOP)

where HNAME is the Hollerith name of the file, IVERS is the file data-version number, NREF is the logical unit number of the file, and the NOP option defines the type of operation to be performed. Options are available for initializing and finalizing the catalog of files as well as for assigning logical unit numbers.

The RITE and REED routines transfer data respectively to or from the units (sequential storage files) assigned by SEEK. The standard form of the REED and RITE argument lists is as follows:

REED(NREF, IREC, ARRAY(I), NWDS, MODE)

RITE(NREF, IREC, ARRAY(I), NWDS, MODE)

where NREF is the logical unit number assigned by SEEK, IREC is the record number in the file, ARRAY(I)

is the array and location in the array to or from which data is transferred, NWDS is the number of single-precision words to be transferred, and MODE dictates the type of operation to be performed. Records may be accessed in any order by REED calls since REED uses the IREC parameter to internally position a file through rewinds, backspaces, and dummy reads in a pseudorandom fashion. Subroutine RITE, however is strictly a sequential operator and therefore records cannot be written out of sequence.

B. Multilevel Random Access Routines

Five standardized-for-trial-use routines DOPC, DRED, DRIT, CRED, and CRIT are used for transferring data between the different levels of storage hierarchy. Figure 1 displays the multilevel strategy which utilizes these routines. The terminology used in describing the multilevel routines which are exactly as defined in Ref. 9 is included here also for convenience:

Fig. 1. Scheme for multilevel, random access data transfers.

Extended Core: That portion of a computing system containing storage locations which serve as buffer for random access data. Extended core may be physically separate from central memory as a peripheral large core memory (LCM) on two-hierarchy memory computers or it may be a portion of central memory which has been designated as extended core on single-hierarchy memory computers.

Fast Core: That portion of a computing system which contains storage for both data and instructions, which is directly coupled to the computations portion of the system, and which is directly coupled to extended core. Fast core may be the entire central memory, i.e., small core memory (SCM) on two-hierarchy memory machines, or it may be that portion of central memory remaining after an extended core portion is designated on single-hierarchy memory machines.

Random Access Data: Data which can be transferred between fast core and extended core in out-of-sequence strings.

Random Access File: (Also called Direct Access File). A named collection of data which is stored on a peripheral storage device. The file data are arranged in blocks which can be transferred between extended

core and peripheral storage randomly, i.e., out of sequence.

Logical File: A random access file. The identity of a logical file in a code is established by an integer variable, the (logical) file reference number.

Block: A subportion of a random access file. The block size is determined by the amount of extended core storage that can be allocated to the random access file. Data transfers between the peripheral device and extended core are block transfers.

String: A subportion of a random access file block. Data transfers between fast core and extended core are string transfers.

Disk: A generic name for peripheral storage device used for storing random access files.

Physical Unit: An identifiable subpart of a disk. One or more physical units compose a disk.

File Group: A collection of one or more random access files. The file-group collection is assigned to a single physical unit.

The scheme shown in Fig. 1 is the standardized multilevel data management strategy using random access disk. Each random access file is composed of blocks and is contained on an accessible storage device, usually disk. Generally, block sizes are determined from the maximum amount of available extended core storage as specified within the code. When needed, these blocks of data are transferred, a block at a time, between extended core and the physical unit. The block contained in extended core at any given time is further partitioned by the code into strings which may then be transferred between extended core and fast core memory.

On two-hierarchy computers such as CDC machines, extended core (LCM) is physically separate from fast core memory (SCM), while on a single-hierarchy computer such as IBM, extended core memory is that portion of central core remaining after fast core memory is designated.

Subroutine DOPC controls the opening and closing of the physical units which are used for storing the multilevel data files. The code determines the maximum size and number of blocks required and passes this information to DOPC. The argument list for DOPC is as follows:

```
DOPC(IOP, NREF, NREF, NGREF, IERRCS, ARRAY(K), MXLEN,
      MXBLOK, LENFIL)
```

A detailed description of the role of the arguments is given in Ref. 9. Basically, three calls to DOPC are required in order to initialize information before any I/O to the physical unit can be performed. The first call (IOP=0) initializes the routine and on single-hierarchy machines, determines the starting address of extended core memory in central core [using ARRAY(K+1) where K is the last word of required fast core memory]. The second call (IOP=1) determines the lengths of the file, while the third call (IOP=2) assigns a physical unit to the file. After a particular physical unit is no longer needed, it is good practice to call DOPC a fourth time (IOP=4) in order to close the unit and free the storage space.

Subroutines DRED and DRIT perform block transfers between disk and extended core memory; DRED reads data from disk into extended core and DRIT writes data from extended core onto disk. For these

transfers, all or part of a requested block may be transferred, however, a transfer must begin with the first word of a block. A complete description of DRED and DRIT is included in Ref. 9 and the argument lists for these routines are as below:

```
DRED(NREF, NBLOK, LCM, NBWDS, MODE)
DRIT(NREF, NBLOK, LCM, NBWDS, MODE)
```

A call to DRED transfers the first NBWDS words from block NBLOK in the logical file reference number NREF to the location in extended core called LCM.

A call to DRIT performs the reverse operation.

These transfers may take place randomly, i.e., out of sequence. Since the time to complete each transfer may be relatively long, the number of such transfers should be minimized, i.e., while a block is a resident in extended core, as much of the data in the block should be processed as possible before another DRED/DRIT call is issued.

The subroutines CRED and CRIT perform string transfers between extended core memory and fast core memory; CRED reads data from extended core to fast core and CRIT writes data from fast core to extended core. For these transfers, all or part of a requested string may be transferred and string lengths are determined in the main code. A formal discussion of CRED and CRIT is included in Ref. 9 and the argument lists for these routines are as follows:

```
CRED(ARRAY(I), LCM, NSWDS, IERR)
CRIT(ARRAY(I), LCM, NSWDS, IERR)
```

A call to CRED transfers NSWDS words starting at location LCM in extended core to the address in fast core starting with ARRAY(I). The parameter NSWDS is defined as the string length, i.e., the number of single-precision words in the string. A call to CRIT performs the reverse operation. Data strings may be transferred to or from extended core in any order. These transfers usually require very little time, and thus can be used more often than the more time-consuming DRED and DRIT calls.

III. DESCRIPTION OF CODE

A block diagram of the I2G code is shown in Fig. 2 which displays the basic flow of the code.

Fig. 2. I2G code block flow diagram.

Specific subroutines are discussed in Appendix B. Throughout the code, file management and I/O for all sequential files is performed by calls to the standard subroutines SEEK, REED, and RITE. All random access storage is managed by using the standard-for-trial-use multilevel routines DOPC, DRED, DRIT, CRED, and CRIT. All of these file management and random access storage routines are described in Sec. II.

A. Code Driver

The code driver routine, I2GDRV, performs the following functions:

- a) Establishes dimensions of the container array "A".
- b) Establishes the maximum amount of extended core storage.
- c) Initializes the SEEK catalog.
- d) Calls SEEK to assign logical unit numbers to the ISOTXS, GRUPXS, and scratch files.
- e) Initializes DOPC (IOP=0).
- f) Calls the program storage initialization routine I2GPIR.
- g) Calls the file processing routines.
- h) Computes and prints the code execution time.
- i) Finalizes SEEK and DOPC.

These operations take place sequentially in the order described.

B. Program Initialization Routine

The main function of the program storage initialization routine, I2GPIR, is to read the ISOTXS file control data (1D record) and from it, determine the pointers for variable dimensioned arrays within the container array "A". (See Sec. IV.A.1.) Applicable 1D data and calculated array pointers are returned to the I2GDRV routine through a COMMON statement labeled LABCOM. Since pointers are calculated in terms of equivalent single words, word counts for Hollerith variables (which must be made double precision on short-word machines, e.g., IBM) are determined by multiplying the number of actual required Hollerith words by the standardized MULT parameter. The MULT parameter is set equal to 1 on long-word (CDG) machines and to 2 on short-word (IBM) machines to accommodate these word count differences. The MULT parameter is described in detail in Ref. 9.

C. File Processing Routines

Each of the file processing routines creates, processes, and/or writes a single GRUPXS record type. For easy identification, the names of these routines were made indicative of their function and/or the record type that they process. For example, the routine that writes the principal cross sections (5D record) is called WRIT5D and the routine which writes the scattering cross sections (8D record) is called WRIT8D. A discussion of the operations of some of the actual routines is provided in Sec. V and a complete list of the routines and their functions is given in Appendix A.

D. Long-Word - Short-Word Computer Conversion

The inherent differences in long-word (CDC) and short-word (IBM) computers require that minor coding changes be made when exchanging the I2G code between installations with different machine types. For example, if a short-word version of the code is required, double-precision references such as REAL*8 are necessary. However, when a long-word version of the code is required, this reference is unnecessary (and not recognized on CDC machines). Thus all REAL*8 references are "commented out" and bracketed by CIBM cards, i.e.,

```
Column 1
  ↓
  CIBM
  C    REAL*8
  CIBM
```

Similarly, short-word versions have long-word statements bracketed by CDC* cards. Another example is the double-precision MULT parameter briefly discussed in Sec. III.B. The long-word MULT=1 statement is bracketed by CDC* cards and the short-word MULT=2 parameter is bracketed by CIBM cards. The nonapplicable statement must be commented out. Conversion between computers is achieved by removing or inserting the "C" in column 1 of the appropriate statements. If desired, a simple code may be written which will recognize the CIBM or CDC* brackets and make the changes automatically.

E. Program-Storage Requirements

On the CDC-7600 computer at LASL, the memory requirements for the I2G code routines are as follows:

<u>Routine</u>	<u>Octal</u>	<u>Decimal</u>
I2GDRV	547	359
I2GPIR	231	153
FILL5D	1 472	826
WRITID	24	20
WRIT1D	72	58
WRIT2D	155	109
WRIT3D	24	20
WRIT4D	24	20
WRIT5D	412	266
WRIT6D	74	60
SCATER	1 102	578
SCTLCM	230	152
SCATLG	1 113	587
SCTLGG	117	127
WRIT7D	311	201
SORTBL	107	71
TBLPOS	25	21
WRIT8D	151	105
TIMER	13	11
SEEK	133	91
RITE	140	96
REED	141	97
DOPC	1 642	930
DRED	211	137
DRIT	227	151
CRED	62	50
CRIT	62	50
TOTAL	12 342	5 346

With a container array dimensioned at 30 000₁₀ words, the total storage requirements of the free-standing version of the code (including the multilevel routines) is 35 346 words. These program storage requirements are associated only with the code itself and do not include requirements for necessary systems routines. The extended core memory requirements discussed in Sec. IV.B are more or less installation dependent, but it is recommended that as much extended core as possible be made available to the code. (See Sec. IV.B.)

F. Calculation of Total Scattering from Partial Scattering Matrices

Only total scattering cross sections are written to the GRUPXS library produced by the I2G code. All partial scattering matrices, i.e., elastic, inelastic, and $n, 2n$, are excluded even when present in the ISOTXS file. If the total scattering matrices are not present

in the ISOTXS file, then they are calculated by summing the partial scattering matrices, i.e.,

$$\sigma_{\text{TOTAL}}(g \rightarrow g') = \sigma_{\text{ELAS}}(g \rightarrow g') + \sigma_{\text{INELAS}}(g \rightarrow g') + \text{IPRAM} \cdot \sigma_{n,2n}(g \rightarrow g'),$$

where

$\sigma_{\text{TOTAL}}(g \rightarrow g')$ is the total cross section for scattering from group g to group g' ,

$\sigma_{\text{ELAS}}(g \rightarrow g')$ is the elastic scattering from g to g' ,

$\sigma_{\text{INELAS}}(g \rightarrow g')$ is the inelastic scattering from g to g' ,

$\sigma_{n,2n}(g \rightarrow g')$ is $n,2n$ transfers from g to g' ,

and

IPRAM is either 1 or 2 depending on whether the $n,2n$ matrix is production or reaction based.

In the specifications for the Version IV ISOTXS file described in Ref. 9, the $n,2n$ matrix was designated to be production based, i.e., the macroscopic $n,2n$ scattering from group g to group g' times the flux in group g yields the total number of neutrons emitted into group g' . In other words, with a Version IV ISOTXS file, the total $n,2n$ reaction cross section for group g can be calculated from the $n,2n$ scattering matrix by

$$\sigma_{n,2n}(g) = 0.5 \sum_{g'} \sigma_{n,2n}(g \rightarrow g').$$

In the Version III ISOTXS specifications it was not clear as to whether the $n,2n$ scattering matrix was reaction-based or production-based. As a result, libraries have been produced in which some are production-based while others are reaction-based. In the former the total $n,2n$ cross section for a group g is defined as above for a Version IV file while in the latter the total $n,2n$ cross section is

$$\sigma_{n,2n}(g) = \sum_{g'} \sigma_{n,2n}(g \rightarrow g').$$

For convenience, the variable IPRAM is included in the I2G code in subroutines SCTLGM and SCATLG for

summing the $n,2n$ matrices. In the basic version

of I2G, production based $n,2n$ matrices are assumed and IPRAM is set equal to 1. However, appropriate comment card brackets around a test for the $n,2n$ matrix, i.e.,

```
Column 1
      ↓
      CVERS3
      C   IF(IDSCT.GE.300.AND.IDSCT.LT.400)IPRAM=2,
      CVERS3
```

have been included for easy conversion to files which have reaction based matrices. If a file which exists without the total scattering matrix has a reaction based $n,2n$ matrix, a GRUPXS file with appropriate total scattering will be created by simply removing the "C" in column 1 of the card bracketed by CVERS3 which defines the variable IPRAM.

G. User Input

There is no user input for the I2G code; however, when the total cross section is not present and therefore must be calculated by summing the partial scattering matrices and when the $n,2n$ matrix is reaction based, the IPRAM parameter must be set accordingly within the code (see Sec. III.F).

H. File Unit Assignments

The I2G code requires four logical units as below:

Unit 6, i.e., TAPE0 or FSET6, -	printed output file
Unit 19	- GRUPXS file
Unit 20	- ISOTXS file
Unit 21	- scratch file

These logical unit numbers, with the exception of unit 6, are assigned by SEEK. Unit 6 is defined by the statement NOUT=6 in the I2GDRV routine.

IV. STORAGE

A. Fast-Core Storage (SCM)

1. Basic Fast-Core Requirements

All of the arrays used in the file processing routines of the I2G code reside in and are subsets

of a single container array, A. The maximum dimension length for the CDC version of the A array is set at 30 000₁₀ single-precision words by a FORTRAN COMMON statement in the code driver program I2GDRV. Figure 3 depicts the structure of the container array used in the I2G code. It gives the name of each of the contained arrays, their lengths (in parenthesis), their physical locations in the A array in relation to the other arrays, and the pointer words (underneath each of the partitioning lines) used in the main program which single out the arrays. Arrays which are specified twice indicate either common arrays or shared storage space. These data may be seen in tabular alphabetical form in Table I. Also included in Table I is the function of each array and the subroutine(s) in which each array is used.

After the principal cross sections are converted, the fast core memory space used by the principal cross-section arrays is no longer needed and is therefore available for processing the scattering cross sections. By reusing this space a significant amount of storage can be conserved. The A array space shared in fast core by the principal cross section and scattering cross-section conversion arrays is enclosed in the dotted rectangle labeled "Common SCM Strings" in Fig. 3. The final pointer words for the arrays used in processing either the

principals or scattering cross sections are ILAST and JLAST, respectively. The maximum amount of SCM space used by the code is equal to the maximum of either ILAST or JLAST.

2. Single- and Double-Hierarchy Machines

On a single-hierarchy storage machine (IBM), extended core memory is defined as that memory which remains in central core when the maximum required amount of fast core memory (MAXSCM parameter in the I2G code) is designated. The MAXSCM parameter is determined in the I2G code by obtaining the maximum of the ILAST and JLAST parameters and adding 1 to it. The MAXSCM parameter is passed through the first call to subroutine DOPC which then defines extended core storage in a single-hierarchy storage setup. In order to ensure that enough extended core space is available, the container array A must be dimensioned greater than or equal to the sum of the maximum expected A array size plus the MAXLCM parameter. (See Sec. IV.B.) When the I2G code is run on a double-hierarchy storage machine (CDC), where extended core is physically separate from fast core, the MAXSCM parameter, as specified as ARRAY(MAXSCM) on the IOP=0 DOPC call, is irrelevant and is therefore ignored.

3. Fast-Core Storage Algorithm

The following storage algorithm can be used for calculating the I2G fast-core storage requirements for a particular ISOTXS library. For this algorithm, it

Fig. 3. Contents of I2G code container array A.

TABLE I
I2G ARRAY DESCRIPTIONS

<u>Array Names</u>	<u>Subroutines</u>	<u>Pointer Words</u>	<u>Position in IE</u>	<u>Remarks</u>
AISO2D	WRIT2D	IP1	12	ISOTXS file data
AISO4D	FILL5D	IP2	13	ISOTXS isotope control data; equivalenced to ISO4D array
CHI	FILL5D,WRIT6D	IP7	18	GRUPXS isotope chi data
CHI3D	WRIT3D	IP1	12	GRUPXS file-wide chi data
FILEID	WRITID	1	-	GRUPXS file identification
GRP2D	WRIT2D	IP1A	35	GRUPXS file data
GRP4D	FILL5D,WRIT4D	IP5	16	GRUPXS isotope control data
IBND	SCATER,SCATLG, SCTLCM,SCTLGG	IP11	22	GRUPXS scattering table lengths
ICHI6D	FILL5D,WRIT6D	IP6	17	number of words in each GRUPXS isotope chi record
IGRP1D	WRITID	1	-	GRUPXS file control data
IGRP4D	FILL5D	IP5	16	GRUPXS isotope control data; equivalenced to GRP4D array
IJJ	SCTLCM,SCTLGG	IJJ	-	ISOTXS scattering control data; in ISO4D array
IJJBND	SCATER,SCATLG, WRIT7D,WRIT8D	IP8	19	GRUPXS scattering control data
ISO4D	FILL5D,SCATER, SCATLG,SCTLCM, SCTLGG	IP2	13	ISOTXS isotope control data equivalenced to AISO4D array
ISRT	SCATER,SCATLG, WRIT7D, SORTBL, WRIT8D	IP15	26	starting locations in GRUPXS scattering cross sections tables
ITOT	FILL5D,SCATER, SCATLG	IP18	34	flag which determines if the total scattering cross sec- tion is present for a particular isotope
IJJJ	SORTBL	IP	-	GRUPXS self-scattering positions contained within IJJBND array
JOBND	SORTBL	1	-	GRUPXS scattering bandwidths contained within IJJBND array
NGP3	SCATER,SCATLG, WRIT7D	IP13	24	number of energy groups per block for a particular block
NWDSPB	SCATER,SCATLG WRIT7D	IP14	25	number of words per block for a particular block
PARCON	SCATER,SCATLG, WRIT7D,SORTBL, TBLPOS,WRIT8D	IP9	20	partially converted scattering cross sections transferred from extended core to fast- core memory
PX5G	FILL5D,WRIT5D	IP4	15	GRUPXS principal cross sections
PXSI	FILL5D,WRIT5D	IP3	14	ISOTXS principal cross sections
SCATG	SCATER,SCATLG, WRIT7D,WRIT8D	IP10	21	GRUPXS scattering cross sections
SCATI	SCATER,SCATLG, SCTLCM,SCTLGG	IP9	20	ISOTXS scattering cross sections
SCATP	SCATER,SCATLG, SCTLCM,SCTLGG	IP10	21	partially converted scattering cross sections transferred from fast-core to extended-core memory

is necessary to define some ISOTXS control parameters which will be used. These parameters are used by the I2G code itself for determining pointer words in the container array A and for computing the value of the maximum required fast-core storage, i.e., the MAXSCM parameter.

NGROUP - number of energy groups in the file
 NISO - number of isotopes (materials) in the file
 MAXUP - maximum number of upscattering groups
 MAXDN - maximum number of downscattering groups
 MAXORD - maximum scattering order
 ICHIST - file-wide fission spectrum flag
 NSCMAX - maximum number of blocks of scattering data
 NSBLOK - subblocking control for scatter matrices
 MAXT - MAXORD+1
 MULT - equal to 1 on long-word (CDC) machines
 equal to 2 on short-word (IBM) machines

Before beginning the calculation of the algorithm, it is first necessary to compute a maximum IMAX,

$$IMAX = \text{the maximum of } \left(\begin{array}{l} \text{MAXT} * \left((1 - \text{NSBLOK} / \text{NGROUP})^{\dagger} * \text{NGROUP} + (\text{NSBLOK} / \text{NGROUP})^{\dagger} \right) \\ \text{or NISO} \end{array} \right).$$

This IMAX parameter is important for determining the number of words in the scattering arrays; SCATI, PARCON, SCATP, and SCATG.

Step 1:

Compute the following value for IBEGIN:

$$IBEGIN = \text{NGROUP} * (6 + 2 * \text{NSCMAX}) + (27 + 2 * \text{NISO}) * \text{MULT} + \text{NSCMAX} * 2 + \text{NISO} + 21.$$

Step 2:

Compute the following 2 values for IPXSCT and ISCTER:

$$\begin{aligned} IPXSCT &= (12 + 2 * \text{MAXT}) * (\text{NISO} + \text{NGROUP}) + \text{NISO} * (3 * \text{MULT} + 10) + 2 * \text{MAXT} + 7 * \text{NGROUP} * (\text{ICHIST} + 1) \\ ISCTER &= 2 * \text{MAXI} * \text{NISO} + 2 * (\text{MAXUP} + \text{MAXDN} + 1) * \text{IMAX} + 4 * \text{NGROUP} \end{aligned}$$

$$\text{MAXSCM} = \text{IBEGIN} + \text{the maximum of } \left(\begin{array}{l} \text{IPXSCT} \\ \text{and} \\ \text{ISCTER} \end{array} \right).$$

On some systems, the user may designate this value as the required SCM for the job and may thus save storage charges.

B. Extended-Core Storage (LCM)

Extended-core storage is used by the I2G code in conjunction with the multilevel data transfers. On some machines such as the CDC computers, extended-core storage actually occurs in large core memory (LCM) a device which is physically separate from fast-core memory. On other machines such as IBM and UNIVAC computers, extended core is a portion of fast (central core) memory. In the I2G code, the MAXLCM parameter, which is initialized in the I2GDRV routine, specifies the amount of available extended core storage. For single storage level computers, i.e., IBM, the total amount of fast-core storage required for an execution may be defined as

$$\text{MAXTOT} = \text{MAXSCM} + \text{MAXLCM} + 1$$

where MAXSCM is described in Sec. IV.A.

For computer systems in which the maximum amount of available LCM is fixed, MAXLCM should be set to

just below this fixed value. However, in systems

Step 3:

Add IBEGIN to the maximum of IPXSCT and ISCTER to obtain MAXSCM

where turn-around time is dependent on the amount of storage being used, MAXLCM may be adjusted to exactly the amount required for each execution.

The I2G code automatically adjusts data storage blocking to make efficient use of extended core space

[†]Note: This computation is performed in "integer arithmetic," i.e., any fractional portions are dropped.

specified by the MAXLCM parameter. When possible, the storage specified by MAXLCM should be large enough to contain all cross sections in one block. This will ensure optimum efficiency in using the multilevel data transfer routines by eliminating potentially slow disk-to-extended-core data transfers and using only the relatively fast core-to-core transfers. In order to contain all of the principal cross sections,

$$\text{MAXPRN} = (12+2*\text{MAXT})*\text{NISO}* \text{NGROUP}^{\dagger}$$

words in extended-core storage are required. The scattering cross sections require

$$\text{MAXSCT} = \text{NISO}*\text{MAXT}*\sum_{g=1}^{\text{NGROUP}} \text{IBND}(g)^{\dagger}$$

total words of extended-core storage where

$$\text{IBND}(g) = \text{MUP}_g + \text{MDN}_g + 1,$$

$$\text{MUP}_g = \text{Minimum of MAXUP or NGROUP}-g$$

and

$$\text{MDN}_g = \text{Minimum of MAXDN or } g-1.$$

Since the principal and scattering cross sections are processed separately, the maximum amount of extended-core storage required is the greater of MAXPRN and MAXSCT.

For libraries which are not totally containable in extended core, the data are blocked and disk-to-extended core transfers are made in addition to the core-to-core transfers. Block sizes for disk transfers are calculated from the MAXLCM parameter by determining the number of groups of data which can be stored within the available extended-core space. For these blocked cases, extended-core storage must be large enough to contain at least one group of data. Thus, for the principal cross sections, the minimum number of required words is

$$\text{MINPRN} = (12+2*\text{MAXT})*\text{NISO}$$

[†] MAXT, NISO, NGROUP, MAXUP, and MAXDN parameters are defined in Sec. IV.A.3.

and the minimum for scattering data is

$$\text{MINSCT} = (\text{MAXUP}+\text{MAXDN}+1)*\text{NISO}*\text{MAXT}$$

single precision words. Thus, the minimum allowable size of MAXLCM is determined by taking the greater of MINPRN and MINSCT. If MAXLCM is less than this value, the library cannot be converted.

V. MULTILEVEL DATA CONVERSION METHOD

In Fig. 4, the multilevel data transfer scheme for converting scattering cross sections in an isotope-ordered library to the corresponding data in a group-ordered library is shown. Principal cross sections are similarly converted. While two groups per block are shown in Fig. 4 for clarity, the actual number of blocks and the number of energy groups per block are problematically determined from the MAXLCM parameter as described in Sec. IV.B. Basically, for both cross-section types, the conversion scheme is as follows:

Isotope-ordered cross sections are read into fast core from the sequential file, stored in extended core by group, and then, if necessary, written from extended core to disk in blocks. After all of the isotopes have been processed, the data blocks on disk are read, in sequence, into extended core, each group of data is transferred to fast core and data is then written on the group-ordered file.

The processes for converting both principal and scattering data are separately described in this section.

The two subroutines which convert the isotope-ordered principal cross sections to group-ordered records are called FILL5D and WRIT5D. The subroutine FILL5D processes the ISOTXS principal cross-section array and sends these data to extended core, and if necessary, to disk. Then, the WRIT5D routine retrieves these data, completes the conversion, and then writes the GRUPXS principal cross-section (5D) record.

The FILL5D routine begins by partitioning extended core and determining the sizes and numbers of blocks. The ISOTXS isotope control data (4D record) are used to determine conversion parameters and also to construct the corresponding GRUPXS isotope control data arrays. Next, as the ISOTXS

Fig. 4. General scheme for multilevel, random access data transfers. (Note: two groups per block shown only for clarity.)

principal cross sections (5D record) are read for each isotope, they are transferred into another array for disposition to extended core. Figure 5 shows the contents of the PXSI array before it is converted and transferred into extended core. Once again, disk transfers are necessary only when the cross sections cannot be completely contained in extended core. The WRIT5D routine retrieves these data from extended core, simplifies these data if possible, (i.e., if partial cross-section types are not present for any isotope, space for that partial is eliminated), and then writes the GRUPXS 5D record.

The eight routines which perform the conversion of the scattering cross sections are SCATER, SCATLG, SCTLGM, SCTLGG, WRIT7D, SORTBL, TBLPOS, and WRIT8D. The pairs SCATER-SCTLGM and SCATLG-SCTLGG perform similar functions with the latter pair requiring less fast-core storage under the following conditions. The SCATLG-SCTLGG pair is called when the number of energy groups equals the number of scattering subblocks over all groups. In this case, only one group at a time is processed while reading the ISOTXS file. Otherwise, the SCATER-SCTLGM pair is called and all groups are processed from ISOTXS at one time. Once in either SCATER or SCATLG, extended-core blocks with group-dependent scatter table lengths are computed before the ISOTXS scattering cross sections are read. The total scattering cross sections are produced and/or processed (see Sec. III.F) and then written to appropriate group-structured storage locations in extended core. These cross sections, as they reside in extended core before disk transfers, are depicted in Fig. 6. The WRIT7D routine, called by either SCATER or SCATLG, retrieves the cross sections from extended core, determines the bandwidths and in-group scattering positions for the data, and then writes the GRUPXS scattering control (7D) record. The final processing takes place in the WRIT8D routine, called by WRIT7D, where the scattering data are written to the GRUPXS file using the scattering control data generated in WRIT7D. At the conclusion of the WRIT8D routine, program control is returned to I2GDRV for a wrap-up finalization of SEEK and DOPC.

Fig. 5. Example SCM-LCM transfers for conversion of principal cross sections.

Fig. 6. Example SCM-LCM transfers for conversion of principal cross sections.

VI. CONDITIONS AND LIMITATIONS

The I2G code, like any other code, has certain known conditions under which it will not properly execute. The following conditions and limitations, if carefully considered and adhered to, should insure proper execution of the code.

- (1) ISOTXS input must be a single Version III or a Version IV binary type sequential file input. Multiple files and BCD card image type input cannot be read by the code.
- (2) Special attention should be paid to the amount of fast-core and extended-core memory space available locally to insure that a large library can be converted. The MAXSCM and MAXLCM parameters may be adjusted within the code to mold the storage requirements of the code with the storage capabilities of an installation. Caution is advised, however, in doing so. (See Sec. IV.B.)
- (3) An extended core block must be able to contain at least one energy group of cross-section data. (See Sec. IV.B.)
- (4) Because of the intrinsic strategy of the I2G code in conversion of the scattering cross sections (see Sec. V) libraries containing many isotopes may require significantly longer conversion times than those libraries containing only a few isotopes. (See Sec. VII.) Accordingly, unnecessary isotopes should be eliminated from the ISOTXS file before I2G processing.
- (5) Only the total scatter matrix is produced by the I2G code. If the total scattering matrix is not present in the ISOTXS input, it is computed by summing the partial scattering matrices. If the total matrix is not present and the $n, 2n$ cross section is reaction based, the IPRAM parameter must be changed within the code. (See Sec. III.F.)

VII. TYPICAL EXECUTION TIMES

Table II shows actual execution times on the LASL CDC computer for the I2G code under various conditions. In case 1, extended core space of at least 50 000 words was needed to contain the file entirely in extended core, disk transfers were forced by setting the maximum extended core size to 20 000 words (MAXLCM=20 000). This resulted in an increased execution time over that of case 2, which used the same library but in which MAXLCM was set large enough to contain the entire file. In cases 3 and 4, while the libraries were relatively large, they were entirely contained in extended core and, thus, execution times were relatively low. Cases 5 and 6, however, were not containable in extended core and thus showed a marked increase in execution time. It can be seen from these tests that libraries which contain a large number of isotopes generally require more execution time than those with many groups and few isotopes. The I2G code was written to be more efficient with large multigroup libraries and the timing tests prove this conclusively. Code execution times are computed by the standardized TIMER routine and printed as output by I2GDRV.

ACKNOWLEDGMENTS

The authors wish to express their sincerest thanks to R. D. O'Dell for his help and contributions to the I2G code and code manual. A special thanks goes to the technical artist Benjamin Atencio, whose extraordinary patience in drawing these detailed figures makes many complicated parts of this manual more clear.

TABLE II
I2G TEST RUN TIMES

Case	Number of Isotopes	Number of Energy Groups	Maximum LCM Size	Size ISOTXS Library	I2G Code Execution Time
1	5	50	20 000	60 718 ₁₀	3.16 s
2	5	50	50 000	60 718 ₁₀	1.23 s
3	1	240	320 000	262 579 ₁₀	3.17 s
4	1	265	320 000	1 327 263 ₁₀	4.27 s
5	9	171	324 000	1 189 316 ₁₀	51.96 s
6	101	50	324 000	4 271 601 ₁₀	361.4 s

REVISSED 11/18/76

ISOTXS-IV
MICROSCOPIC GROUP NEUTRON CROSS SECTIONS

THIS FILE PROVIDES A BASIC BROAD GROUP
LIBRARY, ORDERED BY ISOIOP
FORMATS GIVEN ARE FOR FILE EXCHANGE PURPOSES
ONLY.

APPENDIX A
ISOTXS AND GRUPTS STANDARD FILE DESCRIPTIONS

- REFERENCES
1. K. D. Lathrop, "DTF-IV, A FORTRAN-IV Program for Solving the Multigroup Transport Equation with Anisotropic Scattering," Los Alamos Scientific Laboratory report LA-3373 (November 1965).
 2. W. W. Engle, Jr., "A User's Manual for ANISN, A One-Dimensional Discrete Ordinates Transport Code with Anisotropic Scattering," Union Carbide Corporation report K-1693 (March 1967).
 3. T. R. Hill, "ONETRAN: A Discrete Ordinates Finite Element Code for the Solution of the One-Dimensional Multigroup Transport Equation," Los Alamos Scientific Laboratory report LA-5990-MS (June 1975).
 4. K. V. Lathrop and F. W. Brinkley, "TWOTRAN-II: An Interfaced, Exportable Version of the TWOTRAN Code for Two-Dimensional Transport," Los Alamos Scientific Laboratory report LA-4848-MS (July 1973).
 5. C. I. Baxman, G. M. Hale, and P. G. Young, "Applied Nuclear Data Research and Development," Los Alamos Scientific Laboratory report LA-6472-PR (August 1976).
 6. R. W. Kossin, C. R. Wetsbin, J. E. White, M. W. Green, R. Q. Wright, and J. B. Wright, "The CTR Processed Multigroup Cross-Section Library for Neutronics Studies," Oak Ridge National Laboratory report ORNL-RSIC-37 (to be published).
 7. W. H. Hannum and J. W. Lewellen, "Work of the Committee on Computer Code Coordination," Proc. Conf. New Devel. Reactor Math. and Appl., CONF-710302, 1, 452 (March 29-31, 1971, Idaho Falls, Idaho).
 8. D. K. Butler, Chairman, "Report and Recommendations of the Committee on Computer Code Coordination," prepared for the Advisory Committee on Reactor Physics, Los Alamos, New Mexico (November 4-5, 1970).
 9. R. D. O'Dell, "Standard Interface Files and Procedure for Reactor Physics Codes, Version IV," Los Alamos Scientific Laboratory report LA-6941-MS (September 1977).
 10. G. E. Bostler, R. D. O'Dell, W. M. Resnick, "LASIP-III, A Generalized Processor for Standard Interface Files," Los Alamos Scientific Laboratory report LA-6280-MS (April 1976).

```

C-----
CS      FILE STRUCTURE
CS
CS      RECORD TYPE                PRESENT IF
CS      *****
CS      FILE IDENTIFICATION        ALWAYS
CS      FILE CONTROL                ALWAYS
CS      FILE DATA                  ALWAYS
CS      FILE-WIDE CHI DATA        ICHI,GT,1
CS
CS      ***** (REPEAT FOR ALL ISOTOPES)
CS      *      ISOTOPE CONTROL AND GROUP
CS      *      INDEPENDENT DATA    ALWAYS
CS      *      PRINCIPAL CROSS SECTIONS  ALWAYS
CS      *      ISOTOPE CHI DATA      ICHI,GT,1
CS      *
CS      * ***** (REPEAT TO NSCMAX SCATTERING BLOCKS)
CS      * * ***** (REPEAT FROM 1 TO NSBLOK)
CS      * * * SCATTERING SUB-BLOCK    LORD(N),GT,0
CS      *****
C
C-----

```

```

C-----
CR      FILE IDENTIFICATION
C
CL      HNAME,(HUSE(1),I=1,2),IVERS
C
CH      1+3*MULT=NUMBER OF WORDS
C
CB      FORMAT(11H DV ISOTXS , 1H+,
CB      12A6,1H+,I6)
C
CD      HNAME      HOLLERITH FILE NAME = ISOTXS = (A6)
CD      HUSE(1)    HOLLERITH USER IDENTIFICATION (A6)
CD      IVERS      FILE VERSION NUMBER
CD      MULT        DOUBLE PRECISION PARAMETER
CD                  1- A6 WORD IS SINGLE WORD
CD                  2- A6 WORD IS DOUBLE PRECISION WORD
C
C-----

```

```

C-----
CR      FILE CONTROL  (10 RECORD)
C
CL      NGROUP,NISO,MAXUP,MAXDN,MAXORD,ICHIST,NSCMAX,NSBLOK
C
CH      8=NUMBER OF WORDS
C
CB      FORMAT(4H 10 ,8I6)
C
CD      NGROUP      NUMBER OF ENERGY GROUPS IN FILE
CD      NISO        NUMBER OF ISOTOPES IN FILE
CD      MAXUP        MAXIMUM NUMBER OF UPSCATTER GROUPS
CD      MAXDN        MAXIMUM NUMBER OF DOWNSCATTER GROUPS
CD      MAXORD        MAXIMUM SCATTERING ORDER (MAXIMUM VALUE OF
CD                    LEGENDRE EXPANSION INDEX USED IN FILE),
CD      ICHIST        FILE-WIDE FISSION SPECTRUM FLAG
CD                    ICHIST,EQ,0,    NO FILE-WIDE SPECTRUM
CD                    ICHIST,EQ,1,    FILE-WIDE CHI VECTOR
CD                    ICHIST,GT,1,    FILE-WIDE CHI MATRIX
CD      NSCMAX        MAXIMUM NUMBER OF BLOCKS OF SCATTERING DATA
CD      NSBLOK        SUBBLOCKING CONTROL FOR SCATTER MATRICES, THE
CD                    SCATTERING DATA ARE SUBBLOCKED INTO NSBLOK
CD                    RECORDS(SUBBLOCKS) PER SCATTERING BLOCK,
C
C-----

```

```

C-----
CR      FILE DATA      (2D RECORD)
C
CL      (HSETID(I),I=1,12),(HISONM(I),I=1,NISO),
CL      1(CHI(J),J=1,NGROUP),(VEL(J),J=1,NGROUP),
CL      2(EMAX(J),J=1,NGROUP),EMIN,(LOCA(I),I=1,NISO)
C
CW      (NISO+12)*MULT+1+NISO
CW      +NGROUP*(2+ICHIST*(2/(ICHIST+1)))=NUMBER OF WORDS
C
CB      FORMAT(4H 2D ,1H*,11A6,1H*/      HSETID,HISONM
CB      11H*,A6,1H*,9(1X,A6)/(10(1X,A6)))
CB      FORMAT( 6E12,5)      CHI (PRESENT IF ICHIST,EQ,1)
CB      FORMAT ( 6E12,5)      VEL,EMAX,EMIN
CB      FORMAT(12I6)      LOCA
C
CD      HSETID(I)      HOLLERITH IDENTIFICATION OF FILE (A6)
CD      HISONM(I)      HOLLERITH ISOTOPE LABEL FOR ISOTOPE I (A6)
CD      CHI(J)      FILE-WIDE FISSION SPECTRUM(PRESENT IF ICHIST,EQ,1)
CD      VEL(J)      MEAN NEUTRON VELOCITY IN GROUP J (CM/SEC)
CD      EMAX(J)      MAXIMUM ENERGY BOUND OF GROUP J (EV)
CD      EMIN      MINIMUM ENERGY BOUND OF SET (EV)
CD      LOCA(I)      NUMBER OF RECORDS TO BE SKIPPED TO READ DATA FOR
CD                      ISOTOPE I, LOCA(I)=0
C-----

```

```

C-----
CR      FILE-WIDE CHI DATA      (3D RECORD)
C
CC      PRESENT IF ICHIST,GT,1
C
CL      ((CHI(K,J),K=1,ICHIST),J=1,NGROUP),(ISSPEC(I),I=1,NGROUP)
C
CW      NGROUP*(ICHIST+1)=NUMBER OF WORDS
C
CB      FORMAT(4H 3D , 5E12,5/(6E12,5)) CHI
CB      FORMAT(12I6)      ISSPEC
C
CD      CHI(K,J)      FRACTION OF NEUTRONS EMITTED INTO GROUP J AS A
CD                      RESULT OF FISSION IN ANY GROUP,USING SPECTRUM K
CD      ISSPEC(I)      ISSPEC(I)=K IMPLIES THAT SPECTRUM K IS USED
CD                      TO CALCULATE EMISSION SPECTRUM FROM FISSION
CD                      IN GROUP I
C-----

```

```

C-----
CR      ISOTOPE CONTROL AND GROUP INDEPENDENT DATA      (4D RECORD)
C
CL      HABSID,HIDENT,HMAT,AMASS,EFISS,ECAPT,TEMP,SIGPOT,ADENS,KBR,ICHI,
CL      1IFIS,IALF,INP,IN2N,IND,INT,LTOT,LTRN,ISTRPD,
CL      2(IDSCT(N),N=1,NSCMAX),(LORD(N),N=1,NSCMAX),
CL      3(JBAND(J,N),J=1,NGROUP),N=1,NSCMAX),
CL      4(IJJ(J,N),J=1,NGROUP),N=1,NSCMAX)
C
CW      3*MULT+17+NSCMAX*(2*NGROUP+2)=NUMBER OF WORDS
C
CB      FORMAT(4H 4D ,3(1X,A6)/ 6E12,5/
CB      1(12I6))
C
CD      HABSID      HOLLERITH ABSOLUTE ISOTOPE LABEL - SAME FOR ALL
CD                      VERSIONS OF THE SAME ISOTOPE IN FILE (A6)
CD      HIDENT      IDENTIFIER OF LIBRARY FROM WHICH BASIC DATA
CD                      CAME(E,G, ENDF/B) (A6)
CD      HMAT      ISOTOPE IDENTIFICATION (E.G, ENDF/B MAT NO.) (A6)
CD      AMASS      GRAM ATOMIC WEIGHT
CD      EFISS      TOTAL THERMAL ENERGY YIELD/FISSION (W,SEC/FISS)
CD      ECAPT      TOTAL THERMAL ENERGY YIELD/CAPTURE (W,SEC/CAPT)
CD      TEMP      ISOTOPE TEMPERATURE (DEGREES KELVIN)
CD      SIGPOT      AVERAGE EFFECTIVE POTENTIAL SCATTERING IN
CD                      RESONANCE RANGE (BARN/ATOM)

```

```

CD      ADENS      DENSITY OF ISOTOPE IN MIXTURE IN WHICH ISOTOPE
CD      KBR        CROSS SECTIONS WERE GENERATED (A/BARN*CM)
CD      KBR        ISOTOPE CLASSIFICATION
CD      0=UNDEFINED
CD      1=FISSILE
CD      2=FERTILE
CD      3=OTHER ACTINIDE
CD      4=FISSION PRODUCT
CD      5=STRUCTURE
CD      6=COOLANT
CD      7=CONTROL
CD      ICHI      ISOTOPE FISSION SPECTRUM FLAG
CD      ICHI,EQ,0,  USE FILE-WIDE CHI
CD      ICHI,EQ,1,  ISOTOPE CHI VECTOR
CD      ICHI,GT,1,  ISOTOPE CHI MATRIX
CD      IFIS      (N,F) CROSS SECTION FLAG
CD      IFIS=0, NO FISSION DATA IN PRINCIPAL CROSS
CD      SECTION RECORD
CD      =1, FISSION DATA PRESENT IN PRINCIPAL
CD      CROSS SECTION RECORD
CD      1ALF      (N,ALPHA) CROSS SECTION FLAG
CD      SAME OPTIONS AS IFIS
CD      1NP      (N,P) CROSS SECTION FLAG
CD      SAME OPTIONS AS IFIS
CD      1N2N      (N,2N) CROSS SECTION FLAG
CD      SAME OPTIONS AS IFIS
CD      1ND      (N,D) CROSS SECTION FLAG
CD      SAME OPTIONS AS IFIS
CD      1NT      (N,T) CROSS SECTION FLAG
CD      SAME OPTIONS AS IFIS
CD      1TOT      NUMBER OF MOMENTS OF TOTAL CROSS SECTION PROVIDED
CD      IN PRINCIPAL CROSS SECTIONS RECORD
CD      1TRN      NUMBER OF MOMENTS OF TRANSPORT CROSS SECTION
CD      PROVIDED IN PRINCIPAL CROSS SECTIONS RECORD
CD      1STRPD    NUMBER OF COORDINATE DIRECTIONS FOR WHICH
CD      COORDINATE DEPENDENT TRANSPORT CROSS SECTIONS
CD      ARE GIVEN, IF 1STRPD=0, NO COORDINATE DEPENDENT
CD      TRANSPORT CROSS SECTIONS ARE GIVEN,
CD      1DSC(N)   SCATTERING MATRIX TYPE IDENTIFICATION FOR
CD      SCATTERING BLOCK N, SIGNIFICANT ONLY IF
CD      LORD(N),GT,0
CD      1DSC(N)=000 + NN, TOTAL SCATTERING, (SUM OF
CD      ELASTIC, INELASTIC, AND N,2N SCATTERING
CD      MATRIX TERMS),
CD      =100 + NN, ELASTIC SCATTERING
CD      =200 + NN, INELASTIC SCATTERING
CD      =300 + NN, (N,2N) SCATTERING,----SEE
CD      NOTE BELOW----
CD      WHERE NN IS THE LEGENDRE EXPANSION INDEX OF THE
CD      FIRST MATRIX IN BLOCK N
CD      1LORD(N)  NUMBER OF SCATTERING ORDERS IN BLOCK N, IF
CD      LORD(N)=0, THIS BLOCK IS NOT PRESENT FOR THIS
CD      ISOTOPE, IF NN IS THE VALUE TAKEN FROM
CD      1DSC(N), THEN THE MATRICES IN THIS BLOCK
CD      HAVE LEGENDRE EXPANSION INDICES OF NN,NN+1,
CD      NN+2,...,NN+LORD(N)-1
CD      1JBAND(J,N) NUMBER OF GROUPS THAT SCATTER INTO GROUP J,
CD      INCLUDING SELF-SCATTER, IN SCATTERING BLOCK N,
CD      IF JBAND(J,N)=0, NO SCATTER DATA IS PRESENT IN
CD      BLOCK N
CD      1JJ(J,N)  POSITION OF IN-GROUP SCATTERING CROSS SECTION IN
CD      SCATTERING DATA FOR GROUP J, SCATTERING BLOCK
CD      N,COUNTED FROM THE FIRST WORD OF GROUP J DATA,
CD      IF JBAND(J,N).NE.0 THEN 1JJ(J,N) MUST SATISFY
CD      THE RELATION 1,LE,1JJ(J,N),LE,JBAND(J,N)
CD
EN      NOTE- FOR N,2N SCATTER, THE MATRIX CONTAINS TERMS,
CN      SCAT(J TO G), WHICH ARE EMISSION (PRODUCTION)-
CN      BASED, I.E., ARE DEFINED SUCH THAT MACROSCOPIC
CN      SCAT(J TO G) TIMES THE FLUX IN GROUP J GIVES
CN      THE RATE OF EMISSION (PRODUCTION) OF NEUTRONS
CN      INTO GROUP G.
C-----

```

```

C-----
CR          PRINCIPAL CROSS SECTIONS   (50 RECORD)
C
CL          ((STRPL(J,L),J=1,NGROUP),L=1,LTRN),
CL          1((STOTPL(J,L),J=1,NGROUP),L=1,LTOT),(SNGAM(J),J=1,NGROUP),
CL          2(SFIS(J),J=1,NGROUP),(SNUTOT(J),J=1,NGROUP),
CL          3(CHISO(J),J=1,NGROUP),(SNALF(J),J=1,NGROUP),
CL          4(SNP(J),J=1,NGROUP),(SN2N(J),J=1,NGROUP),
CL          5(SND(J),J=1,NGROUP),(SNT(J),J=1,NGROUP)
CL          6((STRPD(J,I),J=1,NGROUP),I=1,ISTRPD)
C
C#          ((1+LTRN+LTOT+IALF+INP+IN2N+IND+INT+ISTRPD+2*IFIS+
CW          ICHI*(2/(ICHI+1)))*NGROUP=NUMBER OF WORDS
C
CB          FORMAT(4H 5D , 5E12,5/(6E12,5)) LENGTH OF LIST AS ABOVE
C
CD          STRPL(J,L)    PL WEIGHTED TRANSPORT CROSS SECTION
CD                        THE FIRST ELEMENT OF ARRAY STRPL IS THE
CD                        CURRENT (P1) WEIGHTED TRANSPORT CROSS SECTION
CD                        THE LEGENDRE EXPANSION COEFFICIENT FACTOR (2L+1)
CD                        IS NOT INCLUDED IN STRPL(J,L),
CD          STOTPL(J,L)  PL WEIGHTED TOTAL CROSS SECTION
CD                        THE FIRST ELEMENT OF ARRAY STOTPL IS THE
CD                        FLUX (P0) WEIGHTED TOTAL CROSS SECTION
CD                        THE LEGENDRE EXPANSION COEFFICIENT FACTOR (2L+1)
CD                        IS NOT INCLUDED IN STOTPL(J,L),
CD          SNGAM(J)      (N,GAMMA)
CD          SFIS(J)        (N,F)      (PRESENT IF IFIS.GT.0)
CD          SNUTOT(J)      TOTAL NEUTRON YIELD/FISS'ON (PRESENT IF IFIS.GT.0)
CD          CHISO(J)       ISOTOPE CHI (PRESENT IF ICHI.EQ.1)
CD          SNALF(J)       (N,ALPHA)  (PRESENT IF IALF.GT.0)
CD          SNP(J)         (N,P)      (PRESENT IF INP.GT.0)
CD          SN2N(J)        (N,2N)     (PRESENT IF IN2N.GT.0)  ----SEE
CD                        NOTE BELOW----
CD          SND(J)         (N,D)      (PRESENT IF IND.GT.0)
CD          SNT(J)         (N,T)      (PRESENT IF INT.GT.0)
CD          STRPD(J,I)     COORDINATE DIRECTION I TRANSPORT CROSS SECTION
CD                        (PRESENT IF ISTRPD.GT.0)
C
CN          NOTE * THE PRINCIPAL N,2N CROSS SECTION SN2N(J)
CN          IS DEFINED AS THE N,2N REACTION CROSS SECTION,
CN          I.E., SUCH THAT MACROSCOPIC SN2N(J) TIMES THE
CN          FLUX IN GROUP J GIVES THE RATE AT WHICH N,2N
CN          REACTIONS OCCUR IN GROUP J,  THUS, FOR N,2N
CN          SCATTERING, SN2N(J) = 0.5*(SUM OF SCAT(J TO G)
CN          SUMMED OVER ALL G),
C-----

```

```

C-----
CR          ISOTOPE CHI DATA   (60 RECORD)
C
CC          PRESENT IF ICHI.GT.1
C
CL          ((CHIISO(K,J),K=1,ICHI),J=1,NGROUP),(ISOPEC(I),I=1,NGROUP)
C
CW          NGROUP*(ICHI+1)=NUMBER OF WORDS
C
CB          FORMAT(4H 6D , 5E12,5/(6E12,5)) CHIISO
CB          FORMAT(12I6)          ISOPEC
C
CD          CHIISO(K,J)    FRACTION OF NEUTRONS EMITTED INTO GROUP J AS A
CD                        RESULT OF FISSION IN ANY GROUP,USING SPECTRUM K
CD          ISOPEC(I)      ISOPEC(I)=K IMPLIES THAT SPECTRUM K IS USED
CD                        TO CALCULATE EMISSION SPECTRUM FROM FISSION
CD                        IN GROUP I
C
C-----

```

```

C-----
CR      SCATTERING SUB=BLOCK      (70 RECORD)
C
CC      PRESENT IF LORD(N).GT.0
C
CL      ((SCAT(K,L),K=1,KMAX),L=1,LORDN)
C
CC      KMAX=SUM OVER J OF JBAND(J,N) WITHIN THE J=GROUP RANGE OF THIS
CC      SUB=BLOCK. IF M IS THE INDEX OF THE SUB=BLOCK, THE J=GROUP
CC      RANGE CONTAINED WITHIN THIS SUB=BLOCK IS
CC      JL=(M-1)*((NGROUP+1)/NSBLOCK+1)+1 TO JU=MIN(NGROUP,JUP),
CC      WHERE JUP=M*((NGROUP+1)/NSBLOCK+1).
C
CC      LORDN=LORD(N)
CC      N IS THE INDEX FOR THE LOOP OVER NSCMAX (SEE FILE STRUCTURE)
C
CM      KMAX=LORDN=NUMBER OF WORDS
C
CB      FORMAT(4H 7D , 5E12,5/(6E12,5))
C
CD      SCAT(K,L)      SCATTERING MATRIX OF SCATTERING ORDER L, FOR
CD                      REACTION TYPE IDENTIFIED BY IDSCT(N) FOR THIS
CD                      BLOCK. JBAND(J,N) VALUES FOR SCATTERING INTO
CD                      GROUP J ARE STORED AT LOCATIONS K=SUM FROM 1
CD                      TO (J-1) OF JBAND(J,N) PLUS 1 TO K=1+JBAND(J,N).
CD                      THE SUM IS ZERO WHEN J=1. J=TO=J SCATTER IS
CD                      THE IJJ(J,N)=TH ENTRY IN THE RANGE JBAND(J,N).
CD                      VALUES ARE STORED IN THE ORDER (J+JUP),
CD                      (J+JUP+1),...,(J+1),J,(J-1),...,(J-JON),
CD                      WHERE JUP=IJJ(J,N)=1 AND JON=JBAND(J,N)-IJJ(J,N).
C-----

```

CEOF

```

C*****
C                      REVISED 11/30/76
C
CF      GRUPXS=IV
CE      MICROSCOPIC GROUP NEUTRON CROSS SECTIONS
C
CN                      THIS FILE PROVIDES A BASIC BROAD GROUP
CN                      LIBRARY, ORDERED BY GROUP
C
C*****

```

```

C-----
CS      FILE STRUCTURE
CS
CS      RECORD TYPE      PRESENT IF
CS      *****
CS      FILE IDENTIFICATION      ALWAYS
CS      FILE CONTROL      ALWAYS
CS      FILE DATA      ALWAYS
CS      SET CHI DATA      ICHIST.GT.1
CS      ISOTOPE CONTROL AND GROUP
CS      INDEPENDENT DATA      ALWAYS
CS      ***** (REPEAT OVER ALL ENERGY GROUPS)
CS      *      PRINCIPAL CROSS SECTIONS      ALWAYS
CS      *****
CS      ***** (REPEAT OVER NISO ISOTOPES)
CS      *      ISOTOPE CHI DATA      ICHI(I).GT.1
CS      *****
CS      ***** (REPEAT OVER ALL ENERGY GROUPS)
CS      *      SCATTERING CONTROL      ALWAYS
CS      *      ***** (REPEAT TO NSCMAX SCATTERING BLOCKS)
CS      * *      ***** (REPEAT FROM 1 TO NSBLOCK)
CS      * * *      SCATTERING SUB=BLOCK      LORD(N).GT.0
CS      *****
C
C-----

```



```

C-----
CR          FILE IDENTIFICATION
C
CL      HNAME,(HUSE(I),I=1,2),IVERS
C
CW      1+3*MULT=NUMBER OF WORDS
C
CD      HNAME          HOLLERITH FILE NAME = GRUPXS = (A6)
CD      HUSE(I)        HOLLERITH USER IDENTIFICATION (A6)
CD      IVERS          FILE VERSION NUMBER
CD      MULT           DOUBLE PRECISION PARAMETER
CD                      1= A6 WORD IS SINGLE WORD
CD                      2= A6 WORD IS DOUBLE PRECISION WORD
C
C-----

```

```

C-----
CR          FILE CONTROL      (1D RECORD)
C
CL      NGROUP,NISO,MAXUP,MAXDN,MAXORD,ICHIST,NPSCS,NSTRPD,NCHIN,NICHI,
CL      NSCMAX,NSBLOK,NRG,NRH
C
CW      14 = NUMBER OF WORDS
C
CD      NGROUP         NUMBER OF ENERGY GROUPS IN SET
CD      NISO           NUMBER OF ISOTOPES IN SET
CD      MAXUP          MAXIMUM NUMBER OF UPSCATTER GROUPS
CD      MAXDN          MAXIMUM NUMBER OF DOWNSCATTER GROUPS
CD      MAXORD         MAXIMUM SCATTERING ORDER (MAXIMUM VALUE OF
CD                      LEGENDRE EXPANSION INDEX USED IN FILE)
CD      ICHIST         SET FISSION SPECTRUM FLAG
CD                      1= SET FISSION SPECTRUM VECTOR
CD                      .GT,1= NUMBER OF SPECTRA FOR RANGES IN INCIDENT-
CD                      ENERGY DEPENDENT CHI
CD      NPSCS          LENGTH OF THE PRINCIPAL CROSS SECTION RECORDS
CD      NSTRPD         NUMBER OF COORDINATE DIRECTIONS FOR WHICH TRANSPORT-
CD                      CROSS SECTIONS ARE GIVEN, ,LE,3
CD      NCHIN          NUMBER OF ISOTOPES FOR WHICH THERE ARE INCIDENT
CD                      ENERGY DEPENDENT CHI DATA
CD      NICHI          NUMBER OF INCIDENT ENERGY SPECTRA FOR ISOTOPE CHI
CD                      DATA (MAXIMUM)
CD      NSCMAX         MAXIMUM NUMBER OF BLOCKS OF SCATTERING DATA
CD                      FOR EACH GROUP SCATTERED INTO * CAN BE THE
CD                      NUMBER OF TYPES OF SCATTERING FOR WHICH THERE
CD                      ARE DATA.
CD      NSBLOK         BLOCKING CONTROL FOR SCATTER MATRICES, THE
CD                      SCATTERING DATA ARE BLOCKED INTO NSBLOK
CD                      RECORDS PER SCATTERING BLOCK.
CD      NRG            RESERVED
CD      NRH            RESERVED
C
C-----

```

```

C-----
CR          FILE DATA      (2D RECORD)
C
CL      (HSETID(I),I=1,12),(HISONM(I),I=1,NISO),
CL      (CHI(J),J=1,NGROUP),(VEL(J),J=1,NGROUP),
CL      (EMAX(J),J=1,NGROUP),EMIN
C
CW      3*NGROUP+MULT*(NISO+12)+1=NUMBER OF WORDS
C
CD      HSETID(I)      HOLLERITH IDENTIFICATION OF SET (A6)
CD      HISONM(I)      HOLLERITH ISOTOPE LABEL FOR ISOTOPE I (A6)
CD      CHI(J)         SET FISSION SPECTRUM
CD      VEL(J)         MEAN NEUTRON VELOCITY IN GROUP J (CM/SEC)
CD      EMAX(J)        MAXIMUM ENERGY BOUND OF GROUP J (EV)
CD      EMIN           MINIMUM ENERGY BOUND OF SET (EV)
C
C-----

```

```

C-----
CR      SET CHI DATA      (30 RECORD)
C
CC      PRESENT IF ICHIST,GT,1
C
CL      ((CHI(K,J),K=1,ICHIST),J=1,NGROUP),(ISSPEC(I),I=1,NGROUP)
C
CW      NGROUP*(ICHIST+1)=NUMBER OF WORDS
C
CD      CHI(K,J)           FRACTION OF NEUTRONS EMITTED IN GROUP J AS A
CD                          RESULT OF FISSION IN ANY GROUP USING SPECTRUM K
CD      ISSPEC(I)          ISSPEC(I)=K IMPLIES THAT SPECTRUM K IS USED TO
CD                          CALCULATE EMISSION SPECTRUM FROM FISSION IN
CD                          GROUP I
C
C-----

```

```

C-----
CR      ISOTOPE CONTROL AND GROUP INDEPENDENT DATA      (40 RECORD)
C
CL      (HABSID(I),I=1,NISO),(HIDENT(I),I=1,NISO),(HMAT(I)=1,NISO),
CL      (AMASS(I),I=1,NISO),
CL      (EFISS(I),I=1,NISO),(ECAPT(I),I=1,NISO),(XN(I),I=1,NISO),
CL      (TEMP(I),I=1,NISO),(SIGPOT(I),I=1,NISO),(ADENS(I),I=1,NISO),
CL      (KBR(I),I=1,NISO),(IDSC(T(N),N=1,NSCMAX),(LORD(N),N=1,NSCMAX),
CL      (ICHI(I),I=1,NISO),IALF,INP,IN2N,IND,INT,IX,IY
C
CW      NISO*(3*MULT+9)+2*NSCMAX+7=NUMBER OF WORDS
C
CD      HABSID(I)          HOLLERITH ABSOLUTE ISOTOPE LABEL (A6)
CD      HIDENT            IDENTIFIER, USUALLY ENDF NUMBER (A6)
CD      HMAT(I)            MATERIAL REFERENCE FOR IDENTIFICATION (A6)
CD      AMASS(I)           GRAM ATOMIC WEIGHT
CD      EFISS(I)           TOTAL THERMAL ENERGY YIELD/FISSION (W,SEC/FISS)
CD      ECAPT(I)           THERMAL ENERGY YIELD PER CAPTURE (W,SEC/CAP)
CD      XN(I)              RESERVED
CD      TEMP(I)            ISOTOPE TEMPERATURE (DEGREES KELVIN)
CD      SIGPOT(I)          AVERAGE EFFECTIVE POTENTIAL SCATTERING IN
CD                          RESONANCE RANGE (BARN/ATOM)
CD      ADENS(I)           DENSITY OF ISOTOPE IN MIXTURE IN WHICH ISOTOPE
CD                          CROSS SECTIONS WERE GENERATED (A/BARN/CC)
CD      KBR(I)             ISOTOPE CLASSIFICATION
CD                          0=UNDEFINED
CD                          1=FISSILE
CD                          2=FERTILE
CD                          3=OTHER ACTINIDE
CD                          4=FISSION PRODUCT
CD                          5=STRUCTURE
CD                          6=COOLANT
CD                          7=CONTROL
CD      IDSC(T(N)          SCATTERING MATRIX TYPE IDENTIFICATION FOR
CD                          SCATTERING BLOCK N, SIGNIFICANT ONLY IF
CD                          LORD(N),GT,0
CD                          IDSC(T(N)=000+ NN, TOTAL SCATTERING(SUM OF
CD                          ELASTIC, INELASTIC, AND N, 2N SCATTERING)
CD                          =100 + NN, ELASTIC SCATTERING
CD                          =200 + NN, INELASTIC SCATTERING
CD                          =300 + NN, (N,2N) SCATTERING,---SEE
CD                          NOTE BELOW---
CD                          WHERE NN IS EXPANSION ORDER OF BLOCK N DATA,
CD                          DATA ORDERED BY INCREASING EXPANSION ORDER
CD                          FOR EACH OF THE TYPES IN THE ABOVE ORDER
CD      LORD(N)            NUMBER OF SCATTERING ORDERS IN BLOCK N, IF
CD                          LORD(N)=0, THIS BLOCK IS NOT PRESENT FOR THIS
CD                          ISOTOPE BLOCK, IF NN IS THE VALUE TAKEN FROM
CD                          IDSC(T(N), THEN THE MATRICES IN THIS BLOCK
CD                          HAVE LEGENDRE EXPANSION INDICES OF NN,NN+1,
CD                          ....LORD(N)
C

```

```

CD      ICHI(I)      ISOTOPE FISSION SPECTRUM FLAG
CD                        0= USE SET CHI
CD                        1= ISOTOPE CHI VECTOR
CD                        .GT.1= NUMBER OF SPECTRA FOR RANGES IN INCIDENT
CD                        ENERGY DEPENDENT CHI
CD      IALF          (N,ALPHA) CROSS SECTION FLAG
CD                        IALF=1, (N,ALPHA) DATA IN THE FILE
CD                        IALF=0, NO (N,ALPHA) DATA IN THE FILE
CD      INP           (N,P) CROSS SECTION FLAG
CD                        SAME OPTIONS AS FOR IALF
CD      IN2N          (N,2N) CROSS SECTION FLAG
CD                        SAME OPTIONS AS FOR IALF
CD      IND           (N,D) CROSS SECTION FLAG
CD                        SAME OPTIONS AS FOR IALF
CD      INT           (N,T) CROSS SECTION FLAG
CD                        SAME OPTIONS AS FOR IALF
CD      IX            RESERVED
CD      IY            RESERVED
C
CN      NOTE= FOR N,2N SCATTER, THE MATRIX CONTAINS TERMS,
CN      SS(G TO J), WHICH ARE EMISSION (PRODUCTION)-
CN      BASED, I.E., ARE DEFINED SUCH THAT MACROSCOPIC
CN      SS(G TO J) TIMES THE FLUX IN GROUP G GIVES THE
CN      RATE OF EMISSION (PRODUCTION) OF NEUTRONS IN
CN      GROUP J.
C-----

```

```

C-----
CR      PRINCIPAL CROSS SECTIONS      (5D RECORD)
C
CL      ((STRPL(I,L),I=1,NISO),L=1,MAXT),
CL      ((STOTPL(I,L),I=1,NISO),L=1,MAXT),(SNGAM(I),I=1,NISO),
CL      (SFIS(I),I=1,NISO),(SNUTOT(I),I=1,NISO),
CL      (CHISO(I),I=1,NISO),(SNALF(I),I=1,NISO),
CL      (SNP(I),I=1,NISO),(SN2N(I),I=1,NISO),
CL      (SND(I),I=1,NISO),(SNT(I),I=1,NISO),
CL      ((STRPD(I,J),I=1,NISO),J=1,NSTRPD)
C
CN      NPSCS=NISO*(P*MAXT+4*IALF+INP+IN2N+IND+INT+NSTRPD)=NUMBER
CN      OF WORDS
C
CD      STRPL(I,L)     PL WEIGHTED TRANSPORT CROSS SECTIONS
CD                        THE LEGENDRE EXPANSION COEFFICIENT FACTOR (2L+1)
CD                        IS NOT INCLUDED IN STRPL(I,L).
CD      STOTPL(I,L)    PL WEIGHTED TOTAL CROSS SECTIONS
CD                        THE LEGENDRE EXPANSION COEFFICIENT FACTOR (2L+1)
CD                        IS NOT INCLUDED IN STOTPL(I,L).
CD      SNGAM(I)        (N,GAMMA) CROSS SECTION
CD      SFIS(I)         (N,FISSION) CROSS SECTION
CD      SNUTOT(I)       TOTAL NEUTRON YIELD/FISSION
CD      CHISO(I)        ISOTOPE CHI VECTOR
CD      SNALF(I)        (N,ALPHA) (PRESENT IF IALF.GT.0)
CD      SNP(I)          (N,P) (PRESENT IF INP.GT.0)
CD      SN2N(I)         (N,2N) (PRESENT IF IN2N.GT.0) ----SEE
CD                        NOTE BELOW----
CD      SND(I)          (N,D) (PRESENT IF IND.GT.0)
CD      SNT(I)          (N,T) (PRESENT IF INT.GT.0)
CD      STRPD(I,J)      COORDINATE DIRECTION J DEPENDENT TRANSPORT CROSS
CD                        SECTION( PRESENT IF NSTRPD.GT.0)
CD      MAXT            MAXORD+1
C
CN      NOTE = THE PRINCIPAL N,2N CROSS SECTION SN2N FOR
CN      A GIVEN GROUP IS THE N,2N REACTION CROSS SECTION
CN      FOR THAT GROUP, I.E., SUCH THAT MACROSCOPIC
CN      SN2N FOR A GIVEN GROUP TIMES THE GROUP FLUX
CN      GIVES THE RATE AT WHICH N,2N REACTIONS OCCUR
CN      IN THAT GROUP.
C-----

```

```

C-----
CR      ISOTOPE CHI DATA   (6D RECORD)
C
CC      PRESENT IF NCHIN,GT,0 AND ICHI(I),GT,1
C
CL      ((CHISI(K,J),K=1,ICHII,J=1,NGROUP),(ISOPEC(I),I=1,NGROUP)
C
CW      NGROUP*(ICHII + 1)=NUMBER OF WORDS
C
CD      ICHII      =ICHI(I)
CD      CHISI(K,J)  FRACTION OF NEUTRONS EMITTED IN GROUP J AS A RESULT
CD                  OF FISSION IN ANY GROUP USING SPECTRUM K
CD      ISOPEC(I)   ISOPEC(I)*K IMPLIES THAT SPECTRUM K IS USED TO
CD                  CALCULATE EMISSION SPECTRUM FROM FISSION IN
CD                  GROUP I
C-----

```

```

C-----
CR      SCATTERING CONTROL   (7D RECORD)
C
CL      ((JBAND(I,N),I=1,NISO),N=1,NSCMAX)
CL      ((IJJ(I,N),I=1,NISO), N=1,NSCMAX)
C
CW      2*NISO*NSCMAX=NUMBER OF WORDS
C
CD      JBAND(I,N)   NUMBER OF GROUPS THAT SCATTER INTO THE GROUP,
CD                  INCLUDING SELF SCATTER, FOR ISOTOPE I, BLOCK N
CD                  IF JBAND=0, NO SCATTER DATA IS PRESENT IN THAT
CD                  BLOCK.
CD      IJJ(I,N)     POSITION OF JN-GROUP SCATTERING CROSS SECTION IN
CD                  SCATTERING DATA FOR ISOTOPE I, SCATTERING BLOCK
CD                  N, COUNTED FROM THE FIRST WORD OF ISOTOPE I DATA
CD                  IF JBAND(I,N),NE,0 THEN IJJ(I,N) MUST SATISFY
CD                  THE RELATION 1,LE,IJJ,LE,JBAND.
C-----

```

```

C-----
CR      SCATTERING SUB-BLOCK   (8D RECORD)
C
CC      PRESENT IF LORD(N),GT,0
C
CL      ((SCAT(K,L),K=1,KMAX),L=1,LORDN)
C
CC      KMAX=SUM OVER I OF JBAND(I,N) WITHIN THE I-ISOTOPE RANGE OF THIS
CC      SUB-BLOCK. IF M IS THE INDEX OF THE SUB-BLOCK, THE I-ISOTOPE
CC      RANGE CONTAINED WITHIN THIS SUB-BLOCK IS
CC      I=(M-1)*((NISO-1)/NSBLOK + 1)+1 TO IUP=MIN0(NISO,IUP),
CC      WHERE IUP=M*((NISO-1)/NSBLOK + 1).
C
CC      LORDN = LORD(N)
CC      N IS THE INDEX FOR THE LOOP OVER NSCMAX (SEE FILE STRUCTURE)
C
CW      KMAX=LORDN=NUMBER OF WORDS
C
CD      SCAT(K,L)    SCATTERING MATRIX OF SCATTERING ORDER L, FOR
CD                  REACTION TYPE IDENTIFIED BY IDSCT(N) FOR THIS
CD                  BLOCK. IJJ(I) VALUES FOR SCATTERING INTO
CD                  GROUP J ARE STORED AT LOCATIONS K=SUM FROM 1
CD                  TO (I-1) OF JBAND(I) PLUS 1 TO K=I+JBAND(I).
CD                  THE SUM IS ZERO WHEN I=1. J=TO=J SCATTER IS
CD                  THE IJJ(I)-TH ENTRY IN THE RANGE JBAND(I).
CD                  VALUES ARE STORED IN THE ORDER (J+JUP),
CD                  (J+JUP+1),...,(J+1),J,(J-1),...,(J-JDN),
CD                  WHERE JUP=IJJ(I)-1 AND JDN=JBAND(I)-IJJ(I)
C-----

```

```

C
CN      BLOCKING OVER ISOTOPES FOR EACH ORDER IS
CN      NECESSARY IF THE DATA IS TO BE PROCESSED ONCE
CN      SEQUENTIALLY AND THE MACROSCOPIC DATA STORED
CN      ONE ORDER AT A TIME, IN THIS EVENT, EITHER
CN      VALUES OF LORD(N) MUST BE LIMITED TO 1 OR
CN      NSBLOK NE 1,
C
C-----
CEOF

```

APPENDIX B FUNCTIONS OF I2G SUBROUTINES

Table B-I describes the names, functions, and calling routines of all I2G code routines and subroutines.

TABLE B-I
FUNCTIONS OF I2G SUBROUTINES

<u>Routine Name</u>	<u>Functional Description</u>	<u>Calling Routine(s)</u>
I2GDRV	Driver program of code. Performs initializations and calls the file processing routines.	-
I2GPIR	Program initialization routine. Calculates pointers for all fast-core arrays.	I2GDRV
FILL5D	Performs a partial conversion of principal cross sections.	I2GDRV
WRITID	Writes GRUPXS file identification record.	I2GPIR
WRIT1D	Writes GRUPXS file control record.	I2GDRV
WRIT2D	Writes GRUPXS file data record.	I2GDRV
WRIT3D	Writes GRUPXS file-wide chi data record.	I2GDRV
WRIT4D	Writes GRUPXS isotope control and group independent data record.	I2GDRV
WRIT5D	Performs final conversion of and writes GRUPXS principal cross sections records.	I2GDRV
WRIT6D	Writes GRUPXS isotope chi data.	I2GDRV
SCATER	Performs partial conversion of scattering cross sections when NGROUP.GT.NSBLOK.	I2GDRV
SCTLCM	Performs extended core blocking of partially converted scattering cross sections.	SCATER
SCATLG	Performs partial conversion of scattering cross sections when NGROUP.EQ.NSBLOK.	I2GDRV

TABLE B-I (cont)

<u>Routine Name</u>	<u>Functional Description</u>	<u>Calling Routine(s)</u>
SCTLGG	Performs extended-core blocking of partially converted scattering cross sections.	SCATLG
WRIT7D	Computes and writes GRUPXS scattering control record.	SCATER, SCATLG
SORTBL	Computes scattering bandwidths and positions of in-group scattering.	WRIT7D
TBLPOS	Computes table position of in-group scattering	SORTBL
WRIT8D	Completes conversion of and writes GRUPXS scattering cross-section records.	WRIT7D
TIMER	Computes I2G code execution time.	I2GDRV
SEEK	Locates and assigns logical unit numbers for all files in I2G.	I2GDRV
RITE	Writes records onto a sequential disk file from logical unit assigned by SEEK.	All Routines
REED	Reads records from a sequential disk file from logical unit number assigned by SEEK.	All Routines
DOPC	Defines disk space for use in extended core - random disk transfers.	I2GDRV, FILL5D SCATER, SCATLG
DRED	Performs block transfers from disk unit defined by DOPC into extended core.	Most Routines
DRIT	Performs block transfers from extended core into the disk unit defined by DOPC.	Most Routines
CRED	Performs string transfers from extended core into fast-core memory arrays.	Most Routines
CRIT	Performs string transfers from fast core into extended-core memory.	Most Routines