

LA-9303-M, Vol. II (ENDF-324)

Manual

CIC-14 REPORT COLLECTION

REPRODUCTION
COPY

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36.

*The NJOY Nuclear Data Processing System,
Volume II: The NJOY, RECONR, BROADR,
HEATR, and THERMR Modules*



Los Alamos Los Alamos National Laboratory
Los Alamos, New Mexico 87545

This work was supported by the US Department of Energy, Division of Reactor Research and Technology, and the Electric Power Research Institute.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of 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. References herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

LA-9303-M, Vol. II (ENDF-324)
Manual

Issued: May 1982

**The NJOY Nuclear Data Processing System,
Volume II: The NJOY, RECONR, BROADR,
HEATR, and THERMR Modules**

R. E. MacFarlane
D. W. Muir
R. M. Boicourt



Los Alamos Los Alamos National Laboratory
Los Alamos, New Mexico 87545

THE NJOY NUCLEAR DATA PROCESSING SYSTEM, VOLUME II:
THE NJOY, RECONR, BROADR, HEATR, AND THERMR MODULES

by

R. E. MacFarlane, D. W. Muir, and R. M. Boicourt

ABSTRACT

The NJOY nuclear data processing system is a comprehensive computer code package for producing cross sections and related nuclear parameters from ENDF/B evaluated nuclear data. This volume provides detailed descriptions of the NJOY module, which contains the executive program and utility subroutines used by the other modules, and it discusses the theory and computational methods of four of the modules used for producing pointwise cross sections: RECONR, BROADR, HEATR, and THERMR.

VIII. INTRODUCTION TO VOLUME II

The NJOY nuclear data processing system is a comprehensive computer code package for producing pointwise and multigroup cross sections from ENDF/B-IV and -V evaluated nuclear data. A concise description of the code system and references to the ancestors of NJOY are given in Vol. I of this report. This volume provides more detailed discussions of the theory and methods used in four of the modules that prepare pointwise cross-section data. It also describes the executive program that controls the order of execution of the various modules, and it discusses the library of utility routines that are available to all of the processing modules.

NJOY is a very modular system. In fact, each module is essentially a free-standing code. The organization of this report reflects the structure. Each module is described in a separate chapter. In order to allow for easy revision,

each chapter uses independent numbering of figures, tables, equations, and pages, and each chapter contains its own references.

The next chapter describes the overall structure of the NJOY system, the executive program, and the utility subroutines available to the processing modules. This is followed by chapters describing four of the modules that produce "pointwise" ENDF (PENDF) libraries. RECONR reconstructs pointwise cross sections from ENDF resonance parameters and interpolation laws, BROADR Doppler broadens these cross sections to any desired temperature, HEATR generates heat and radiation damage production cross sections, and THERMR adds elastic and inelastic thermal cross sections for free and bound scatterers.

IX. NJOY

The modular structure of NJOY is shown in Fig. 1. The term "module" is used here in a very restrictive sense: a module is a block of coding that communicates with other modules only through logical units (the terms "tape" and "file" will be used interchangeably in this report). This means that every module is essentially a freestanding program. Figure 1 illustrates the "overlay" version of the code. Here the NJOY level consists of a simple executive program for linking modules together and a set of utility subroutines available to all modules. Other structures are possible. For example, the linking of modules could be handled by the normal sequencing capabilities of the operating system; the NJOY utilities would then be made available to the loader as a relocatable library. The restrictive definition of the term "module" used here makes it possible to choose whichever of these two configurations is most suitable for a particular operating system, makes it easy to add new modules, and protects a module against changes or repairs in another module.

A. The Executive Program

This is the main program of the NJOY system. It simply reads a module name in free format and calls in the requested module. The first card read by any module contains the unit numbers for the various input and output files. In this way, the output of one module can be assigned to be the input of another module, thereby linking the modules to perform the desired processing task. Table I gives an example of the linking procedure.

NJOY

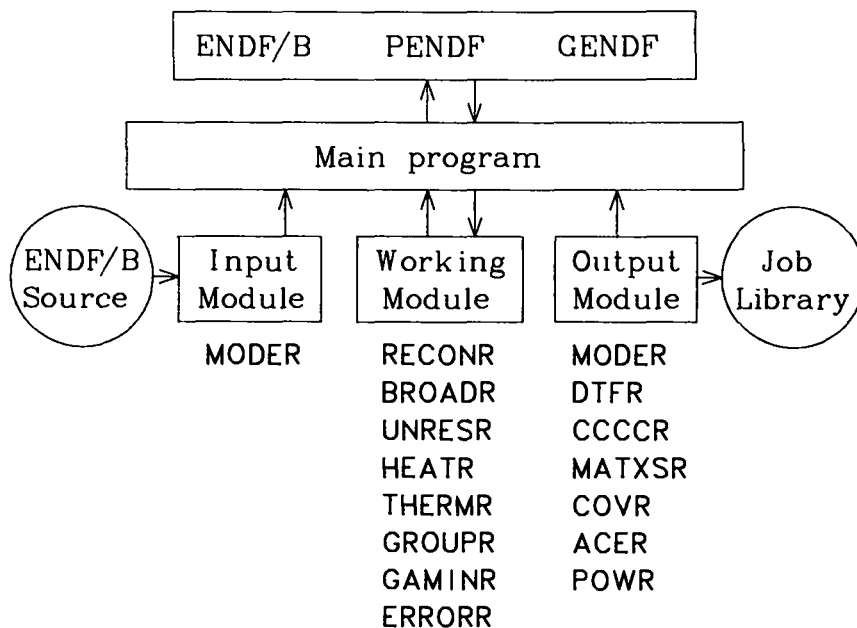


Fig. 1.
Basic structure of the NJOY code (overlay configuration).

TABLE I

EXAMPLES OF LINKING MODULES TOGETHER IN THE OVERLAY CONFIGURATION

```

    [mount an ENDF/B tape on unit 20]
    0
    5
    *RECONR*
    20 21
    [input lines for RECONR]
    *GROUPT*
    20 21 0 22
    [input lines for GROUPT]
    *DTFR*
    22 23 21
    [input lines for DTFR]
    *STOP*
    [DTF-format card images written on unit 23]
  
```

The main program also sets the page length (NPAGE) for blocked binary files (see below) and assigns the unit numbers for system input and output. NJOY expects these numbers to be less than 10 (the normal choice is 5 for input and 6 for output). In a time-sharing environment, it is often helpful to have a short print for the terminal while still preserving the long listing for the system printer. Such an option is provided by IOPT=1. This option changes the input and output (NSHORT) to unit 7, which can be equivalenced to the terminal (TTY). The final common parameter is IVERF, which should be 4 to process ENDF/B-IV evaluations and 5 for ENDF/B-V.

The input instructions for the NJOY module are given as comment cards at the beginning of the module. They are reproduced here for the convenience of the user (see also Vol. I: User's Manual).

```

C *
C *---INPUT SPECIFICATIONS (FREE FORMAT)-----*
C *
C * CARD 1 INPUT OPTION *
C * IOPT 0 FOR CARD INPUT AND FULL OUTPUT *
C * 1 FOR TERMINAL INPUT WITH SHORT OUTPUT ON TERMINAL *
C * CARD 2 *
C * IVERF ENDF/B VERSION NUMBER (4 OR 5 ONLY) *
C * CARD 3 *
C * MOPT SIX CHARACTER MODULE NAME DELIMITED WITH *
C * E.G., *RECONR* (ONLY FIRST FOUR CHARACTERS ARE *
C * USED. REPEAT FOR EACH MODULE DESIRED). USE *
C * *STOP* TO TERMINATE PROGRAM. *
C *
C * SEE THE COMMENTS AT THE START OF EACH MODULE FOR *
C * ITS SPECIFIC INPUT INSTRUCTIONS. *
C *

```

The example in Table I clarifies their use.

B. Interface Files

Another requirement of a good modular system is that the input and output files be in a common format so that modules can work with each other's output in a flexible way. Since NJOY is basically an ENDF/B processing code, ENDF/B-compatible formats (see Sec. X.D) were chosen for linking modules together. "Input" and "output" modules (see Fig. 1) can be specified to communicate with other formats (the "outside world"). However, if the user desires, the RECONR PENDF tape can be run through BROADR to produce a new Doppler-broadened PENDF

tape for GROUPT. Many other combinations are possible. These common-format files also provide for convenient restarts at many points in the calculational sequence. For example, if a user is trying to produce pointwise cross sections at 300 K, 600 K, and 900 K and runs out of time while working on 900 K, he can save the partially completed PENDF tape and restart from 600 K. Multigroup modules use specially constructed groupwise ENDF formats (GENDF) that are compatible with the multigroup output modules. A GENDF tape from GROUPT can be saved in the NJOY data library, run through CCCC to produce one output format, and then run through MATXS for another output format.

In NJOY, unit numbers from 20 through 99 are used for storing results or linking modules, units 10 through 19 are reserved for scratch files, which can be destroyed after a module has completed its job, and units below 9 are reserved for the system.

There are special utility routines to open, close, and reposition files. These routines can be modified to adapt NJOY to a particular operating system.

OPENZ(LUN,NEW)

Open the unit = ABS(LUN). If LUN > 0, use coded (formatted) mode, and if LUN < 0, use binary mode. Destroy on close or job termination if $10 \leq \text{LUN} < 20$. If NEW = 1, destroy the file on this unit (if it exists) and open a new file.

CLOSZ(LUN)

Close the file with unit = ABS(LUN).

REPOZ(LUN)

Reposition (rewind) the unit = ABS(LUN).

SKIPRZ(LUN,NREC)

Skip NREC records forward or backwards.

Caution: Some systems have a call for this option; others can use loops of backspace and dummy reads as given in the NJOY code. Both these operations work well for systems that use "linked-list" data structures for I/O files. On some systems, however, backspace is implemented as a rewind followed by forward dummy reads to the desired location. In such cases (for example, VAX), SKIPRZ must be recoded to avoid calling BACKSPACE repeatedly.

This strategy is similar to the approach standardized for FORTRAN-77. If callable open and close operations (or the equivalent) are not available on the target system, a fixed set of units can be defined on a program card (CDC) or job control deck (IBM), and these routines can be replaced with versions that simply return to the calling program. Caution: some units may be used for binary I/O in one part of the program and coded (formatted) I/O in another.

C. Free-format Input

For a card-input program, free-form input is convenient, but in a time-sharing environment, it is almost essential. Therefore, a subroutine FREE has been included among the NJOY utilities to provide a simple free-format input capability. This routine contains a machine-dependent subroutine PACK, that may have to be adapted to local conditions.

FREE(NIN,Z,NZA,NCW)

NIN input logical unit containing free-format card images

Z(I) dimensioned variable containing numbers decoded from input cards

NZA on call, number of words desired
on return, number of words found

NCW number of Hollerith characters to be loaded in each word, blank fill to right

All numbers read from the input cards are returned as real in Z. The calling program can convert selected numbers to integer mode as required. Hollerith variables are returned in integer form using the internal N-bit code of the machine. If NCW is larger than the number of characters per word, successive locations of Z will be used.

Fields on the input cards are delimited by any character not used for another purpose (+,-,number,E,H,*,R,/). For exponent fields, the E must be present, and spaces are not allowed before the E. Decimal points are not required after numbers. Hollerith fields may use nHstring or *string*. The character / terminates the input for one call to FREE (it may involve more than one card) leaving any unread variables unchanged. This feature is often used to default variables from the right. The nR specification causes the number following R to be repeated n times. Some input examples follow.

<u>legal</u>			<u>illegal</u>	
12	12.	1.2E1	1.2+1	1.2 E1
U235		4HU235	4RU235	(does not right-justify)
5R1.0		3R1.1E6		

Other examples will be found in input samples throughout this report.

FREE contains several parameters that may have to be changed when converting between different machines: NBPC is the number of bits per character for Hollerith data (6 on CDC, 8 on IBM), MACHWD is the number of Hollerith characters in a machine word (10 on CDC, 4 on IBM), and RNDOFF is a constant that should be approximately equivalent to one bit in the last place for the target machine. The rest of the machine dependence is incorporated into FUNCTION PACK, which inserts characters into words. Two versions are supplied: one is based on masking for CDC machines, and the other uses one-byte variables and equivalent for IBM systems.

D. ENDF Input-Output

The ENDF/B evaluated nuclear data files are well documented elsewhere,¹ but for the convenience of the reader, some features of the format will be described here.

ENDF/B "tapes" are subdivided internally into "materials" (MAT), "files" (MF), and "sections" (MT). A MAT contains all data for a particular evaluation for an element or isotope (for example, MAT1276 is an evaluation for 8-0-16). A "file" contains a particular type of data for that MAT: MF=3 is cross-section versus energy data; MF=15 contains secondary photon energy distributions. A "section" refers to a particular reaction [for example, MT=2 is elastic scattering and MT=107 is the (n, α) reaction]. Every record contains the current MAT, MF, and MT values. Two materials are separated by a record with MAT=0 (the material-end or MEND record). Two files are separated by a record with MF=0 (the file-end or FEND record). Two sections are separated by a record with MT=0 (the section-end or SEND record). Finally, the tape is terminated with a record with MAT=-1 (tape-end or TEND record).

NJOY has a set of utility subroutines for locating desired positions on an ENDF tape.

FINDF(MAT,MF,MT,NIN)

Search NIN backward or forward for the first record with this
MAT,MF,MT

TOSEND(NIN,NOUT1,NOUT2,A)

TOFEND(NIN,NOUT1,NOUT2,A)

TOMEND(NIN,NOUT1,NOUT2,A)

TOTEND(NIN,NOUT1,NOUT2,A)

Skip forward past the next SEND, FEND, MEND, or TEND Card on NIN. If
NOUT1 and/or NOUT2 are nonzero, copy the records. Input and output
files must be in the same mode.

The data on an ENDF tape are written in 7 different kinds of "structures",
each of which has a binary and a formatted form (the words "coded", "formatted",
and "BCD" will often be used interchangeably even though the actual representa-
tion might be ASCII or display code). The structures are: (1) TAPEID, a
Hollerith title for the tape; (2) CONT, a control record (includes SEND, FEND,
MEND, and TEND); (3) LIST, a list of data items; (4) HOLL, a list of Hollerith
words; (5) TAB1, a one-dimensional tabulation of data pairs; (6) TAB2, a two-
dimensional tabulation control record; and (7) DICT, an index ("dictionary") to
the sections found in the MAT. It should be noted that HOLL is a special case
of LIST and DICT is a special case of CONT.

In binary mode, each "structure" is written as a single logical record as
follows:

TAPEID[MAT,MF,MT/A(I),I=1,17]*

where MAT=tape number, MF=MT=0, and the Hollerith data are 16A4,A2;

CONT[MAT,MF,MT/C1,C2,L1,L2,N1,N2];

LIST[MAT,MF,MT/C1,C2,L1,L2,N1,N2/
A(I),I=1,N1];

HOLL[MAT,MF,MT/C1,C2,L1,L2,N1,N2/
A(I),I=1,N1];

* In ENDF/B manuals, the slash is used as a logical divider. Replace it with a
comma and add parentheses when constructing a FORTRAN I/O list.

where MF=1, MT=451, and each line of Hollerith characters is stored in A as 16A4,A2;

```
TAB1[MAT,MF,MT/C1,C2,L1,L2,N1,N2/  
      NBT(I),JNT(I),I=1,N1/  
      X(I),Y(I),I=1,N2],
```

where NBT and JNT are the interpolation table and Y(X) is the one-dimensional tabulation;

```
TAB2[MAT,MF,MT/C1,C2,L1,L2,N1,N2/  
      NBT(I),JNT(I),I=1,N1],
```

where the interpolation table is to be used to control a series of N2 LIST or TAB1 structures that follow; and

```
DICT[MAT,MF,MT/0.,0.,MFS,MTS,NCS,MODS],
```

where there is a record for each section in the material (MFS,MTS) giving the card count (NCS) for that section. For ENDF/B-V, MODS indicates the revision number for that section.

The ENDF/B procedure manual¹ explains how these structures are combined to represent various physical quantities.

In order to make these records practical, limits have been established that keep the record length below approximately 10 000 words. In BCD mode, each "structure" is broken up into many card images, each containing 6 data words, followed by MAT, MF, MT, and a line sequence number. There is no intrinsic limit to the length of a data structure written in BCD form because a program reading the data can normally be coded to use the data in "pages" of reasonable size. The MINX code² was forced to use BCD formats to handle the large tabulations found on PENDF tapes. Analysis shows that this code uses more than 50% of its running time coding and decoding BCD formats. In order to eliminate this waste, a "blocked binary" format has been developed for the ENDF/B data structures. A structure is divided up into several logical records of intermediate length (typically about 300 words), each having the following form:

```
[MAT,MF,MT,NB,NW/A(I),I=1,NW],
```

where NB is the number of words remaining in the data structure (the last record has NB=0). This type of record is compatible with the official ENDF binary record, but is also adaptable to paging methods. The page size can be chosen to optimize input/output rates for a particular computer system.

A set of utility subroutines has been devised to handle both blocked-binary and paged-BCD input and output.

TPIDIO(NIN,NOUT,NSCR,A,NB,NW)

Read/write the Hollerith tape identification record to/from array A (NB=0,NW=17).

CONTIO(NIN,NOUT,NSCR,A,NB,NW)

Read/write a control record to/from A (NB=0,NW=6). Uses ACONT for END cards.

ACONT(NOUT,NSCR)

Write an end record on the desired units.

LISTIO(NIN,NOUT,NSCR,A,NB,NW)

Read/write the first record or page of a list record to/from A. If NB is not zero, continue with MOREIO, as illustrated in Examples 1 and 2 below.

HOLLIO(NIN,NOUT,NSCR,A,NB,NW)

Read/write the first record or page of the Hollerith descriptive data (MF1,MT451) to/from A, taking account of the 16A4,A2 format needed in BCD mode. If NB is not zero, use MOREIO.

TAB1IO(NIN,NOUT,NSCR,A,NB,NW)

Read/write the first record or page of a TAB1 structure. If NB is not zero, use MOREIO.

TAB2IO(NIN,NOUT,NSCR,A,NB,NW)

Read/write a TAB2 structure (NB=0).

DICTIO(NIN,NOUT,NSCR,A,NB,NW)

Read/write the entire material dictionary (really an index) to/from A. On entry, NW is the number of entries in the dictionary. MOREIO is not used.

MOREIO(NIN,NOUT,NSCR,A,NB,NW)

Read/write continuation records or pages to/from the array A. Returns NB=0 after processing the last record or page.

CXFP(X,F,S,N)

This routine is used by some of the other ENDF/B routines to prepare formatted output without the normal FORTRAN "E". Floating-point numbers are output as $\pm 1.23456 \pm NN$ or $\pm 1.234567 \pm N$, depending on the size of the exponent.

In these calling sequences, the unit numbers can be positive, negative, or zero. Positive numbers mean BCD mode, negative numbers mean blocked-binary mode, and zero means the file corresponding to this position in the calling sequence is not used. All of these routines use one area of labelled common

COMMON/CONT/C1,C2,L1,L2,N1,N2,MAT,MF,MT,NS,NSP,NSC

where C1 through MT have their usual ENDF meanings, NSP is the sequence number for NIN, NS is the sequence number of NOUT, and NSC is the sequence number for NSCR. Two examples may help to make clear the use of these routines.

Example 1. Read All Data

```
LOC=1
CALL TAB1IO(NIN,0,0,A(1),NB,NW)
10 IF (NB.EQ.0) GO TO 20
   LOC=LOC+NW
   CALL MOREIO(NIN,0,0,A(LOC),NB,NW)
   GO TO 10
20 [process data in A]
```

Example 2. Paging

```
CALL TAB1IO(NIN,0,0,A(1),NB,NW)
10 [process this page of data in A]
   IF (NB.EQ.0) GO TO 20
   CALL MOREIO(NIN,0,0,A(1),NB,NW)
   GO TO 10
20 CONTINUE
```

When NIN is BCD, paging is automatic. Positive and negative unit numbers can be mixed in TPIDIO, CONTIO, LISTIO, etc., when mode conversion is desired.

The advantages of the blocked-binary mode are demonstrated in Table II for several characteristic processing tasks.

E. Buffered Binary Scratch Storage

During the execution of a program, there are often times when large amounts of data need to be stored in mass storage temporarily. In order to make such

scratch storage as efficient as possible, NJOY includes a pair of utility sub-routines that automatically buffer such data through fast memory to disk and/or large core memory (LCM).

LOADA(I,A,NA,NTAPE,BUF,NBUF)

FINDA(I,A,NA,NTAPE,BUF,NBUF)

where

I = data point number (I must increase, except I=1 causes a rewind and I<0 flushes the fast memory buffer to mass storage)

A = array containing data to be stored or destination of data to be read

NA = number of words to be transmitted (must be the same for all I)

NTAPE = logical unit number of disk file

BUF = fast-memory buffer array

NBUF = length of buffer array

When a point is to be saved, LOADA stores it in BUF. When BUF becomes full, it is automatically dumped to disk. When a point is to be retrieved, FINDA checks to see whether the desired point is in BUF. If not, it reads through the disk until the desired point is in memory. It then returns the desired point. When NA is small, using LOADA/FINDA reduces the number of I/O operations dramatically.

TABLE II

EXAMPLES OF EFFICIENCY GAINS OBTAINED WITH BLOCKED-BINARY INPUT

<u>Test</u>	<u>BCD</u>	<u>BB</u>
²³⁵ U Doppler broadening	169.	72.1
²³⁵ U P ₃ elastic matrix	10.9	4.99
²³⁵ U (n,2n) matrix	4.51	.838
Iron Doppler broadening	139.	46.5

Sometimes it is necessary to find a particular part of the buffered data. In such cases, use

SCANA(E,IP,NP,NA,NTAPE,BUF,NBUF)

where E is a value for the first of the NA words, and IP points to part of the data whose first word is either equal to E or is the first value less than E.

F. Dynamic Storage Allocation

In many large computer codes, storage requirements may change continually throughout the execution of a problem. If maximum use is to be made of the available memory, it is necessary to reallocate and repack storage in response to the requirements of the calculation. In NJOY, these functions are handled by the STORAG package of 4 subroutines.

STORAG(IAMAX,NIDMAX,IPR,A)

Initialize variably dimensioned dynamic storage allocation system for the container array A. IAMAX = length of container array. NIDMAX = maximum number of data identifiers that will be needed at one time. IPR = print flag (normally 0, use 1 to suppress most routine messages).

RESERV(ID,NWORDS,INDEX,A)

Reserve NWORDS in A for the data set identified by ID. ID can be a left-adjusted Hollerith name or a number less than or equal to 9999. Space will be allocated at the top of A if possible. If insufficient space is available, A will be repacked, and another attempt to reserve space will be made. If NWORDS = -1, repack A and assign all available words to this ID. INDEX points to the first word for data set ID in A.

RELEAS(ID,NWORDS,A)

Release all but NWORDS of the space assigned to ID in A. NWORDS = 0 deletes this ID. If NWORDS is less than zero, this ID and all ID entries above it are deleted. Note that repacking of A only takes place when the released space is really needed (see RESERV).

FINDEX(ID,INDEX,A)

Find the index for the data set ID. Using FINDEX is good practice if there is any chance that A might have been repacked since RESERV was called.

The NWORDS=-1 option in RESERV is useful when the number of words in a data set is not known in advance -- an example,


```

NW=-1
CALL RESERV(3HSIG,NW,LSIG,A)
READ(NIN)NW,(A(LSIG+I-1),I=1,NW)
CALL RELEAS(3HSIG,NW,A)

```

STORAG prints out routine messages (if IPR=0) so that the user can monitor the use of memory. The following example from THERMR illustrates several characteristics of STORAG.

```

1  STORAG  10/20000
2  ID SCR   1/ 2050
3  ID BUFO  2/ 3050
4  ID BUFN  3/ 4050
5  ID STK   4/ 4110
6  ID FL    5/19963
7  XX FL           406
8  XX STK           -1
9  ID E      4/ 4095
.
.
.

```

In line 1, STORAG is initialized with 20 000 words of core for up to 10 identifiers. In lines 2, 3, 4, and 5, space is reserved for SCR, BUFO, BUFN, and STK. The number before the slash is the ordinal number assigned to the identifier, and the second number is the total amount of storage used so far. In line 6, space for FL was reserved with NWORDS=-1. Therefore, 20 000 words less the STORAG table were allocated. If repacking had been necessary, a "REPACKING" message would have appeared here. The program determined that only 406 words were needed for FL, and the remainder of the storage was released in line 7. The maximum storage used to this point was $4110 + 406 = 4516$. Farther on, the code was finished with STK and FL, and both were released by a single call with NWORDS=-1 as indicated by line 8. Finally, line 9 shows a new identifier being assigned. Note that position 4 in the STORAG table was reused.

The STORAG system is compact and easy to use. The overhead required to use it is very small unless frequent repacking is required.

G. ENDF/B Utility Routines

There are several operations performed on ENDF/B data that are needed in so many other modules that it is practical to put them into the NJOY level.

TERP1(X1,Y1,X2,Y2,X,Y,I)

Interpolate for $y(x)$ between $y_1(x_1)$ and $y_2(x_2)$ using the ENDF/B interpolation law I [I=1 means $y=y_1$, I=2 means y is linear in x , I=3 means y is linear in $\ln(x)$, I=4 means $\ln(y)$ is linear in x , and I=5 means $\ln(y)$ is linear in $\ln(x)$].

TERPA(Y,X,XNET,IDIS,A,IP,IR)

Interpolate for $y(x)$ in the TAB1 structure in array A. The routine searches for the correct interpolation range starting from IP and IR (initialize to 2 and 1 for first call). It returns XNEXT, the next x value in the tabulation. IDIS is set to 1 if there is a discontinuity at XNEXT, it is zero otherwise).

GETY1(X,XNEXT,IDIS,Y1,ITAPE,A)

GETY2(X,XNEXT,IDIS,Y2,ITAPE,A)

Find $y(x)$ in a TAB1 structure starting at the current location on ITAPE by paging the data through array A. GETY1 and GETY2 are identical for occasions when two different tapes are being searched at the same time. XNEXT and IDIS behave as in TERPA. The array A must be at least NPAGE+50 words in length. These routines are normally used to retrieve cross sections from MF=3.

GRAL(XL,YL,XH,YH,X1,X2,I)

This function returns the integral from x_1 to x_2 of an ENDF/B function with interpolation law I (see TERP1). XL, YL, XH, and YH are the low and high limits of the interpolation panel.

INTEGA(F,X1,X2,A,IP,IR)

Integrate the TAB1 function stored in A from x_1 to x_2 . The routine automatically determines the correct interpolation law for each panel or fraction of a panel and uses GRAL to compute each part of the integral. Set IP=2 and IR=1 on the first call to INTEGA. In subsequent calls, the previous values of IP and IR will usually provide a good starting point for searching in the TAB1 structure.

H. Code Conversion

Standardization of the computer industry has not yet reached the point where it is possible to write a truly machine-independent FORTRAN code. However, by using fairly simple commands and isolating some functions in utility

subroutines, it is possible to minimize the number of changes that have to be made to convert a typical CDC code to an IBM machine. Furthermore, many of the changes can be made automatically with a simple preprocessing code (see Vol. I, App. D). NJOY uses the following trick:

```
      .  
      .  
      .  
CCDC      INTEGER H(5)  
CCDC  
CIBM  
C      REAL*8 H(5)  
CIBM  
      .  
      .  
      .
```

The variable H is intended to hold Hollerith data. To convert from CDC to IBM, simply add a C in column 1 of every card image bracketed by CCDC cards and remove the C from column 1 of every card image bracketed by CIBM cards.

Machine-dependent aspects of free-form input and interface I/O have been discussed above. Several other conversion problems are discussed here.

BANNER

This subroutine prints the NJOY banner on the output file. It includes a user field LAB, which should be changed to properly identify the user's installation. It also includes a variable MX, which can be used to indicate which machine was used at large computing centers; remove CALL MACH(MX) if a corresponding capability is not available. The date and time-of-day routines used here may have to be replaced with local equivalents.

ERROR

This subroutine should result in a fatal error exit and must be adjusted to reflect the local system. Special features such as trace-back information or saving files for later analysis can be performed here.

TIMER

This routine will have to be revised in many systems. The coding given is appropriate for CDC machines.

SIGFIG

Because of the many comparisons and searches that it makes, NJOY often has to match two numbers that are different only in the few least significant bits. This routine is intended to make such numbers exactly equal to each other by truncating the numbers to a given number of digits and removing any low-significance junk resulting from nonterminating binary fractions. This problem is not so common on short-word-length machines, but it might still be necessary to convert this routine for some machines.

I. Error Messages

NJOY***ILLEGAL ENDF/B VERSION NUMBER

Only 4 and 5 are allowed. ENDF/B-III data can be processed with IVERF=4.

NJOY***ILLEGAL OPTION

Use 0 for card-image input or 1 for TTY.

NJOY***ILLEGAL MODULE NAME

Check spelling, and check for missing (/) or incorrect item counts in the preceding module. Only the first four characters of each name are used.

OPENZ***ILLEGAL UNIT NUMBER

CLOSZ***ILLEGAL UNIT NUMBER

Units less than 10 are reserved for the system.

TOMEND***MODE CONVERSION NOT ALLOWED

TOFEND***MODE CONVERSION NOT ALLOWED

TOSEND***MODE CONVERSION NOT ALLOWED

Input and output units must both be binary or both be BCD.

FINDF***MAT---MF---MT---NOT ON TAPE

Desired section cannot be found.

STORAG***STORAGE EXCEEDED

There is not enough storage allocated to hold even the directory table.

RESERV***STORAGE EXCEEDED. NEED---MORE WORDS FOR ID---.

Container array is not large enough to hold desired data, even after re-packing. The message gives an estimate of the additional storage required.

RESERV***ID---ALREADY DEFINED

An ID must be released before being reassigned.

RESERV***POINTER SEQUENCE ERROR

The directory at the start of the container array has probably been clobbered.

RESERV***EXCEEDED MAXIMUM NUMBER OF ID-S

See NIDMAX in STORAG.

RESERV***REQUESTED RESERVE OF ZERO WORDS

Check coding that called RESERV.

RELEAS***ID---NOT DEFINED

Check coding and spelling.

RELEAS***ATTEMPT TO RELEASE MORE WORDS THAN STORED

Self-explanatory. Check coding.

FINDEX***ID---NOT DEFINED

Check coding and spelling.

J. References for NJOY

1. R. Kinsey, "ENDF-102, Data Formats and Procedures for the Evaluated Nuclear Data Files, ENDF," Brookhaven National Laboratory report BNL-NCS-50496 (ENDF 102) 2nd. Ed. (ENDF/B-V) (October 1979).
2. C. R. Weisbin, P. D. Soran, R. E. MacFarlane, D. R. Harris, R. J. LaBauve, J. S. Hendricks, J. E. White, and R. B. Kidman, "MINX, A Multigroup Interpretation of Nuclear X-Sections from ENDF/B," Los Alamos Scientific Laboratory report LA-6486-MS (ENDF-237) (1976).

X. RECONR

The RECONR module is used to reconstruct resonance cross sections from resonance parameters and to reconstruct cross sections from ENDF/B nonlinear interpolation schemes. The output is written as a pointwise-ENDF tape (PENDF) with all cross sections on a unionized energy grid suitable for linear interpolation to within a specified tolerance. Redundant reactions (for example, total or inelastic) are reconstructed to be exactly equal to the sum of their parts at all energies. The resonance parameters are removed from File 2, and the material dictionary is corrected to reflect all changes. Resonance reconstruction uses methods based on RESEND¹ and linearization uses the method developed for MINX.² RECONR has the following advantages over the RESEND module of MINX:

- Efficient use of dynamic storage allocation and a new stack structure allow large problems to be run without the use of secondary overlays.
- The unionized grid improves the accuracy, usefulness, and ENDF/B compatibility of the output.
- A correct material dictionary is provided.
- Approximate $\psi\chi$ Doppler broadening may be used to speed up reconstruction for narrow-resonance materials.
- A resonance-integral criterion is added to the normal linearization criterion, in order to reduce the number of points added to the tabulation to represent "unimportant" resonances.

A. ENDF/B Cross Section Representations

A typical cross section derived from an ENDF/B evaluation is shown in Fig. 1. The low-energy cross sections are "smooth". They are described in File 3 (see IX.D for a review of ENDF/B nomenclature) using cross-section values given on an energy grid with a specified law for interpolation between the points. In the resolved resonance range, resonance parameters are given in File 2, and the cross sections for resonance reactions have to be obtained by adding the contributions of all the resonances to "backgrounds" from MF3. At still higher energies comes the unresolved region where explicit resonances are no longer defined. Instead, the cross section is computed from statistical distribution of resonance parameters given in File 2 and backgrounds from File 3. Finally, at the highest energies, the smooth MF3 representation is used again.

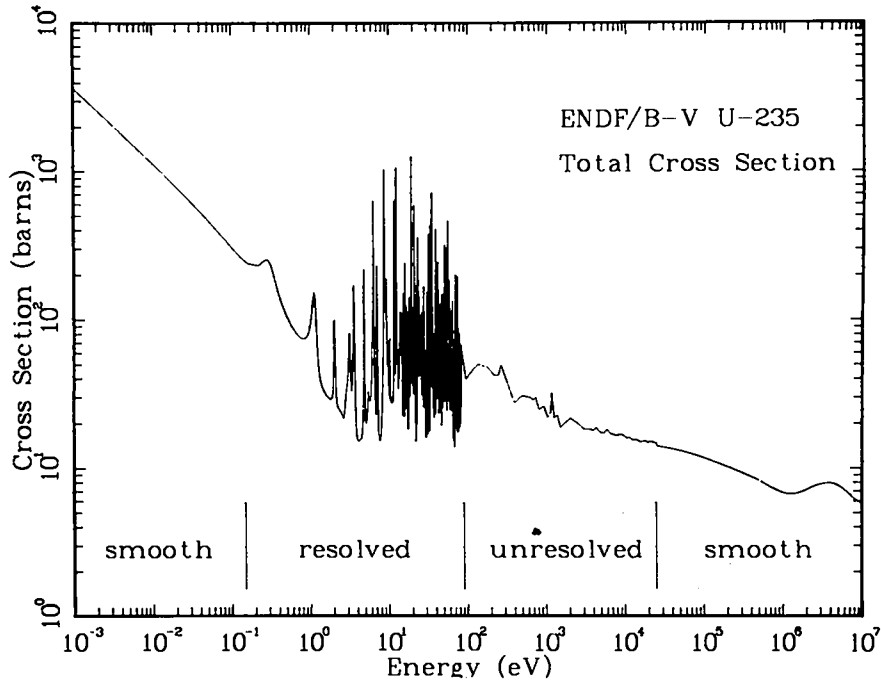


Fig. 1.

A typical cross section reconstructed from an ENDF/B evaluation using RECONR. The smooth, resolved, and unresolved energy regions use different representations of the cross sections.

For medium-mass isotopes, the unresolved range is usually omitted. For the lightest isotopes, the resolved range is also omitted, the resonance cross sections being given directly in the "smooth" format. In addition, several different resonance parameter representations are allowed. It is the purpose of RECONR to take all of these separate representations and produce a simple cross-section-versus-energy representation such as that shown in Fig. 1.

B. Unionization and Linearization Strategy

Several of the cross sections found in ENDF/B evaluation are summation cross sections (for example, total, inelastic, and sometimes n2n and fission), and it is important that each summation cross section be equal to the sum of its parts. However, if the partial cross sections are represented with nonlinear interpolation schemes, the sum cannot be represented by any simple interpolation

law. A typical case is the sum of elastic scattering (MT2 interpolated linearly to represent a constant) and radiative capture (MT102 interpolated log-log to represent $1/v$). The total cross section cannot be represented accurately by either scheme unless the grid points are very close together. This effect leads to significant balance errors in multigroup transport codes and to splitting problems in continuous-energy Monte Carlo codes.

Furthermore, the use of linear-linear interpolation (that is, σ linear in E) can be advantageous in several ways. The data can be plotted easily, they can be integrated easily, cross sections can be Doppler broadened efficiently (see BROADR), and, finally, linear data can be retrieved efficiently in continuous-energy Monte Carlo codes.

Therefore, RECONR puts all cross sections on a single unionized grid suitable for linear interpolation. As described in more detail below, RECONR makes one pass through the ENDF/B material to select the energy grid, then a second pass to compute cross sections on this grid. Each cross section on the PENDF tape (except for the redundant summation cross sections) is exactly equal to its ENDF/B value. The summation cross sections are then obtained by adding up the partials at each grid point.

While RECONR is going through the reactions given in the ENDF/B evaluation, it also checks the reaction thresholds against the Q value and atomic weight ratio to the neutron (AWR) given for the reaction. If the condition

$$\text{threshold} \geq \frac{\text{AWR}+1}{\text{AWR}} Q \quad (1)$$

is not satisfied, the threshold energy is moved up to satisfy the condition and an informative message is printed if the change exceeds 0.1%.

If desired, the unionized grid developed from the ENDF/B file can be supplemented with "user grid points" given in the input data. The code automatically adds 1.E-5 eV, 0.0253 eV, and 20 MeV to the grid if they are not already present.

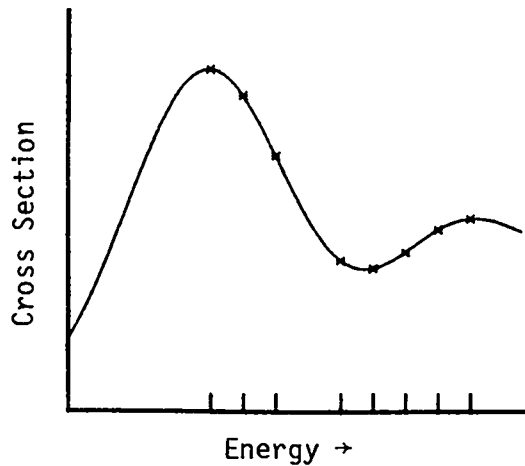
C. Linearization and Reconstruction Methods

Linearization (LUNION) and resonance reconstruction (RESXS) both function by inserting new energy grid points between the points of an original grid using an "inverted stack". The general concepts involved are illustrated with a simple example shown in Fig. 2. The stack is first primed with two starting values. For linearization, they will be two adjacent points on the original grid. For reconstruction, they will usually be the peaks of two adjacent resonances. The stack is said to be inverted because the lower energy is at the "top" (I=2).

This interval or "panel" is now divided into two parts, and the cross section computed at the intermediate point is compared to the result of linear interpolation between the adjacent points. If the two values do not agree within various criteria, the top of the stack is moved up one notch (I=3), and the new value is inserted (I=2). The code then repeats the checking process for the new (smaller) interval at the top of the stack. The top of the stack rises until convergence is achieved for the top interval. The top energy and cross section are then saved on a scratch file, the stack index is decremented, and the checks are repeated. This process is continued with the top of the stack rising and falling in response to the complexity of the cross section until the entire panel ΔE has been converged (I=1). The stack is then reprimed with the bounds of the next panel. The process continues until the entire energy range for linearization or reconstruction has been processed.

This new stack logic enables a panel to be divided into parts as small as $\Delta E/2^n$ where n is the stack size (currently 20), and several different cross sections (elastic, capture, fission) can easily be stored in arrays of this size. By contrast, RESEND used several arrays 500 words long and sometimes ran out of storage while subdividing between resonances.

Intervals are subdivided differently for linearization and resonance reconstruction. In the latter case, the interval is simply divided in half as in RESEND. For linearization the method developed by D. R. Harris for MINX² is used. Analytic formulas are used to choose the optimum intermediate point; this point turns out to be the energy value where the slope of the actual interpolation function equals the slope of the linear interpolate. Formulas are provided for each of the nonlinear ENDF/B interpolation laws: σ is linear in $\ln(E)$; $\ln(\sigma)$ is linear in E ; and $\ln(\sigma)$ is linear in $\ln(E)$.



Stack Version	Energy Grid Now Stored in Stack ^a	Result of Convergence Test on Lowest-Energy Segment
1	2 1	Not converged, add midpoint
2	3 2 1	"
3	4 3 2 1	"
4	5 4 3 2 1	Converged, write lowest E to disk.
5	4 3 2 1	"
6	3 2 1	"
7	2 1	Not converged, add midpoint
8	3 2 1	"
9	4 3 2 1	Converged, write lowest E to disk.
10	3 2 1	"
11	2 1	Not converged, add midpoint
12	3 2 1	Converged, write lowest E to disk.
13	2 1	"
14	1	Finished. Read energy of next resonance and repeat.

^aNumbers above energy markers indicate location in the stack. Total length of stack may be as large as 20.

Fig. 2.
Inverted-stack method used in LUNION and RESXS.

The convergence criterion used for linearization is that the linearized cross section at the intermediate point is within the fractional tolerance ERR of the actual cross section specified by the ENDF law. More complicated criteria are used for resonance reconstruction.

There are two basic problems that arise if a simple fractional tolerance test is used to control resonance reconstruction. First, as points are added to the energy grid, adjacent energy values may become so close that they will be rounded to the same number when a formatted output file is produced or when the machine-dependent limit for decimal single-precision accuracy is reached. It clearly makes no sense to continue to add grid points after this limit is reached. Through the use of dynamic format construction, the energy resolution available for formatted NJOY output is 7 significant figures (that is, $\pm 1.234567 \pm n$) rather than the usual 5 or 6 (see Section X.D). On "short-word" machines (32-36 bits per word), the limit set by precision is also about 7 significant figures. On "long-word" machines (typically 60-64 bits per word), binary output files can be used, and NJOY can produce up to 15 significant figures if necessary.

Significant figure control is implemented as follows: each intermediate energy is truncated to NDIGIT significant figures before the corresponding cross sections are computed, and if the resulting number is equal to either of the adjacent values, the interval is declared to be converged. Thus, no identical energies are produced, but an unpredictable loss in accuracy results. The error in the area of this interval is certainly less than $0.5 \cdot \Delta\sigma \cdot \Delta E$, so this value is added to an error estimate and a count of panels truncated by the significant figure check is incremented for a later informative diagnostic message.

The second basic problem alluded to above is that a very large number of resonance grid points arise from straightforward linear reconstruction of the resonance cross sections of some isotopes. Many of these points come from narrow, weak, high-energy resonances, which do not need to be treated accurately in many applications. As an example, the capture and fission resonance integrals important for thermal reactors must be computed with a $1/E$ flux weighting. If the resonance reconstruction tolerance is set high (say 1%) to reduce the cost of processing, the resonance integrals will be computed to only 1% accuracy. However, if the high-energy resonances (whose importance is reduced by the $1/E$ weight and the $1/v$ trend of the capture and fission cross sections) are treated with less accuracy than the low-energy resonances, then it is likely

that one can achieve an overall reduction in the number of points (hence computing cost), or increased accuracy in computed resonance integrals, or both. Since $1/E$ weighting is not realistic in all applications (for example, in fast reactors), user control of this "thinning" operation must be provided.

Based on these arguments, the following approach was chosen to control the problem of very large files. First, panels are subdivided until the elastic and capture cross sections are converged to within $ERRMAX$, where $ERRMAX \geq ERR$. These two tolerances are normally chosen to form a reasonable band, such as 10% and 0.5%, to ensure that all resonances are treated at least roughly (for example, for plotting). If the resonance integral ($1/E$ weight) in a particular panel is large, the panel is further subdivided to achieve an accuracy of ERR (say 0.5%). However, if the contribution to the resonance integral from any one interval gets small, the interval will be declared converged, and the local value of the cross section will end up with some intermediate accuracy. Once again, the contribution to the error in the resonance integral should be less than $0.5 \cdot \Delta\sigma \cdot \Delta E$. This value is added into an accumulating estimate of the error, and a count of panels truncated by the resonance integral check is incremented.

The problem with this test is that RECONR does not know the value of the resonance integral in advance, so the tolerance parameter $ERRINT$ is not the actual allowed fractional error in the integral. Instead, it is more like the resonance integral error per grid point (barns/point). Thus, a choice of $ERRINT=ERR/10000$ with $ERR=0.001$ would limit the integral error to about 0.001 barn if 10000 points resulted from reconstruction. Since important resonance integrals vary from a few barns to a few hundred barns, this is a reasonable choice. The integral check can be suppressed by setting $ERRINT$ very small or $ERRMAX=ERR$.

When resonance reconstruction is complete, RECONR provides a summary of the possible resonance integral error due to significant figure reduction and the integral check over several coarse energy bands (see Fig. 3). The last band covers the unresolved range, if present. The parameter $NDIGIT$ and the parameters $ERRMAX$ and $ERRINT$, taken together, should be considered as knobs that can increase or decrease the errors in their respective columns to get an appropriate balance of accuracy and economy for a particular application.

D. Resonance Representations

RECONR uses the resonance formulas as implemented in the original RESEND code¹ with three changes: a more efficient calculation of multilevel Breit-Wigner cross sections developed by C. Lubitz of the Knolls Atomic Power Laboratory (General Electric Co.) and coded by P. Rose of the Brookhaven National Laboratory, the addition of competitive widths introduced for ENDF/B-V, and a $\psi\chi$ Doppler-broadening calculation for single-level Breit-Wigner and Adler-Adler resonance shapes. An expanded discussion of the following formulas can be found in the ENDF/B-V format manual.³

The subroutine that computes single-level Breit-Wigner cross sections (CSSLBW) uses

$$\sigma_n = \sigma_p + \sum_r \sigma_m \{ [\cos 2\phi_\ell - (1 - \frac{\Gamma_n}{\Gamma_t})] \psi(\theta, x) + \sin 2\phi_\ell \chi(\theta, x) \} , \quad (2)$$

$$\sigma_f = \sum_r \sigma_m \frac{\Gamma_f}{\Gamma_t} \psi(\theta, x) , \quad (3)$$

$$\sigma_\gamma = \sum_r \sigma_m \frac{\Gamma_\gamma}{\Gamma_t} \psi(\theta, x) , \quad \text{and} \quad (4)$$

$$\sigma_p = \sum_\ell \frac{4\pi}{k^2} (2\ell+1) \sin^2 \phi_\ell , \quad (5)$$

where σ_n , σ_f , σ_γ , and σ_p are the neutron (elastic), fission, radiative capture, and potential scattering components of the cross section arising from the given resonances. There can be "background" cross sections in File 3 that must be added to these values to account for competitive reactions such as inelastic scattering or to correct for the inadequacies of the single-level representation with regard to multilevel effects or missed resonances. The sums extend over all the resolved resonances r that may belong to different spin sequences ℓ , J (L and AJ in the code). Each resonance is characterized by its total, neutron,

ESTIMATED MAXIMUM ERROR DUE TO
 RESONANCE INTEGRAL CHECK (ERRMAX,ERRINT)
 AND SIGNIFICANT FIGURE TRUNCATION (NDIGIT)

UPPER ENERGY	ELASTIC INTEGRAL	PERCENT ERROR RES-INT	PERCENT ERROR SIG-FIG	CAPTURE INTEGRAL	PERCENT ERROR RES-INT	PERCENT ERROR SIG-FIG
1.55E+02	7.45E+00	.000	0.000	2.82E-02	.009	0.000
4.96E+02	5.94E+00	.000	0.000	1.71E-01	.002	0.000
1.63E+03	4.12E+00	.000	0.000	6.80E-03	.149	0.000
5.20E+03	6.49E+00	.000	.000	1.61E-02	.134	.008
1.73E+04	9.66E+00	.001	.000	1.74E-02	.200	.097
5.62E+04	4.20E+00	.004	.009	1.19E-02	.216	4.284
1.78E+05	3.50E+00	.008	.007	5.64E-03	.257	5.265

POINTS ADDED BY RESONANCE RECONSTRUCTION = 12309
 POINTS AFFECTED BY RESONANCE INTEGRAL CHECK = 6969
 POINTS AFFECTED BY SIGNIFICANT FIGURE REDUCTION = 1262
 POINTS REMOVED BY BACKTHINNING = 201
 FINAL NUMBER OF RESONANCE POINTS = 12749

162.036S

.....

Fig. 3.

Sample of RECONR resonance-integral and significant-figure error summary.

fission, and capture widths ($\Gamma_t, \Gamma_n, \Gamma_f, \Gamma_\gamma$) and its maximum value ($S_{MAX} = \sigma_m / \Gamma_t$ in the code)

$$\sigma_m = \frac{4\pi}{k^2} g_J \frac{\Gamma_n}{\Gamma_t}, \quad (6)$$

where g_J is the spin statistical factor

$$g_J = \frac{2J+1}{4I+2}, \quad (7)$$

I is the total spin SPI given in File 2, and k is the neutron wave number that depends on incident energy E and the atomic weight ratio to the neutron for the isotope, AWRI, as follows:

$$k = (2.196771 \times 10^{-3}) \frac{AWRI}{AWRI+1} \sqrt{E} \quad . \quad (8)$$

The neutron width in these equations is energy dependent due to the penetration factors P_ℓ ; that is,

$$\Gamma_n(E) = \frac{P_\ell(E) \Gamma_n}{P_\ell(|E_r|)} \quad , \quad (9)$$

$$P_0 = \rho \quad , \quad (10)$$

$$P_1 = \frac{\rho^3}{1+\rho^2} \quad , \quad \text{and} \quad (11)$$

$$P_2 = \frac{\rho^5}{9+3\rho^2+\rho^4} \quad , \quad (12)$$

where E_r is the resonance energy and $\rho = ka$ depends on the channel radius (RA)

$$a = 0.123 AWRI^{1/3} + 0.08 \quad . \quad (13)$$

The phase shifts are given by

$$\phi_0 = \hat{\rho} \quad , \quad (14)$$

$$\phi_1 = \hat{\rho} - \tan^{-1} \hat{\rho} \quad , \quad \text{and} \quad (15)$$

$$\phi_2 = \hat{\rho} - \tan^{-1} \frac{3\hat{\rho}}{3-\hat{\rho}^2} \quad , \quad (16)$$

where $\hat{\rho} = \hat{k}a$ depends on the scattering radius AP given in the file. The final components of the cross section are the actual line shape functions ψ and χ . At zero temperature,

$$\psi = \frac{1}{1+x^2} \quad , \quad (17)$$

$$\chi = \frac{x}{1+x^2} \quad , \quad (18)$$

$$x = \frac{2(E-E'_r)}{\Gamma_t} \quad , \quad \text{and} \quad (19)$$

$$E'_r = E_r + \frac{S_\ell(|E_r|) - S_\ell(E)}{2(P_\ell(|E_r|))} \Gamma_n(|E_r|) \quad , \quad (20)$$

in terms of the shift factors

$$S_0 = 0 \quad , \quad (21)$$

$$S_1 = -\frac{1}{1+\rho^2} \quad , \quad \text{and} \quad (22)$$

$$S_2 = -\frac{18+3\rho^2}{9+3\rho^2+\rho^4} \quad (23)$$

To go to higher temperatures, define

$$\theta = \frac{\Gamma_t}{\sqrt{\frac{4kTE}{AWRI}}} \quad , \quad (24)$$

where k is the Boltzman constant and T is the absolute temperature. The line shapes ψ and χ are now given by

$$\psi = \frac{\sqrt{\pi}}{2} \theta \operatorname{Re}W\left(\frac{\theta x}{2}, \frac{\theta}{2}\right) \quad \text{and} \quad (25)$$

$$\chi = \frac{\sqrt{\pi}}{2} \theta \operatorname{Im}W\left(\frac{\theta x}{2}, \frac{\theta}{2}\right) \quad , \quad (26)$$

in terms of the complex probability function (see QUICKW, WTAB, and W, which came from the MC² code⁴)

$$\begin{aligned} W(z) &= e^{-z^2} \operatorname{erfc}(-iz) \\ &= \frac{i}{\pi} \int_{-\infty}^{\infty} \frac{e^{-t^2}}{z - t} dt \quad , \end{aligned} \quad (27)$$

where $z = x + iy$. The $\psi\chi$ method is not as accurate as kernel broadening (see BROADR) because the backgrounds (which are sometimes quite complex) are not broadened, and terms important for energies less than about $16kT/AWRI$ are neglected; however, the $\psi\chi$ method is less expensive than BROADR. The current version of RECONR includes Doppler broadening for the single-level Breit-Wigner representation only.

The Lubitz-Rose method used for calculating multilevel Breit-Wigner cross section (CSMLBW) is formulated as follows:

$$\sigma_n(E) = \sum_{\ell} \sigma_{n\ell}(E) \quad , \quad (28)$$

$$\sigma_{n\ell}(E) = \frac{\pi}{k^2} \sum_J g_J \left| 1 - U_{nJ}(E) \right|^2, \quad \text{and} \quad (29)$$

$$U_{nJ}(E) = e^{2i\phi_\ell} - \sum_r \frac{i\Gamma_{nr}}{E_r - E - i\Gamma_r/2}, \quad (30)$$

where the symbols are the same as those used above. Expanding the complex operations gives the actual formula used

$$\sigma_{n\ell}(E) = \frac{\pi}{k^2} \sum_J g_J \left\{ \left(1 - \cos 2\phi_\ell - \sum_r \frac{\Gamma_{nr}}{\Gamma_r} \frac{2}{1+x_r^2} \right)^2 + \left(\sin 2\phi_\ell + \sum_r \frac{\Gamma_{nr}}{\Gamma_r} \frac{2x_r}{1+x_r^2} \right)^2 \right\}, \quad (31)$$

where the sums over r are limited to resonances in spin sequence ℓ, J . The fission and capture cross sections are the same as for the single-level option.

The allowed values of J for this sum are limited to the range

$$|S - \ell| \text{ to } |S| + \ell,$$

where S is the magnitude of the channel spin $I - \frac{1}{2}$ and I is the target spin (SPI).

The multilevel Adler-Adler representation is defined for $\ell=0$ only. The total cross sections are given by

$$\begin{aligned} \sigma_t(E) = & \frac{4\pi}{k^2} \sin^2 \phi_0 + \frac{\pi\sqrt{E}}{k^2} \left\{ \sum_r \frac{1}{v_r} [(G_r \cos 2\phi_0 + H_r \sin 2\phi_0) \psi(\theta, x) \right. \\ & \left. + (H_r \cos 2\phi_0 - G_r \sin 2\phi_0) \chi(\theta, x)] \right. \\ & \left. + A_1 + A_2/E + A_3/E^2 + A_4/E^3 + B_1 E + B_2 E^2 \right\}, \quad (32) \end{aligned}$$

where

$$x = \frac{\mu_r - E}{v_r} , \quad (33)$$

and where v_r is the resonance half-width (corresponds to $\Gamma/2$ in the Breit-Wigner notation), μ_r is the resonance energy, G_r is the symmetric total parameter, H_r is the asymmetric total parameter, and the A_i and B_i are coefficients of the total background correction.

The fission and capture cross sections both use the form

$$\sigma_x(E) = \frac{\pi\sqrt{E}}{k^2} \left\{ \sum_r \frac{1}{v_r} [G_r\psi(\theta,x) + H_r\chi(\theta,x)] + A_1 + A_2/E + A_3/E^2 + A_4/E^4 + B_1E + B_2E^2 \right\} , \quad (34)$$

where the values of G , H , A_i , and B_i appropriate for the desired reaction are used.

Doppler broadening can be applied as for the SLBW case, except note that Γ_r in Eq. (24) must be replaced with $2v_r$. Doppler-broadened Adler-Adler cross sections are more accurate than SLBW cross sections because the background is smoother. However, cross sections below about 16kT/AWRI will still be inaccurate. An example of the agreement between $\psi\chi$ broadening and the more accurate kernel broadening (see BROADR) is shown in Fig. 4.

Infinitely dilute cross sections in the unresolved energy range are computed in CSUNR1 or CSUNR2 using average resonance parameters and probability distributions from File 2. With the approximations used, these cross sections are not temperature dependent; therefore, the results are a good match to resolved resonance data generated using TEMPR > 0. The formulas used are based on the single-level approximation with interference:

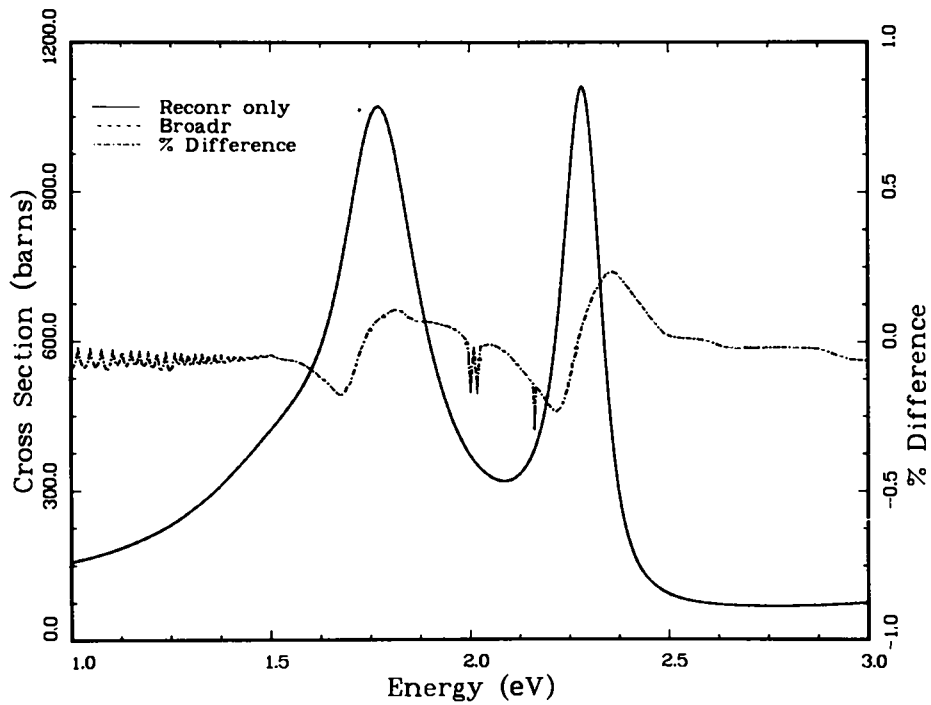


Fig. 4.

Comparison of Doppler-broadened cross sections generated with the $\psi\chi$ method (RECONR only) and the kernel method (BROADR) for ^{233}U at 300 K. The reconstruction tolerance was 0.2%.

$$\sigma_n(E) = \sigma_p + \frac{2\pi^2}{k^2} \sum_{\ell, J} \frac{g_J}{\bar{D}} [\bar{\Gamma}_n^2 R_n - 2\bar{\Gamma}_n \sin^2 \phi_\ell] , \quad (35)$$

$$\sigma_x(E) = \frac{2\pi^2}{k^2} \sum_{\ell, J} \frac{g_J}{\bar{D}} \bar{\Gamma}_n \bar{\Gamma}_x R_x , \quad (36)$$

$$\sigma_p = \frac{4\pi}{k^2} \sum_{\ell} (2\ell + 1) \sin^2 \phi_\ell , \quad (37)$$

where x stands for either fission or capture, $\bar{\Gamma}_i$ and \bar{D} are the appropriate average widths and spacing for the ℓ, J spin sequence, and R_i is the fluctuation integral for the reaction and spin sequence (see GNRL). These integrals are simply the averages taken over the chi-square distributions specified in the file; for example,

$$\begin{aligned} \bar{\Gamma}_n \bar{\Gamma}_f R_f &= \left\langle \frac{\Gamma_n \Gamma_f}{\Gamma} \right\rangle \\ &= \int dx_n P_\mu(x_n) \int dx_f P_\nu(x_f) \int dx_c P_\lambda(x_c) \frac{\Gamma_n(x_n) \Gamma_f(x_f)}{\Gamma_n(x_n) + \Gamma_f(x_f) + \Gamma_\gamma + \Gamma_c(x_c)}, \end{aligned} \quad (38)$$

where $P_\mu(x)$ is the chi-square distribution for μ degrees of freedom. The integrals are evaluated with the quadrature scheme developed for MC²-II⁵ giving

$$R_f = \sum_i W_i^\mu \sum_J W_J^\nu \sum_k W_k^\lambda \frac{Q_i^\mu Q_j^\nu}{\bar{\Gamma}_n Q_i^\mu + \bar{\Gamma}_f Q_j^\nu + \Gamma_\gamma + \bar{\Gamma}_c Q_k^\lambda} . \quad (39)$$

The W_i^μ and Q_i^μ are the appropriate quadrature weights and values for μ degrees of freedom, and Γ_γ is assumed to be constant (many degrees of freedom). The competitive width $\bar{\Gamma}_c$ is assumed to effect the fluctuations, but a corresponding cross section is not computed. The entire competitive cross section is supposed to be in the File 3 total cross section as a smooth background.

It should be noted that the reduced average neutron width (AMUN) is given in the file, and

$$\bar{\Gamma}_n = \Gamma_n^0 \sqrt{E} V_\ell(E) , \quad (40)$$

where the penetrabilities for the unresolved region are defined as

$$V_0 = 1 \quad , \quad (41)$$

$$V_1 = \frac{\rho^2}{1 + \rho^2} \quad , \quad \text{and} \quad (42)$$

$$V_2 = \frac{\rho^4}{\rho + 3\rho^2 + \rho^4} \quad . \quad (43)$$

Other parameters are defined as for SLBW.

Unresolved parameters can be given as independent of energy, only fission widths dependent on energy, or as fully energy dependent. The first two options are processed in CSUNR1 and the last one in CSUNR2. The ENDF/B-V formats specify that cross sections are to be computed at the specified energy points, and the cross sections are to be computed for energies between these points by interpolation. However, this procedure gives unreasonable results for the energy-independent evaluations carried over from earlier versions of ENDF/B. Therefore, RECONR is allowed to linearize the unresolved cross section using interpolation on parameters. For most applications, the numbers in this energy range are replaced by UNRESR where a different strategy is used to select intermediate points.

E. Code Description

The flow of this module is controlled by the RECONR program. The first step is to read cards 1, 2, and 3 of the user's input. The TAPEID record of the input tape (NENDF) is read and printed, then the new TAPEID record is written on the output tape (NPEND). RECONR is now ready to enter the loop over desired materials.

For each material, STORAG is used to allocate space for the energy nodes and for scratch storage (ENODE,SCR) and RUIN is called to read cards 4 through 7 of the user's input. RUIN automatically adds the ENDF/B energy limits of 1×10^{-5} eV and 20 MeV and the thermal energy 0.0253 eV to any energy grid

points entered by the user. If the reconstruction temperature (TEMPR) is greater than zero, a table of ψ and χ functions is generated (the W table; see WTAB and QUICKW). The FINDF utility subroutine is then used to find the first card of file 1 (MF=1,MT=451) for the desired material.

File 1 on the input ENDF/B tape is examined to obtain certain constants and flags and to analyze the dictionary (ANLYZD). The "dictionary" is really an index to all the files and sections (reactions) appearing for the MAT. ANLYZD determines which reactions should be considered "redundant"; that is, the summation reactions that will be included on the PENDF tape. The total cross section (MT=1 for neutrons, MT=501 for photons) will always be included; the non-elastic cross section (MT=3) will be included if it is needed for photon production (that is, MF12, MT3 is found); the inelastic cross section (MT=4) will be included if MT51 through MT91 occurs, and the total fission reaction (MT18) will be called redundant if the partial fission representation (MT19, 20, 21, 38) is found. Space for the new material dictionary is then reserved (MFS, MTS, NCS). Section identification and card counts will be entered into these arrays as they are determined.

The next step is to read File 2, which contains resolved and unresolved resonance parameters (if any). The array RES is assigned to contain the File 2 data and RDFIL2 is called to read them. While the resonance parameters are being stored, RECONR adds each resonance energy to its list of energy nodes (ENODE). In the unresolved energy range, RECONR uses the energies of tabulated parameters or fission widths if available. If the evaluation uses energy-independent parameters, RECONR creates additional node energies with equal lethargy spacing. The energy nodes are sorted into order and duplications are removed. When control is returned to RECONR, any unused space in the RES array is released to be made available for other uses.

The subroutine LUNION is used to linearize and unionize the ENDF/B data. Space is reserved for two buffers to be used by LOADA/FINDA and for the linearization stack (Y and X). The length of the stack (NDIM) determines the smallest possible subdivision of each panel (energy points as close as 2^{-NDIM} times the panel width can be generated). Since the number of energies in the union grid may soon exceed the capacity of any reasonable small-core array, the existing list of energy nodes is copied to binary scratch storage (LOADA/FINDA). This

storage system consists of the buffers BOLD and BNEW and the scratch units IOLD and INEW. The energy grid points will "ping pong" back and forth between units 14 and 15 as the union grid is built up. LUNION now starts with MT=2 and checks each reaction in sequence to determine whether the current grid (on IOLD) is sufficient to represent the reaction to within the desired tolerance using linear interpolation. If not, RECONR uses ISLIN1 to select the optimum points to be added to the new grid (on INEW). INEW and IOLD are swapped and the next MT is processed. When all nonredundant reactions have been examined, the list of energies in LOADA/FINDA storage is the desired linearized and unionized grid. The storage used is released.

This grid is used as the starting point for resonance reconstruction in RESXS. RESXS first reserves space for the LOADA/FINDA buffers BUFR and BUFG, the linearization stack (X and Y), and the partial cross sections (SIG). The length of the stack (NDIM) determines the smallest possible subdivision of a panel between two nodes (energy points as close as 2^{-NDIM} times the panel width can be generated). RESXS then examines the grid on NGRID (IOLD from LUNION) panel by panel. Grid points are added and cross sections computed until the convergence criteria discussed in Section C are satisfied. The cross sections are copied to NOUT through LOADA/FINDA and RESXS continues to the next panel. This procedure is continued until all panels are converged. The result is a tape (NOUT) containing the energy grid in the resonance region and the total, elastic, fission, and capture cross sections at each energy point.

Unionization is obtained automatically in the resonance region since the three partials are computed simultaneously in SIGMA. This routine calls CSNORP if there are no resonance parameters, CSSLBW for single-level Breit-Wigner parameters, CSMLBW for multilevel Breit-Wigner parameters, CSAA for Adler-Adler parameters, and CSUNR1 or CSUNR2 for unresolved resonance parameters. A new feature of NJOY is the ability to reconstruct the cross sections at TEMPR by ψ_x broadening if single-level Breit-Wigner (SLBW) or Adler-Adler (AA) parameters are given. The Doppler-broadened resonance shapes are obtained using QUICKW (see description in UNRESR) in CSSLBW or CSAA and the linearization procedure proceeds as before.

The resonance cross sections on NGRID are merged with the ENDF/B cross sections in EMERGE. First, the background grid from LUNION is merged with the

resonance grid from RESXS and written onto the LOADA/FINDA file, which will accumulate the total cross section and any other redundant reactions required (IOLD/INEW). A loop is then set up over all nonredundant reactions. For each grid point, the ENDF cross section is obtained by interpolation. If this grid point has a resonance contribution on NRES, it is added. The resulting net cross section at this point is added into the appropriate redundant cross sections on IOLD/INEW and also saved on NGRID. When all the energies for this reaction have been processed, the cross sections on NGRID are converted into a TAB1 record and written onto NSCR. This loop is continued until all reactions have been processed. When EMERGE is finished, NSCR contains cross sections for all the nonredundant reactions, and IOLD contains the redundant summation cross sections.

Control now passes to RECOU, which writes the new file 1 comments and dictionary. It then steps through the reactions on NSCR and IOLD. Redundant reactions are converted to TAB1 records and inserted in the correct order. Non-redundant reactions are simply copied. Finally a MEND record is added and control is returned to RECONR.

RECONR either directs that this process be repeated for another isotope or writes a TEND record and terminates. The result is a new tape in ENDF format containing the desired pointwise cross sections. Note that only files 1, 2, 3, and 13 are included for neutron tapes. Only 1 and 23 are included for photon tapes.

F. Input Instructions

The input instructions for each module are given in the code as comment cards at the beginning of each module. They are reproduced here for the convenience of the reader (see also Vol. I, User's Manual).

```

C *---INPUT SPECIFICATIONS (FREE FORMAT)-----*
C *
C * CARD 1
C *   NENDF   UNIT FOR ENDF/B TAPE
C *   NPEND   UNIT FOR PENDF TAPE
C * CARD 2
C *   LABEL   66 CHARACTER LABEL FOR NEW PENDF TAPE
C *           DELIMITED WITH *, ENDED WITH /.
C * CARD 3
C *   MAT     MATERIAL TO BE RECONSTRUCTED
C *   NCARDS  NUMBER OF CARDS OF DESCRIPTIVE DATA FOR NEW MF1
C *           (DEFAULT=0.)
C *   NGRID   NUMBER OF USER ENERGY GRID POINTS TO BE ADDED
C *           (DEFAULT=0.)
C *   ERR     FRACTIONAL RECONSTRUCTION TOLERANCE USED WHEN
C *           RESONANCE-INTEGRAL ERROR CRITERION (SEE ERRINT)
C *           IS NOT SATISFIED.
C *   TEMPR   RECONSTRUCTION TEMPERATURE (DEG KELVIN)
C *           (DEFAULT=0.)
C *   NDIGIT  NO. SIGNIFICANT DIGITS (DEFAULT=6)
C *   ERRMAX  FRACTIONAL RECONSTRUCTION TOLERANCE USED WHEN
C *           RESONANCE-INTEGRAL ERROR CRITERION IS SATISFIED
C *           (ERRMAX.GE.ERR. DEFAULT=20.*ERR)
C *   ERRINT  MAXIMUM RESONANCE-INTEGRAL ERROR (IN BARNS)
C *           PER GRID POINT (DEFAULT=ERR/10000)
C * CARD 5
C *   CARDS   NCARDS OF DESCRIPTIVE COMMENTS FOR MT451
C *           EACH CARD DELIMITED WITH *, ENDED WITH/.
C * CARD 6
C *   ENODE   USERS ENERGY GRID POINTS
C *           CARDS 3, 4, 5, 6 MUST BE INPUT FOR EACH MATERIAL DESIRED
C *           MAT=0/ TERMINATES EXECUTION OF RECONR.
C *

```

A sample input for processing two isotopes from ENDF/B-IV tape 407 follows:

```

*RECONR*
20 -21
*PENDF TAPE FOR U-235 AND PU-239 FROM T407*/
1261 2 0
.005 0./
*92-U-235 FROM T407*/
*PROCESSED WITH NJOY*/
1264 2 0
.005 0./
*94-PU-239 FROM T407*/
*PROCESSED WITH NJOY*/
0/

```

The resulting PENDF tape will contain the desired TAPEID card, followed by U235, a MEND card, PU239, a MEND card, and a TEND card.

G. Error Messages

RUINAILLEGAL NDIGIT**

Value must be between 1 and 15. Any value above 7 will be ineffective on a short-word computer.

RDFIL2*STORAGE IN ENODE EXCEEDED**

Too many energy nodes including the user's nodes and the energies from MF2. Increase NODMAX in RECONR.

RDFIL2*STORAGE IN A EXCEEDED**

Too much resonance data. The main container array is too small. Increase /STORE/ and JX in RECONR or decrease buffer sizes NBUFG, NBUFR, or NBUF.

ANLYZD*TOO MANY REDUNDANT REACTIONS**

Increase the size of MTR(10) and MTRT(10) in /RECON/ and increase NMTMAX=10 in RECONR.

LUNION*EXCEEDED STACK**

Increase length of linearization stack NDIM (currently 20).

RESXS*STACK EXCEEDED**

Increase length of reconstruction stack NDIM (currently 20).

CSMLBW*NOT CODED FOR T.GT.0 DEG K**

The ψ_x Doppler-broadening option is only coded for single-level Breit-Wigner and Adler-Adler resonance parameters. Use TEMPR=0. on input.

CSAA*LI=**

Error in format of evaluation.

RECOUT*FOR MF -- MT --**

Indexing and pair count for this section do not make sense.

H. Input-Output Units

The following logical units are used:

- 10 NSCR1 in RECONR, NOUT in LUNION, and NIN in EMERGE. Contains copy of nonredundant sections from original ENDF/B tape.
- 11 NSCR2 in RECONR; NGRID in LUNION, RESXS, and EMERGE. Contains union grid for ENDF/B tape (not counting resonances).
- 12 NSCR3 in RECONR, NOUT in RESXS, and NRES in EMERGE. Contains resonance grid and cross sections.
- 13 NSCR4 in RECONR is used for two separate purposes. In RESXS it is a binary scratch file NSCR used for the unthinned resonance data. In EMERGE and RECOU it is NEMERGE and contains the nonredundant reactions on the union grid.
- 14/15 IOLD/INEW in LUNION. Used locally only to accumulate union grid for ENDF/B cross sections.
- 14/15 IOLD/INEW in EMERGE. Used locally only to accumulate summation cross sections on union grid.
- 20-99 User's choice for NENDF and NPEND to link RECONR with other modules.
- 5,6,7 System I/O units (see NJOY).

Note that 11, 12, 14, and 15 are always binary. Unit 10 has the same mode as NENDF. Unit 13 is binary when used in RESXS, and it has the same mode as NPEND elsewhere. NPEND can have a different mode than NENDF.

I. Storage Allocation

Storage allocation in RECONR is sensitive to (1) the amount of resonance parameter data, (2) the size of the resonance reconstruction stack, (3) the use of ψ_x broadening, and (4) the sizes of LOADA/FINDA buffers. Other storage requirements are minor.

Buffer sizes can be reduced or increased at will. The result is a storage/speed tradeoff with no change in capability or accuracy. See NBUFG=2000, NBUFR=2000, and NBUF=2000 in RECONR.

The ψ_x broadening option requires 7688 words of additional storage. Therefore, the container array in /STORE/ can be reduced significantly if ψ_x is not required. No code changes are needed--just avoid TEMPR greater than zero.

Resonance reconstruction in RESXS uses $5 \times \text{NDIM}$ words. NDIM determines the smallest subdivision of a panel that can be obtained. Using NDIM=20 allows points to be generated with spacing as small as one-millionth of the panel size (2^{20}).

J. References for RECONR

1. O. Ozer, "RESEND: A Program to Preprocess ENDF/B Materials With Resonance Files into Pointwise Form," Brookhaven National Laboratory report BNL-17134 (1972).
2. C. R. Weisbin, P. D. Soran, R. E. MacFarlane, D. R. Harris, R. J. LaBauve, J. S. Hendricks, J. E. White, and R. B. Kidman, "MINX: A Multigroup Interpretation of Nuclear X-Sections from ENDF/B," Los Alamos Scientific Laboratory report LA-6486-MS (ENDF-237) (1976).
3. R. Kinsey, Ed., "ENDF-102, Data Formats and Procedures for the Evaluated Nuclear Data File, ENDF," Brookhaven National Laboratory report BNL-NCS-50496 (ENDF 102), 2nd Edition (ENDF/B-V) (1979).
4. B. J. Toppel, A. L. Rago, and D. M. O'Shea, "MC², A Code to Calculate Multigroup Cross Sections," Argonne National Laboratory report ANL-7318 (1967).
5. H. Henryson II, B. J. Toppel, and C. G. Stenberg, "MC²-2: A Code to Calculate Fast Neutron Spectra and Multigroup Cross Sections," Argonne National Laboratory report ANL-8144 (ENDF-239) (1976).

XI. BROADR

BROADR generates Doppler-broadened and thinned cross sections in PENDF format starting from piecewise linear cross sections in PENDF format. The input cross sections can be from RECONR or from a previous BROADR run. The code is based on SIGMA1¹ by D. E. Cullen of Lawrence Livermore National Laboratory. The method is often called "kernel broadening" because it is based on a detailed integration of the integral equation defining the effective cross section. It is a fully accurate method, treating all resonance and nonresonance cross sections including multilevel effects. BROADR differs from SIGMA1 in the following ways.

- An alternate calculation is used for low energies and high temperatures that corrects a numerical problem of the original code.
- Variable dimensioning is used, which allows the code to be run on large or small machines with full use of whatever storage is made available.
- All low-threshold reactions are broadened and thinned in parallel on a union grid. This makes the code run several times faster than the original SIGMA1.
- Binary input and output can be used. This roughly halves the time required for a typical run.
- The summation cross sections total, nonelastic, and sometimes fission (MT18) or n2n are reconstructed to equal the sum of their parts.
- The file "dictionary" is updated.

A. Doppler-Broadening Theory

The effective cross section for a material at temperature T is defined to be that cross section that gives the same reaction rate for stationary target nuclei as the real cross section gives for moving nuclei. Therefore,

$$\rho v \bar{\sigma}(v, T) = \int d\vec{v}' \rho |\vec{v} - \vec{v}'| \sigma(|\vec{v} - \vec{v}'|) P(\vec{v}', T) \quad , \quad (1)$$

where \vec{v} is the velocity of the incident particle, \vec{v}' is the velocity of the target, ρ is the density of target nuclei, σ is the cross section for stationary

nuclei, and $P(\vec{v}', T)$ is the distribution of target velocities in the laboratory system. For many cases of interest, the target motion is isotropic and the distribution of velocities can be described by the Maxwell-Boltzmann function

$$P(\vec{v}', T) d\vec{v}' = \frac{\alpha^{3/2}}{\pi^{3/2}} e^{-\alpha v'^2} d\vec{v}' \quad , \quad (2)$$

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

Equation (1) can be partially integrated in terms of the relative speed $V = |\vec{v} - \vec{v}'|$ to give the standard form of the Doppler-broadened cross section:

$$\bar{\sigma}(v) = \frac{\alpha^{1/2}}{\pi^{1/2} v^2} \int_0^{\infty} dV \sigma(V) V^2 \{e^{-\alpha(V-v)^2} - e^{-\alpha(V+v)^2}\} \quad . \quad (3)$$

It is instructive to break this up into two parts:

$$\bar{\sigma}(v) = \sigma^*(v) - \sigma^*(-v) \quad , \quad (4)$$

where

$$\sigma^*(v) = \frac{\alpha^{1/2}}{\pi^{1/2} v^2} \int_0^{\infty} dV \sigma(V) V^2 e^{-\alpha(V-v)^2} \quad . \quad (5)$$

The exponential function in Eq. (5) limits the significant part of the integral to the range

$$v - \frac{4}{\sqrt{\alpha}} < V < v + \frac{4}{\sqrt{\alpha}} \quad .$$

For $\sigma^*(-v)$, the integral depends only on velocities satisfying

$$0 \leq v < \frac{4}{\sqrt{\alpha}} .$$

These results can be converted to energy units using

$$E_m = \frac{1}{2} m \left(\frac{4}{\sqrt{\alpha}} \right)^2 = \frac{16kT}{A} .$$

Some examples are given in Table I. Doppler-broadening effects will be important below this energy and for any features such as resonances, thresholds, or artificial discontinuities in evaluations that are not slowly-varying with respect to $2\sqrt{E_m E}$. As an example for ^{235}U at 100 eV, Doppler effects are important for features smaller than about 0.8 eV.

The numerical evaluation of Eq. (5) developed for SIGMA1 assumes that the cross section can be represented by a piecewise linear function of energy to acceptable accuracy. This is just the form of the NJOY PENDF tapes (see RECONR). Defining the reduced variables $y = \sqrt{\alpha v}$ and $x = \sqrt{\alpha V}$, the cross section becomes

$$\sigma(x) = \sigma_i + s_i(x^2 - x_i^2) \tag{6}$$

TABLE I
ENERGY PARAMETER FOR EFFECTIVE DOPPLER BROADENING

<u>Target</u>	<u>Temperature</u>	<u>Energy Parameter (E_m)</u>
H ₂	300 K	0.2 eV
U-235	300 K	0.0017 eV
U-235	1.0 keV	69 eV

with slope $s_i = (\sigma_{i+1} - \sigma_i)/(x_{i+1}^2 - x_i^2)$. Equation (5) can now be written as

$$\begin{aligned} \sigma^*(y) &= \frac{1}{\pi^{1/2} y^2} \sum_{i=0}^N \int_{x_i}^{x_{i+1}} \sigma(x) x^2 e^{-(x-y)^2} dx \\ &= \sum_i \{A_i[\sigma_i - s_i x_i^2] + B_i s_i\} \end{aligned} \quad (7)$$

where $x_0 = 0$, $x_{N+1} = \infty$,

$$\begin{aligned} A_i &= \frac{1}{y^2} H_2 + \frac{2}{y} H_1 + H_0, \quad \text{and} \\ B_i &= \frac{1}{y^2} H_4 + \frac{4}{y} H_3 + 6H_2 + 4yH_1 + y^2 H_0, \end{aligned} \quad (8)$$

and where " H_n " is shorthand for $H_n(x_i - y, x_{i+1} - y)$. The extrapolations to zero and infinity assume a constant cross section ($s_0 = s_N = 0$). The H functions are the incomplete probability integrals defined by

$$H_n(a,b) = \frac{1}{\sqrt{\pi}} \int_a^b z^n e^{-z^2} dz. \quad (9)$$

These functions can be computed in two ways. First,

$$H_n(a,b) = F_n(a) - F_n(b) \quad (10)$$

where

$$F_n(a) = \frac{1}{\sqrt{\pi}} \int_a^{\infty} z^n e^{-z^2} dz \quad . \quad (11)$$

These functions satisfy a recursion relation that can be used to obtain

$$F_0(a) = \frac{1}{2} \operatorname{erfc}(a) \quad ,$$

$$F_1(a) = \frac{1}{2\sqrt{\pi}} e^{-a^2} \quad , \quad (12)$$

⋮

$$F_n(a) = \frac{n-1}{2} F_{n-2}(a) + a^{n-1} F_1(a) \quad ,$$

⋮

where $\operatorname{erfc}(a)$ denotes the complementary error function

$$\operatorname{erfc}(a) = \frac{2}{\sqrt{\pi}} \int_a^{\infty} e^{-z^2} dz \quad . \quad (13)$$

However, when $F_n(a) \approx F_n(b)$, the difference in Eq. (10) may lose significance. In such cases $H_n(a,b)$ can be computed by a direct Taylor expansion of the defining integral.² Write

$$\begin{aligned}
H_n(a,b) &= \frac{1}{\sqrt{\pi}} \int_0^b z^n e^{-z^2} dz - \frac{1}{\sqrt{\pi}} \int_0^a z^n e^{-z^2} dz \\
&= G_n(b) - G_n(a) \quad .
\end{aligned} \tag{14}$$

But by Taylor's Theorem,

$$G_n(b) - G_n(a) = \frac{b-a}{1!} G_n'(a) + \dots + \frac{(b-a)^m}{m!} G_n^{(m)}(a) + \dots \quad . \tag{15}$$

Also,

$$G_n^{(m)}(x) = \frac{d^{m-1}}{dx^{m-1}} [x^n e^{-x^2}] = e^{-x^2} P_n^m(x) \quad , \tag{16}$$

where $P_n^m(x)$ is a polynomial with recursion relation

$$P_n^m(x) = \frac{d}{dx} P_n^{m-1}(x) - 2xP_n^{m-1}(x) \tag{17}$$

with $P_n^1 = x^n$. From this point, it is straightforward to generate terms until the desired number of significant figures are obtained.

When interpreting BROADR output, it is useful to remember several important features of the Doppler-broadening process. A $1/v$ cross section remains unchanged. Contrary to "popular knowledge", the area under a resonance does not remain unchanged unless $E \gg kT/A$. In fact, each resonance develops a new $1/v$ tail. Finally, a constant cross section (for example, elastic scattering) develops a $1/v$ tail at low energies after Doppler broadening. These effects are shown in Figs. 1, 2, and 3; they can be best understood by noting that the Doppler process preserves reaction rate $v\sigma(v)$ according to Eq. (1), and a finite reaction rate is expected for $T > 0$ even as $v \rightarrow 0$.

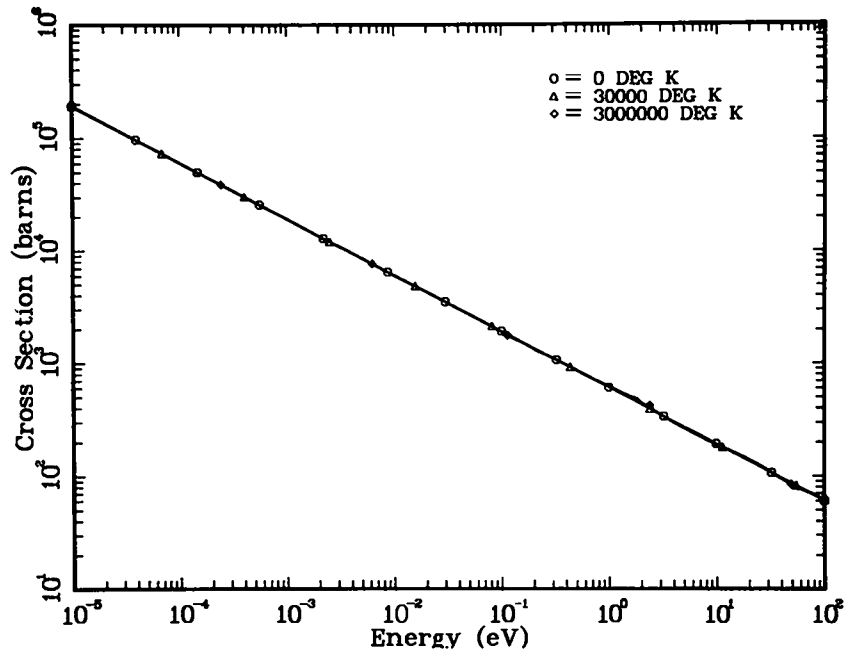


Fig. 1.

The (n,α) cross section for ^{10}B from ENDF/B-V for three different temperatures showing that a $1/v$ cross section is invariant under Doppler broadening.

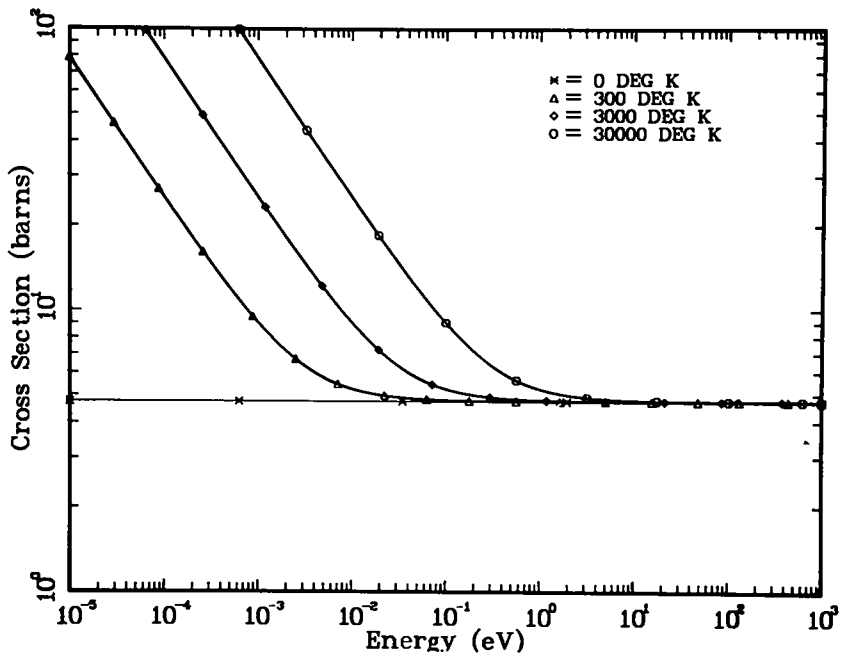


Fig. 2.

The elastic cross section for carbon from ENDF/B-V showing that Doppler broadening a constant cross section adds a $1/v$ tail.

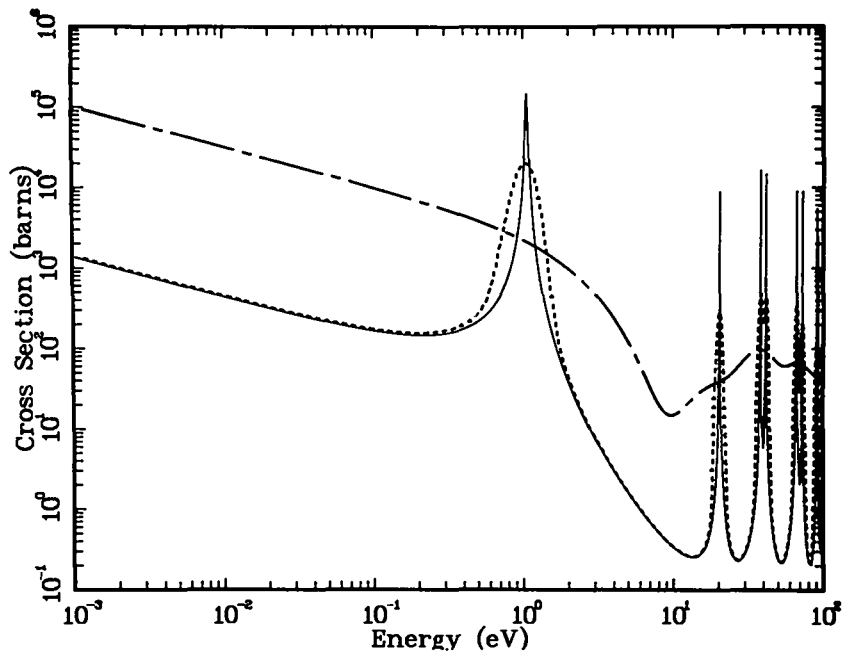


Fig. 3.

The (n,γ) cross section for ^{240}Pu from ENDF/B-V for several temperatures showing the effects of Doppler broadening on resonances. The temperatures are 0 K (solid), 30 000 K (dotted), and 300 000 K (dash-dot). The higher resonances behave in the classical manner even at 30 000K ; note that the line shape returns to the asymptotic value in the wings of the resonance. All resonances at 300 000 K (and to a lesser extent the first resonance for 30 000 K) show the additional $1/v$ component that appears when kT/A is large with respect to the resonance energy.

B. Data-Paging Methodology

A piecewise linear representation of a reaction cross section of a resonance material may require a very large number of energy points. For example, ENDF/B-IV U-238 (MAT1262) requires 57 400 points for the total cross section for 0.5% precision (ERRMAX=ERR). It is impractical to load all these points into memory simultaneously. However, the discussion following Eq. (5) in the theory section shows that only a limited energy range around the point of interest is required.

The strategy used is to stage the cross-section data into three "pages" of NPAGE points each. Points in the center page can then be broadened using the NPAGE or more points on each side of the point of interest. If $v = 4/\sqrt{\alpha}$ and

$v + 4/\sqrt{\alpha}$ are both included in the three-page range, accurate broadening can be performed. If not, a diagnostic warning is printed; the user should repeat the calculation with a smaller temperature step or larger page size.

There are many different reaction cross sections for each material. However, the cross sections for high velocities are normally smooth with respect to $32kT/A$ for any temperatures outside of stellar atmospheres; therefore, they do not show significant Doppler effects. The code uses the input value THNMAX or the lowest threshold (typically > 100 keV) below the input THNMAX as a breakpoint. No Doppler broadening or thinning is performed above that energy. Furthermore, the A_i and B_i factors in Eq. (7) depend only on the energy (or velocity) values and not on the cross sections. Since the A_i and B_i are expensive to compute, the code computes them only once for the points of a unionized energy grid (see RECONR). The sum of Eq. (7) is accumulated for all the nonthreshold reactions simultaneously. This trick makes BROADR several times faster than the original SIGMA1.

C. Coding Details

The code begins by reading the user's input (see Section D). Storage is then allocated for the LOADA/FINDA buffers (IBUFO and IbufN) and for the scratch storage (ISCR). The buffer length NBUF can be changed at will (currently NBUF=1000).

The input PENDF tape is searched for the desired material (MAT1). If the restart option is set (ISTART=1), the temperatures less than or equal to TEMP1 for MAT1 are assumed to have been broadened previously and they are copied to the output file. In either case, the files for TEMP1 are copied to a scratch file on unit NSCR1 (currently set to 10).

Next NSCR1 is rewound and examined reaction by reaction. The energy grid from the total cross section (MT1) is saved on scratch storage using LOADA. If the input tape has not been through RECONR, the BROADR module will still run, but at possibly reduced accuracy. The next low-threshold reaction (less than $E_{MIN} = 1$ eV) is located on NSCR1. The energy points are retrieved from scratch file IOLD (12 or 13) using FINDA, the cross sections for this reaction are computed on this grid, and the results are stored on scratch file INEW (13 or 12) using LOADA. The units for IOLD and INEW are then exchanged, and the entire process is repeated for the next low-threshold reaction.

The final result of this process is a list of NREAC low threshold reaction types in MTR (usually MT2, MT18, and MT102), the first high threshold or the input value in THNMAX, and a scratch file IOLD containing the energy grid and all the low threshold reactions (there are N2IN points).

Now that the number of reactions to be broadened simultaneously is known (NREAC), storage for data paging can be assigned. The total amount of storage available is $NAMAX - 2 * NBUF - NWSCR - 40$. The value of NAMAX should be as large as possible (current value is 30 000). This space is divided up into the largest possible page size, NPAGE. An overflow region NSTACK is also allocated. STORAG is used to allocate three pages for energies (E), three pages for each reaction cross section (S), one extended page for the broadened energy grid (EB), and three extended pages for the broadened cross sections (SB). This system is designed to use the available storage with maximum efficiency.

The cross sections on IOLD are now broadened and thinned by FILE3 (see below) and the results are written on scratch unit INEW using LOADA.

The "dictionary" from NSCR1 (actually an index) is revised to reflect any thinning and written on the output PENDF tape (NOUT). Note that the new temperature is written into the first word of the Hollerith data record to simplify later searching.

The broadened cross sections are now converted into ENDF TAB1 records and merged with the unbroadened cross sections on NSCR1. The total cross section (and sometimes nonelastic, fission, and n2n) is reconstructed to equal the sum of its parts. The new Doppler broadened "MAT" on NOUT is a legal PENDF file with the same MAT number as the original data but a new temperature.

The process is now repeated for each of the NTEMP2 final temperature TEMP2 requested. Note that after each step INEW contains the new data and IOLD contains the previous data. If the "bootstrap" option is set (ISTRAP=1), these units are interchanged. For this option, TEMP2(IT) is always obtained from TEMP2(IT-1). Because of thinning, the broadening runs faster at each step. The accumulation of error is usually not a problem. For ISTRAP=0, TEMP1 is used for the starting temperature every time.

The broadening and thinning calculations are directed by FILE3 (except for the parallel processing and input/output this subroutine is taken from SIGMA1). The routine loads data into the appropriate core pages from scratch file IOLD,

calls BROADN to broaden it, calls THINB to thin it, and writes the broadened and thinned results onto scratch file INEW.

BROADN is also nearly unchanged from SIGMA1. The energy grid points just loaded into E by FILE3 are converted to the dimensionless variables x and y [see Eq. (6)]. A loop is then set up over the y values in the center page. Referring to Eqs. (7) and (8), the sum is accumulated for x_i below y until the terms become insignificant. If necessary, the cross section is extended to $E=0$ as a constant. The $H_n(a,b)$ functions are produced by FUNKY and HUNKY using either Eq. (10) or the alternate method of direct expansion. The calculation is ordered to take advantage of previously computed values of F_n . A similar loop is performed for $x_i > y$. For low energies, the term $\sigma^*(-y)$ is then computed and added to the sum. The broadened cross sections are stored in SB, and the energies are converted back to eV and stored in EB.

The THINB routine follows SIGMA1 except that no thinning is performed above the minimum threshold (THNMAX). All points in a given interval are tested for their deviation from the straight line connecting the endpoints. If all are within tolerance, all can be removed; the interval is extended to one more point, and the test is repeated. If any point fails, the last point is accepted as an output point. The thinned data remain in EB and SB for FILE3.

HUNKY has been modified to implement the alternate $H_n(a,b)$ calculation when necessary (see HNABB). When using the direct method, F_n values from the previous step are used in the difference of Eq. (10) and FUNKY is called to get the new values. The A_i and B_i of Eq. (18) are related to the S1 and S2 here.

FUNKY evaluates $F_n(a)$ by the recursion formula of Eq. (12) using a rational approximation to the reduced complementary error function.²

HNABB implements the alternate calculation described by Eqs. (14)-(17). The series expansion is continued until about six significant figures are guaranteed (see EPS in HNABB). HNABB is called when only four significant figures are reliable in HUNKY (see TOLER in HUNKY).

D. User Input

The following input instructions have been copied from the comment cards at the start of BROADR and are also given in Vol. I: User's Manual.


```

C *---INPUT SPECIFICATIONS (FREE FORMAT)-----*
C *
C * CARD 1
C *   NIN      INPUT PENDF TAPE
C *   NOUT     OUTPUT PENDF TAPE
C * CARD 2
C *   MAT1     MATERIAL TO BE PROCESSED
C *   NTEMP2   NUMBER OF FINAL TEMPERATURES (MAXIMUM=6)
C *   ISTART   RESTART (0 NO, 1 YES)
C *   ISTRAP   BOOTSTRAP (0 NO, 1 YES)
C *   TEMP1    STARTING TEMPERATURE FROM NIN
C *   ERRTHN   FRACTIONAL TOLERANCE FOR THINNING
C *   THNMAX   MAX. ENERGY FOR BROADENING AND THINNING
C *             (DEFAULT=1 MeV)
C * CARD 3
C *   TEMP2    FINAL TEMPERATURES (DEG KELVIN)
C * CARD 4
C *   MAT1     NEXT MAT NUMBER TO BE PROCESSED WITH THESE
C *             PARAMETERS.  TERMINATE WITH MAT1=0.

```

Note that TEMP1 need not occur on NOUT. The restart option enables the user to add new temperatures to the end of an existing PENDF tape. This option is also useful if a job runs out of time while processing, for example, the fifth temperature. The job can be restarted from the partial NOUT. The first four temperatures will be copied to the new NOUT and broadening will continue for temperature five. The bootstrap option speeds up the code by using the broadened and thinned result for TEMP2(I-1) as the starting point to obtain TEMP2(I). The THNMAX parameter can be used to speed up a calculation or to prevent the broadening of inappropriate data such as unresolved cross sections or evaluations using histogram or sharp triangular representations at high energies (for example, ENDF/B-V lead).

The following example prepares a single output tape containing AM-241 and AM-243 from ENDF/B-IV at two temperatures each.

```

*BROADR*
20 21
1056 2 0 1 0. .001/
300. 1200.
1057
0/

```

Unit 20 contains a RECONR-generated BCD PENDF tape containing 0 K cross sections for the two isotopes. Four materials will be generated on unit 21 with 0.1% accuracy. The default THNMAX of 1 MeV will be used.

E. Error Messages

BROADR***TOO MANY LOW THRESHOLD REACTIONS

The current limit is 9. Check TT, MTR, and NTT in BROADR, TT in FILE3, and SBT in BROADN. Too many reactions might also strain the total storage (see A and NAMAX in BROADR).

BROADR***INPUT AND OUTPUT MUST BE SAME MODE

Use coded to coded, or blocked binary to blocked binary. The latter is much faster due to the several tape copies performed in BROADR.

BROADR***STORAGE EXCEEDED

Insufficient storage to update dictionary. Increase NWSCR in BROADR.

WARNING***BROADENING TRUNCATED AT A = ---

The page is too small for the temperature difference requested. Increase total storage available or repeat the calculation with smaller temperature steps and ISTRAP = 1. The normal maximum size of A is 4.0 and A is inversely proportional to $T_i - T_{i-1}$.

F. Input/Output Units

10 NSCR1 in BROADR. Contains the ENDF/B data at the initial temperature.

12/13 IOLD/INEW in BROADR. Contains union energy grid and low threshold reactions.

20-26 User's choice for NIN and NOUT to link with other modules.

Units 12 and 13 will always be binary. Unit 10 will have the same mode as NIN and NOUT (binary mode is recommended).

G. Storage Allocation

All storage used is divided in the most efficient way possible. The container array in /STORE/ and NAMAX should be made as large as possible. NBUF can be increased or decreased at will--large values will give faster execution. NWSCR depends on the size of the ENDF/B dictionary and 1000 words is sufficient for all current evaluations.

H. References for BROADR

1. D. E. Cullen, "Program SIGMA1 (Version 77-1): Doppler Broaden Evaluated Cross Sections in the Evaluated Nuclear Data File/Version B (ENDF/B) Format," Lawrence Livermore National Laboratory report UCRL-50400, Vol. 17, Part B (1977).
2. M. Abramowitz and I. Stegun, Handbook of Mathematical Functions (Dover Publications, New York, 1965).

XII. HEATR

The HEATR module generates pointwise heat production cross sections and radiation damage energy production for specified reactions and adds them to an existing PENDF tape. The heating and damage numbers can then be easily group averaged, plotted, or reformatted for other purposes. An option of use to evaluators checks ENDF/B files for neutron/photon energy-balance consistency. The advantages of HEATR include:

- Heating and damage are computed in a consistent way.
- All ENDF/B neutron and photon data are used.
- Kinematic checks are available to improve future evaluations.

A. Theory of Nuclear Heating

Heating is an important parameter of any nuclear system. It may represent the product being sold--as in a power reactor--or it may effect the design of peripheral systems such as shields and structural components.

Nuclear heating can be conveniently divided into neutron heating and photon heating (see Fig. 1). The neutron heating at a given location is proportional to the local neutron flux and arises from the kinetic energy of the charged products of a neutron induced reaction (including both charged secondary particles and the recoil nucleus itself). Similarly, the photon heating is proportional to the flux of secondary photons transported from the site of previous neutron reactions. It is also traceable to the kinetic energy of charged particles (for example, electron-positron pairs and recoil induced by photoelectric capture).

Heating, therefore, is often described by the KERMA¹ (Kinetic Energy Release in Materials) factors $k_{ij}(E)$ defined such that the heating rate in a mixture is given by

$$H(E) = \sum_i \sum_j \rho_i k_{ij}(E) \phi(E) \quad (1)$$

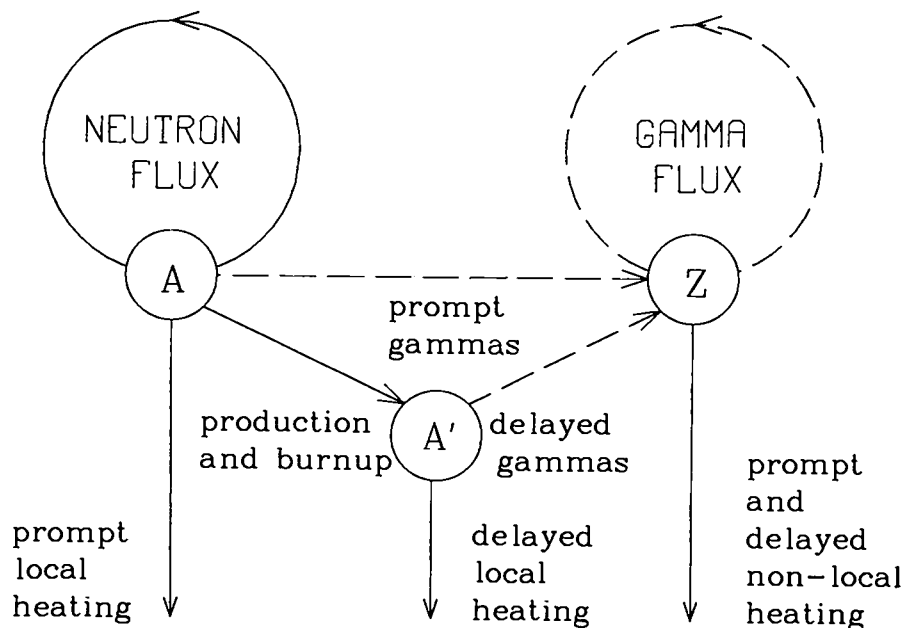


Fig. 1.

Components of nuclear heating. HEATR treats the prompt local neutron heating only.

where ρ_i is the number density of material i , $k_{ij}(E)$ is the kerma factor for material i and reaction j at incident energy E , and $\phi(E)$ is the neutron or photon scalar flux at E . Kerma is used just like a microscopic reaction cross section except that its units are energy \times cross section (eV-barns for HEATR).

The "direct method" for computing the kerma factor is

$$k_{ij}(E) = \sum_{\ell} \bar{E}_{ij\ell}(E) \sigma_{ij}(E) \quad (2)$$

where the sum is carried out over all charged products of the reaction including the recoil nucleus, and $\bar{E}_{ij\ell}$ is the total kinetic energy carried away by the ℓ th species of secondary charged particle. Unfortunately, ENDF/B does not include the detailed spectral information needed to evaluate Eq. (2).

For this reason, NJOY computes most kerma factors by the "energy-balance method".² The energy allocated to neutrons and photons is simply subtracted from the available energy to obtain the energy carried away by charged particles:

$$k_{ij}(E) = (E + Q_{ij} - \bar{E}_{ijn} - \bar{E}_{ij\gamma})\sigma_{ij}(E) \quad , \quad (3)$$

where Q_{ij} is the mass-difference Q value for reaction j , \bar{E}_n is the total energy of secondary neutrons including multiplicity, and \bar{E}_γ is the energy of secondary photons including photon yield.

This method is well suited for use with ENDF/B, which contains neutron and photon spectral data but not the particle spectra required by the direct method. The disadvantage of this method is that the kerma factor sometimes depends on a difference between large numbers. In order to obtain accurate results, extreme care must be taken with the evaluation to ensure that photon and neutron yields and average energies are consistent. In fact, the lack of consistency in ENDF/B-IV often reveals itself as negative kerma factors.

However, this is not always the defect it seems to be. It must be remembered that heating has both neutron and photon components. A negative kerma might indicate that too much energy has been included with the photon production in the evaluation. This will result in excessive photon heating if most of the photons stay in the system. However, the negative kerma will have just the right magnitude to cancel this excess heating. The energy-balance method guarantees conservation of total energy in large homogeneous systems.

In this context, large and homogeneous means that most neutrons and photons stay in their source region. It is clear that energy-balance errors in the evaluation affect the spatial distribution of heat and not the total system heating when the energy-balance method is employed.

A final problem with the energy-balance method occurs for the elemental evaluations in ENDF/B. Isotopic Q values and cross sections are not available in the files. It will usually be possible to define quite adequate cross sections, yields, and spectra for the element. However, it is clear that the available energy should be computed with an effective Q given by

$$\bar{Q} = \frac{\sum_i \rho_i \sigma_i Q_i}{\sum_i \rho_i \sigma_i} , \quad (4)$$

where ρ_i is the atomic fraction of isotope i in the element. This number is energy dependent and can be represented only approximately by the single constant Q allowed in ENDF/B.

For elastic scattering, the neutron kerma factor can be directly evaluated without reference to photon data. For other reactions, conservation of momentum and energy can be used to estimate the kerma or to compute minimum and maximum limits for the heating. HEATR includes an option that tests the energy-balance kerma factors against these kinematic limits, thereby providing a valuable test of the neutron-photon consistency of the evaluation. If the energy-balance heating numbers for a particular isotope should fail these tests, and if the isotope is important for a "small" system, an improved evaluation is probably required. The alternative of making ad hoc fixes to improve the local heat production is dangerous because the faults in the neutron and/or photon data revealed by the tests may lead to significant errors in neutron transport and/or photon dose and nonlocal energy deposition.

In practice, an exception to this conclusion must be made for the radiative capture reaction (n,γ) . The difference between the available energy $E+Q$ and the total energy of the emitted photons is such a small fraction of $E+Q$ that it is difficult to hold enough precision to get reasonable recoil energies. Moreover, the emitted photons cause a component of recoil whose effect is not normally included in evaluated capture spectra. Finally, the "element problem" cited above is especially troublesome for capture because the available energy may change by several MeV between energies dominated by resonances in different isotopes of the element, giving rise to many negative or absurdly large heating numbers. These problems are more important for damage calculations (see below) where the entire effect comes from recoil and the compensation provided by later deposition of the photon energy is absent.

For these reasons, HEATR estimates the recoil due to radiative capture using conservation of momentum. The recoil is the vector sum of the "kick"

caused by the incident neutron and the kicks due to the emission of all subsequent photons. Assuming that all photon emission is isotropic and that the directions of photon emission are uncorrelated, the photon component of recoil depends on the average of E_γ^2 over the entire photon spectrum

$$E_R \cong \frac{E}{A+1} + \frac{\overline{E_\gamma^2}}{2(A+1)mc^2} , \quad (5)$$

where mc^2 is the neutron mass-energy. The second term is important below 25-100 keV. This formula gives an estimate that works for both isotopes and elements and has no precision problems. However, it does not explicitly conserve energy, and isotopes with bad capture photon data can still cause problems.

B. Theory of Damage Energy

Damage to materials caused by neutron irradiation is an important design consideration in fission reactors and is expected to be an even more important problem in fusion power systems. There are many radiation effects that may cause damage; for example, direct heating, gas production (for example, helium embrittlement), and the production of lattice defects.

A large cluster of lattice defects can be produced by the primary recoil nucleus of a nuclear reaction as it slows down in a lattice. It has been shown that there is an empirical correlation between the number of displaced atoms (DPA) and various properties of metals such as elasticity. The number of displaced atoms depends on the total available energy E_a and the energy required to displace an atom from its lattice position E_d . Since the available energy is used up by producing pairs,

$$DPA = \frac{E_a}{2E_d} . \quad (6)$$

The values of E_d used in practice are chosen to represent the empirical correlations, and a wide range of values is found in the literature^{3,4} (see Table I for some examples). The energy available to cause displacements is what HEATR calculates. It depends on the recoil spectrum and the partition of recoil energy between electronic excitations and atomic motion. The partition function used was given by Robinson⁵ based on the electronic screening theory of Lindhard⁶ (see Fig. 2). The results are suitable for metals only.

The damage output from HEATR is the damage energy production cross section (eV-barns). As in Eq. (1), multiplying by the density and flux gives eV/s. Dividing by $2E_d$ gives displacements/s.

TABLE I
TYPICAL VALUES FOR THE ATOMIC DISPLACEMENT ENERGY NEEDED TO COMPUTE DPA
(displacements per atom)

<u>Material</u>	<u>Energy (eV)</u>
Aluminum	33.8 ^a
Copper	37.5 ^a
Stainless Steel	50.0 ^a 33 ^b
Titanium	37.5 ^a
Vanadium	50.0 ^a
Tantalum	68 ^b
Niobium	75.0 ^a

^aRef. 3, with an efficiency of 0.8.

^bRef. 4, as given.

C. Computation of Kerma Factors

The ENDF/B files do not usually give photon production data for all partial reactions. Redundant reactions such as nonelastic (MT3) and inelastic (MT4) are often used. It is still possible to compute partial kerma factors for these redundant reactions by reordering Eq. (3) as follows:

$$k_{iJ}(E) = \sum_{j \in J} k_{ij}^n(E) - \sum_{l \in J} \bar{E}_{il\gamma} \sigma_{il}(E) \quad , \quad (7)$$

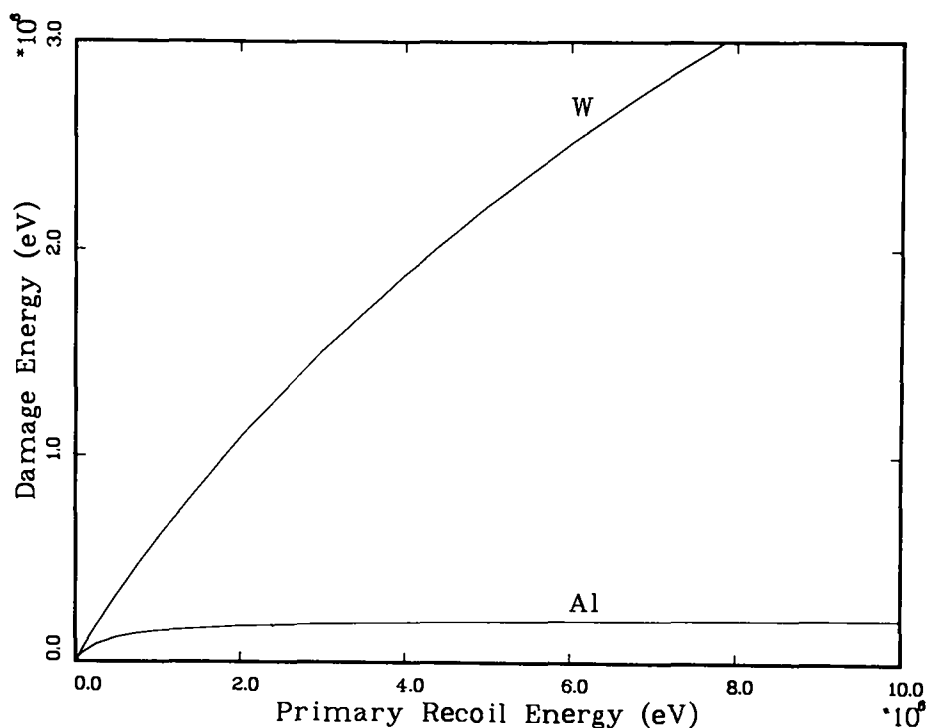


Fig. 2.

Examples of the portion of the primary recoil energy that is available to cause lattice displacements in metallic lattices. The remaining energy leads to electronic excitations.

where j runs over all neutron partials contained in J and ℓ runs over all photon partials in J . The total kerma is well defined, but partial kermas should be used only with caution.

HEATR loops through all the neutron reactions on the ENDF/B tape and computes the neutron contributions needed for the first term. These are

$$k_{ij}^n(E) = [E + Q_{ij} - \bar{E}_{ijn}(E)]\sigma_{ij}(E) \quad (8)$$

The Q value is zero for elastic and inelastic scattering. For (n,n') particle reactions represented by scattering with an LR flag set,⁷ Q is the ENDF "C1" field from MF3. For all other reactions, Q is the "C2" field from MF3. In the case of fission, the component of delayed fission energy from File

1 (MT-458) is subtracted from Q to give a "prompt" result. HEATR allows the user to override any Q value with his own number.

The \bar{E}_n value is defined to include multiplicity in Eq. (8). The multiplicity is either implicit (for example, 2 for n2n) or is retrieved from the ENDF/B file (fission $\bar{\nu}$). The average energy per neutron is computed differently for discrete two-body reactions and continuum reactions.

For elastic and discrete inelastic scattering (MT2, 51-90),

$$\bar{E}_n = \frac{E}{(A+1)^2} (1 + 2rf_1 + r^2) \quad , \quad (9)$$

where f_1 is the center-of-mass average scattering cosine from MF4 and r is the effective mass ratio. For elastic scattering $r = A$, but for threshold scattering

$$r = A \sqrt{1 - \frac{(A+1)S}{AE}} \quad , \quad (10)$$

where S is the negative of the C2 field from MF3.

For continuum scattering, the average energy per neutron is computed from the secondary neutron spectrum g in MF5 using

$$\bar{E}_n(E) = \int_0^{E-U} E' g(E \rightarrow E') dE' \quad , \quad (11)$$

where U is defined in MF5. If g is tabulated (LAW 1, LAW 5), the integral is carried out analytically for each panel by making use of the ENDF/B interpolation laws. For the simple analytic representations (LAW 7, 9, 11), the average energies are known.⁸

The neutron cross sections required by Eq. (8) are obtained from an existing PENDF file (see RECONR and BROADR).

When the neutron sum in Eq. (7) is complete, the code processes the photon production files. If the evaluation does not include photon data, HEATR returns only the first sum. This is equivalent to assuming that all photon energy is deposited locally consistent with the fact that there will be no contribution to the photon transport source from this material.

Discrete photon yields and energies are obtained from MF12 or MF13. Continuum photon data are obtained from MF15, and the average photon energy and E_Y^2 are computed. For radiative capture, the photon term becomes

$$E_Y \sigma_Y = \left(E + Q - \frac{E}{A+1} + \frac{\overline{E_Y^2}}{2(A+1)mc^2} y_Y \right) \sigma_Y , \quad (12)$$

where y_Y is the capture photon yield from MF12. This corrects the capture contribution from Eq. (8) by conservation of momentum. For other reactions, Eq. (8) is sufficient, and the product of \bar{E}_Y , y_Y , and σ_Y is subtracted from the neutron contribution.

Note that if there are no photon files for the evaluation, the resulting kerma factors are equivalent to assuming that all photon energy is deposited locally. The same result can be forced using the LOCAL input parameter.

D. Kinematic Limits

As an option provided mainly as an aid to evaluators, HEATR will compute the kinematic maximum and minimum kerma factors and compare them with the energy-balance result. The formulas used are as follows. For elastic scattering (MT2),

$$\bar{E}_R = \frac{2AE}{(A+1)^2} (1 - f_1) , \quad (13)$$

where \bar{E}_R is the expected recoil energy.

For discrete inelastic scattering (MT51-90), the photon momentum is neglected to obtain

$$\bar{E}_R = \frac{2AE}{(A+1)^2} \left[1 - f_1 \sqrt{1 - \frac{(A+1)E_Y}{AE}} \right] - \frac{E_Y}{A+1} , \quad (14)$$

where $E_Y = -C2$ from MF3. For continuum inelastic scattering (MT91), secondary neutrons are assumed to be isotropic in the LAB system giving

$$\bar{E}_R = \frac{E - \bar{E}_n}{A} , \quad (15)$$

and

$$\bar{E}_Y = \frac{(A-1)E - (A+1)\bar{E}_n}{A} , \quad (16)$$

where \bar{E}_Y is the average photon energy expected for this representation. For radiative capture (MT102),

$$\bar{E}_R = \frac{E}{A+1} + E_K , \quad (17)$$

$$\bar{E}_Y = Q + \frac{AE}{A+1} - E_K , \quad (18)$$

where

$$E_K = \frac{\left[\frac{AE}{A+1} + Q \right]^2}{2M_R c^2} \left\{ 1 - \frac{\left[\frac{AE}{A+1} + Q \right]}{M_R c^2} \right\}, \quad (19)$$

with

$$M_R c^2 = 939.512(A + 1) - Q \quad (20)$$

being the mass-energy in MeV.

For two-body scattering followed by particle emission (MT51-91, LR flag set), a minimum and maximum can be defined

$$(E'_R + E_x)_{\min} = \bar{E}_R, \quad (21)$$

$$(E'_R + E_x)_{\max} = \bar{E}_R + Q + (E_y)_{\max}, \quad (22)$$

where \bar{E}_R is the value from Eq. (10), (11), or (12), Q is the C2 field from MF3, and $(E_y)_{\max}$ is the -C2 field from MF3. In these equations E'_R is the recoil energy and E_x is the energy of the charged product. For absorption followed by particle emission (MT103-120),

$$(E_R + E_x)_{\min} = \frac{E}{A + 1 - x}, \quad (23)$$

$$(E_y)_{\max} = Q + \frac{A - x}{A + 1 - x} E, \quad (24)$$

$$(E_R + E_x)_{\max} = E + Q, \quad (25)$$

where Q is the C2 field from MF3 and x is the particle mass ratio (x = 1 gives a minimum for all reactions). For neutron continuum scattering (MT16, 17, 22-37),

$$(E_R + E_x)_{\min} = 0 \quad \text{and} \quad (26)$$

$$(E_R + E_x)_{\max} = E + Q - \bar{E}_n, \quad (27)$$

where Q is the C2 field from MF3. Finally, for fission (MT18-21, 38), the limits are

$$(E_R)_{\min} = E + Q - \frac{1}{2}\bar{E}_n - 15 \text{ MeV} \quad \text{and} \quad (28)$$

$$(E_R)_{\max} = E + Q - \bar{E}_n, \quad (29)$$

where Q is the prompt fission Q less neutrinos.⁹

These values are intended to be very conservative. Note that E_K is only significant at very low neutron energy. In order to reduce unimportant error messages, a tolerance band is applied to the above limits. If all checks are satisfied, the resulting kerma factors should give good local heating results even when 99.8% of the photons escape the local region.

E. Computation of Damage Energy

The formulas used for calculating damage energy are derived from the same sources as the heating formulas given above, except in this case, the effects of scattering angle do not result in simple factors like f_1 because the Robinson partition function is not linear. Instead, it is calculated as follows:

$$P(E) = \frac{E_R}{1 + F_L (3.4008\varepsilon^{1/6} + 0.40244\varepsilon^{3/4} + \varepsilon)} \quad (30)$$

if $E_R \geq 25.0$ eV and zero otherwise. In Eq. (30), E_R is the primary recoil energy,

$$\varepsilon = E_R/E_L \quad , \quad (31)$$

$$E_L = 30.724 Z_R Z_L (Z_R^{2/3} + Z_L^{2/3})^{1/2} (A_R + A_L)/A_L \quad , \quad (32)$$

$$F_L = \frac{0.0793 Z_R^{2/3} Z_L^{1/2} (A_R + A_L)^{3/2}}{(Z_R^{2/3} + Z_L^{2/3})^{3/4} A_R^{3/2} A_L^{1/2}} \quad , \quad (33)$$

and Z_i and A_i refer to the charge and atomic number of the lattice nuclei (L) and the recoil nuclei (R). The function behaves like E_R at low recoil energies and then levels out at higher energies. Therefore, the damage energy production cross section is always less than the heat production cross section.

For elastic and two-body discrete-level inelastic scattering,

$$E_R(E, \mu) = \frac{AE}{(A+1)^2} (1 - 2M\mu + M^2) \quad , \quad (34)$$

where the "effective mass" is given by

$$M = \sqrt{1 - \frac{A+1}{A} \frac{(-Q)}{E}} \quad , \quad (35)$$

and μ is the center-of-mass scattering cosine. The damage energy production cross section is then obtained from

$$D(E) = \sigma(E) \int_{-1}^1 f(E, \mu) P(E_R[E, \mu]) d\mu \quad , \quad (36)$$

where f is the angular distribution from the ENDF/B File 4. This integration is performed with a 20-point Gaussian quadrature. Discrete-level reactions with LR flags to indicate, for example, $(n, n')\alpha$ reactions are treated in the same way at present. The additional emitted particles are ignored.

Continuum reactions (n, n') give a recoil spectrum

$$E_R(E, E', \mu) = \frac{1}{A} (E - 2\sqrt{EE'} + E') \quad , \quad (37)$$

where E' is the secondary neutron energy, μ is the laboratory cosine, and the photon momentum has been neglected. The damage becomes

$$D(E) = \sigma(E) \int_0^\infty dE' \int_{-1}^1 d\mu f(E, \mu) g(E \rightarrow E') P(E_R[E, E', \mu]) \quad , \quad (38)$$

where g is the secondary energy distribution from ENDF/B File 5. In the code, the angular distribution is defaulted to isotropic, and a 4-point Gaussian quadrature is used for the angular integration. For analytic representations of g , an adaptive integration to 5% accuracy is used for E' ; for tabulated File 5 data, a trapezoidal integration is performed using the energy grid of the file. The same procedure is used for $(n, 2n)$, $(n, n\alpha)$, etc., with no account being taken of the extra emitted particles.

The recoil for radiative capture must include the momentum of the emitted photon below 25-100 keV giving

$$E_R = \frac{E}{A+1} - 2\sqrt{\frac{E}{A+1}} \sqrt{\frac{E_Y^2}{2(A+1)mc^2}} \cos \phi + \frac{E_Y^2}{2(A+1)mc^2} , \quad (39)$$

where ϕ is the angle between the incident neutron direction and emitted photon direction. If subsequent photons are emitted in a cascade, each one will add an additional term of E_Y^2 and an additional angle. A complete averaging of Eq. (39) with respect to $P(E_R)$ would be difficult and would require angular correlations not present in ENDF/B. However, damage calculations are still fairly crude, and an estimate for the damage obtained by treating the neutron "kick" and all the photon kicks independently should give a reasonable upper limit because

$$\int_{-1}^1 D(E_R) d(\cos\phi) \leq D\left(\frac{E}{A+1}\right) + \sum_Y D\left(\frac{E_Y^2}{2M_R c^2}\right) . \quad (40)$$

Finally, for the (n,particle) reactions, the primary recoil is given by

$$E_R = \frac{1}{A+1} (E^* - 2\sqrt{aE^*E_a} \cos\phi + aE_a) , \quad (41)$$

where a is the mass ratio of the emitted particle to the neutron, E^* is given by

$$E^* = \frac{(A+1-a)}{A+1} E , \quad (42)$$

and the particle energy E_a is approximated as being equal to the smaller of the available energy

$$Q + \frac{AE}{A + 1} , \quad (43)$$

or the Coulomb barrier energy

$$\frac{1.029 \times 10^6 zZ}{a^{1/3} + A^{1/3}} \quad (\text{in eV}) , \quad (44)$$

where z is the charge of the emitted particle and Z is the charge of the target. A more reasonable distribution would be desirable, but this one has the advantage of eliminating an integration, and most results are dominated by the kick imparted by the incident neutron anyway. The angular distribution for the emitted particle is taken as isotropic in the lab. At high incident energies, direct interaction processes would be expected to give rise to a forward-peaked distribution, thereby reducing the damage. However, the importance of this effect is also reduced by the dominance of the neutron kick.

Figure 3 gives a typical result of a damage energy production calculation, showing the separate contributions of elastic, inelastic, and absorption processes.

F. Coding Details

The main program starts by reading user input, assigning storage pointers, and locating the desired material on the PENDF tape. INIT is called to examine the dictionary. Flags are set if MF12 or 13 is present and if MT18 or 19 is used. This subroutine also saves the grid of the total cross section (MT1) on the LOADA/FINDA scratch file that will be used to accumulate the kerma factors, damage, and kinematic checks (if requested).

Now NHEAT is called. After allocating itself some temporary storage, it copies MF1 from the ENDF tape to be used for the retrieval of fission $\bar{\nu}$ by CONBAR. A loop is set up over all nonredundant reactions in MF3. For each reaction, the appropriate Q value is chosen, the cross sections are retrieved with GETY1, and the average neutron energies and damage energies are calculated with DISBAR, CONBAR, CAPDAM, and DISDAM. The neutron part of the kerma and the

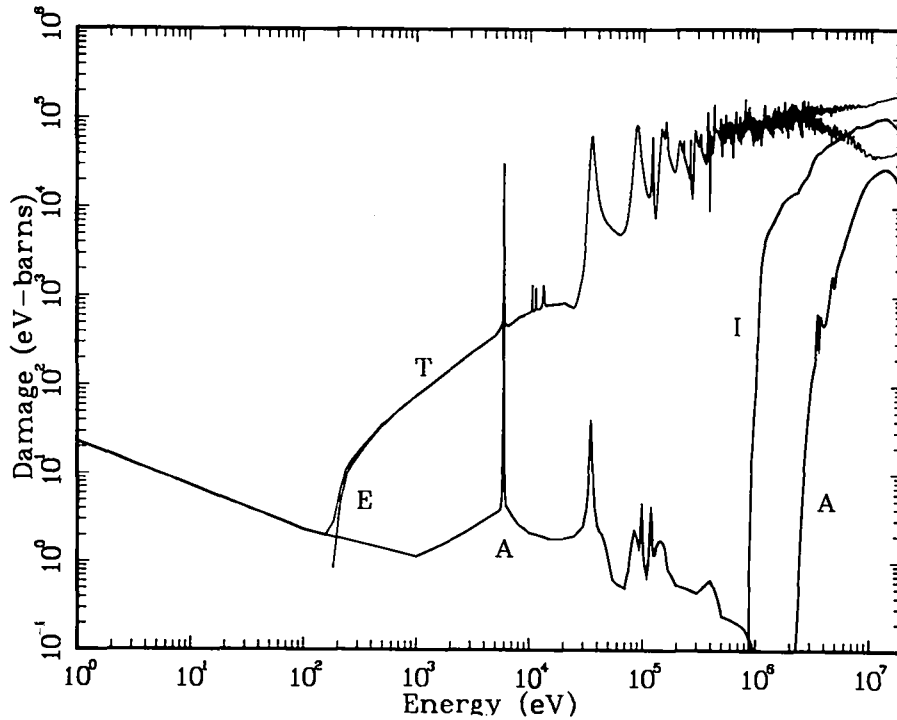


Fig. 3.

Components of radiation damage energy production for ^{27}Al from ENDF/B-V (A is absorption, I is inelastic scattering, E is elastic scattering, and T is total).

damage function are computed and added into the appropriate partial reactions being accumulated on the LOADA/FINDA file. If desired, kinematic limits are computed and added onto the LOADA/FINDA file. This loop is continued for all reactions and all grid energies.

The DISBAR routine is used by NHEAT to compute the average secondary energy for elastic (MT2) or discrete inelastic scattering (MT51-90) using the P_1 scattering coefficient from MF4 (see GETFLE). Similarly, CONBAR computes the average secondary energy and damage energy for continuous distributions described in MF5. Analytic representations use simple formulas coded into ANABAR or a combination of adaptive and Gaussian quadrature in ANADAM. Tabulated data are interpolated from the MF5 table using TABBAR or integrated using trapezoidal and Gaussian quadratures in TABDAM.

If the ENDF/B material includes photon production data, the energy carried away by photons is subtracted from the accumulating kerma factors in GHEAT. The damage cross section is also corrected for photon momentum. First, a scratch

file is prepared containing MF12 and 13. Transition probability arrays are converted using CONVER if present. A loop is set up over all reactions in MF12 and MF13. Tabulated energy distributions are integrated using GAMBAR (both \bar{E}_y and \bar{E}_y^2 are computed for MT102). In order to avoid requiring MF3, MT3 pointwise data, the code uses MT1-MT2 to compute the nonelastic neutron cross section if required.

The final steps are accomplished in HOUT. The partial kermas and damage from the LOADA/FINDA file are recast into TAB1 records and written onto the new PENDF tape using MT numbers from the 300 series for kerma (that is, capture kerma = 300 + 102 = 402) and a special 444 series for damage (444 = total damage, 445 = elastic, 446 = inelastic, and 447 = disappearance). The material dictionary is updated to include the new sections.

If the long print is requested, the Q values, \bar{E}_n , σ , heating, and damage are printed for each neutron reaction on a special coarse energy grid. Similarly, the average photon energy EBAR, yield, cross section, and the photon part of the heating (indicated as a subtraction) are printed. For MF12, MT102 the print is a little different. It shows the photon-induced recoil and damage energy as EGAM and EDAM and the corrected heating as HEATING and DAMAGE, rather than the change due to capture photons as in other reactions. Also, for MF12, MT102 a check is made of the total photon energy computed from MF12 and MF15 versus the approximate available energy $E + Q$, and the percent difference ERR is printed if greater than 1% (modest differences are expected for the light isotopes due to the neglect of recoil). This percent difference will appear as an energy-balance error in a heating calculation for a large system. Finally, a summary print of the partial kerma factors and damage energy cross section on the coarse grid is printed. On option, the kinematic limits are included in this summary print. The coarse energy grid is chosen in NHEAT during the processing of the first reaction. At present, decade steps are used below 1 eV, factor of two steps from 1 eV to 100 keV, quarter-lethargy steps above 100 keV, and ~ 1 MeV steps above 2 MeV.

The kinematic checks are intended for evaluators and other people familiar with ENDF/B photon representations. The MT301 (total) column always makes sense, but partial kermas are only defined for reactions that appear in File 12 or 13; even then, some energy ranges may not be defined. As an example, many files use MT4 and MT102 to represent the photons at low energies and MT3

at high energies; do not pay attention to 304 and 402 above the breakover point. Another example might indicate the care required in interpreting these "error" flags. In ²⁷Al, some of the proton emission is given as pseudo-level (n,n'p) reactions in MT51-90. The corresponding photons are given in MT28. Clearly MT328 makes no sense, and neither does MT304.

G. Input Instruction

The input instructions that follow are reproduced from the comment cards in HEATR (see also Vol. I: User's Manual).

```

C  *---INPUT SPECIFICATIONS (FREE FORMAT)-----*
C  *
C  * CARD 1
C  *   NENDF   UNIT FOR ENDF/B TAPE
C  *   NIN     UNIT FOR INPUT PENDF TAPE
C  *   NOUT    UNIT FOR OUTPUT PENDF TAPE
C  * CARD 2
C  *   MATD    MATERIAL TO BE PROCESSED
C  *   NPK     NUMBER OF PARTIAL KERMA DESIRED (DEFAULT=0)
C  *   NQA     NUMBER OF USER Q VALUES (DEFAULT=0)
C  *   NTEMP   NUMBER OF TEMPERATURES TO PROCESS
C  *           (DEFAULT=0. MEANING ALL ON PENDF)
C  *   LOCAL   0/1=GAMMA RAYS TRANSPORTED/DEPOSITED LOCALLY
C  *           (DEFAULT=0)
C  *   IPRINT  PRINT (0 MIN. 1 MAX. 2 CHECK) (DEFAULT=0)
C  * CARD 3
C  *   MTK     MT NUMBERS FOR PARTIAL KERMA DESIRED
C  *           TOTAL (MT301) WILL BE PROVIDED AUTOMATICALLY.
C  *           PARTIAL KERMA FOR REACTION MT IS MT+300
C  *           AND MAY NOT BE PROPERLY DEFINED UNLESS
C  *           A GAMMA FILE FOR MT IS ON ENDF TAPE.
C  *           SPECIAL VALUES ALLOWED--
C  *           303 NON-ELASTIC (ALL BUT MT2)
C  *           304 INELASTIC (MT51 THRU 91)
C  *           318 FISSION (MT18 OR MT19, 20, 21, 38)
C  *           401 DISAPPEARANCE (MT102 THRU 120)
C  *           DAMAGE ENERGY PRODUCTION VALUES--
C  *           444 TOTAL
C  *           445 ELASTIC (MT2)
C  *           446 INELASTIC (MT51 THRU 91)
C  *           447 DISAPPEARANCE (MT102 THRU 120)
C  *           CARDS 4 AND 5 FOR NQA GT 0 ONLY
C  * CARD 4
C  *   MTA     MT NUMBERS FOR USERS Q ONLY
C  * CARD 5
C  *   QA      USER SPECIFIED Q VALUES (EV)
C  *

```

As an example, consider a HEATR run for ENDF/B-IV U-235 with partial kermas and user-specified Q values:

```
*HEATR*
20 21 22
1261 4 3/
303 304 318 402
19 20 21
172.65E6 172.65E6 172.65E6
```

The list of partial kermas is obtained by seeing which reactions appear in MF12 and MF13. The total (MT301) is provided automatically. The PENDF tape will have partial kermas for 301, 303, 304, 318, and 402, but the values will not be printed.

H. Error Messages

HEATR***REQUESTED TOO MANY KERMA MTS

8 values in addition to MT301 are allowed with kinematic checks, otherwise 26 can be requested.

HEATR***REQUESTED TOO MANY Q VALUES

Limited to 30 only.

HEATR***MODE CONVERSION NOT ALLOWED BETWEEN NIN AND NOUT

Both units must be BCD (positive) or blocked binary (negative).

NHEAT***NEUTRON BINDING ENERGY FOR SEQUENTIAL N2N MISSING. Q'S SHOULD BE ENTERED ON A DATA CARD AS A NEG. NO. IN EV

Self-explanatory. Reflects a problem in the ENDF/B evaluation for Be-9.

NHEAT***STORAGE EXCEEDED

Insufficient storage for diagnostic energy grid. See ELIST (100) and ILMAX.

CONBAR***NKTOT GT NKMAX

More than 12 subsections found. See NKMAX and D1, D2, E1, E2, and LOC, all dimensioned 12.

CONBAR***TABULATED SUBSECTION MUST BE LAST

Required by organizational problems. This situation is satisfied in vers. IV and V. Other evaluations may need to be modified.

CONBAR***INSUFFICIENT STORAGE FOR RAW ENDF DATA.

Main container array is too small. Increase/STORE/ and NAMAX in HEATR.

HGTYLD***ILLEGAL LND

Assumes a maximum of six time groups for delayed neutrons.

HGTYLD***STORAGE EXCEEDED

Increase NWMAX in NHEAT (currently 2500).

TABBAR***CODED FOR LF=1 AND LF=5 ONLY

Self-explanatory. Should not occur.

HGTFLE***DESIRED ENERGY ABOVE HIGHEST ENERGY GIVEN

Fault in the evaluation.

HGTFLE***NOT ENOUGH STORAGE FOR RAW DISTRIBUTIONS

Main container array too small. Increase /STORE/ and NAMAX in HEATR.

GETCO***LIMITED TO 21 LEGENDRE COEFFICIENTS

Normal ENDF/B limit.

GETCO***LAB TO CM CONVERSION NOT CODED

Discrete scattering data should be in the center-of-mass system already.

HCONVR***ENERGY READ IN DOES NOT MATCH PREVIOUS ENERGIES

Something is wrong with the data in MF12, L0=2 (transition probability arrays).

GHEAT***NOT CODED FOR L0=2

Will not occur since L0=2 data has been transformed to L0=1 format by CONVER.

GAMBAR***REQUESTED ENERGY AT HIGHEST GIVEN ENERGY

Some fault in MF15 data.

GAMBAR***STORAGE EXCEEDED IN A.

Increase container array /STORE/ and parameter NAMAX in HEATR.

I. Storage Allocation

Variably dimensioned dynamic storage allocation is used for most data. Storage requirements are dominated by the length of MF5 or MF15 for the evaluation. The size of common /STORE/ and the parameter NAMAX in HEATR may be adjusted accordingly. The LOADA/FINDA buffer size NBUF may be decreased or increased at will. The code is currently dimensioned as follows:

100	coarse grid points
30	auxiliary Q values
26	partial kermas (8 when kinematic limits are requested)
12000	total storage

J. References for HEATR

1. M. A. Abdou, C. W. Maynard, and R. Q. Wright, "MACK: A Computer Program to Calculate Neutron Energy Release Parameters (Fluence-to-Kerma Factors) and Multigroup Reaction Cross Sections from Nuclear Data in ENDF Format," Oak Ridge National Laboratory report ORNL-TM-3994 (July 1973).
2. D. W. Muir, "Gamma Rays, Q-Values, and Kerma Factors," Los Alamos Scientific Laboratory report LA-6258-MS (March 1976).
3. T. A. Gabriel, J. D. Amburgy, and N. M. Greene, "Radiation-Damage Calculations: Primary Knock-On Atom Spectra, Displacement Rates, and Gas Production Rates," Nucl. Sci. Eng. 61, 21 (1976).
4. D. G. Doran, "Neutron Displacement Cross Sections for Stainless Steel and Tantalum Based on a Linhard Model," Nucl. Sci. Eng. 49, 130 (1972).
5. M. T. Robinson, in Nuclear Fusion Reactors (British Nuclear Energy Society, London, 1970).
6. J. Lindhard, V. Nielsen, M. Scharff, and P. V. Thomsen, Kgl. Dansk, Vidensk. Selsk, Mat-Fys. Medd. 33 (1963).
7. R. Kinsey, "ENDF-102, Data Formats and Procedures for the Evaluated Nuclear Data File, ENDF," Brookhaven National Laboratory report BNL-NCS-50496 (ENDF-102) (October 1979).
8. ibid, p. 5.13.
9. R. Sher, S. Fiarman, and C. Beck, "Fission Energy Release for 16 Fissioning Nuclides," unpublished data (October 1976).

XIII. THERMR

The THERMR module generates pointwise neutron scattering cross sections in the thermal energy range and adds them to an existing PENDF tape. The cross sections can then be group-averaged, plotted, or reformatted in subsequent modules. Elastic cross sections are generated for hexagonal lattices using an extended version of the method of HEXSCAT¹ and for noncrystalline materials such as CH₂ and ZrH by direct evaluation using the incoherent approximation. Inelastic cross sections and energy-to-energy matrices can be produced for free scatterers or for bound scatterers when ENDF/B scattering functions are available.^{2,3} This function has previously been performed using FLANGE-II.⁴ THERMR has the following advantages over HEXSCAT and FLANGE-II:

- The energy grid for coherent elastic scattering is produced adaptively so as to represent the sharp Bragg edges to a specified tolerance using linear interpolation.
- The secondary energy grid for inelastic incoherent scattering is produced adaptively so as to represent all structure with linear interpolation.
- Incoherent cross sections are computed by integrating the incoherent matrix for consistency.
- Free incoherent scattering is normalized to the Doppler broadened elastic scattering cross section in order to provide an approximate representation of resonance scattering and to preserve the correct total cross section.
- Discrete angle representations are used to avoid the limitations of Legendre expansions.

A. Coherent Elastic Scattering

The thermal coherent scattering from a powdered crystal may be represented as follows:⁵

$$\sigma^{\text{coh}}(E \rightarrow E', \mu) = \sigma_c \frac{\pi \hbar^2}{4MEV_0} \sum_{\substack{|\vec{\tau}| < \tau_{\text{max}} \\ \vec{\tau} \neq 0}} \frac{|F(\vec{\tau})|^2}{\tau N} e^{-2W_D \tau^2} \delta(\mu - \mu_0) \delta(E' - E) \quad (1)$$

where

$$\tau_{\max} = \sqrt{\frac{8ME}{h^2}} \quad , \quad (2)$$

$$\mu_0 = 1 - \frac{\pi^2 \tau^2}{4ME} \quad , \quad (3)$$

and where E is the incident neutron energy, E' is the secondary neutron energy, μ is the scattering cosine in the laboratory reference system, σ_c is the characteristic coherent scattering cross section for the material, M is the target mass, V_0 is the volume of the unit cell, N is the number of atoms per unit cell, F is the form factor, W_D is the Debye-Waller coefficient, and τ is one of the reciprocal lattice wave-vectors.

The sum can be simplified by lumping all terms with the same value of τ together and defining a single factor $f(\tau)$. Then

$$\sigma^{\text{coh}}(E, \mu) = \sigma_c \frac{\pi^2}{4MEV_0} \sum_{\tau \neq 0}^{\tau < \tau_{\max}} f(\tau) \delta(\mu - \mu_0[\tau]) \quad (4)$$

This sum is easily performed for any E if a sorted list of precomputed τ and $f(\tau)$ values is available. As τ gets large, the values of τ get more and more closely spaced. In order to save storage and run time, a range of τ values can be lumped together to give a single effective τ and $f(\tau)$. This device washes out the Bragg edges at high energies while preserving the proper average cross section and angular dependence. The current grouping factor is 5% (see EPS in SIGC).

Lattice constants (given in SIGC for graphite, Be, and BeO), form factor formulas (see FORM) and methods for computing reciprocal lattice vectors were borrowed directly from HEXSCAT.

The energy grid for E is obtained adaptively (see COH). A panel extending from just above one Bragg edge to just below the next higher edge is subdivided by successive halving until linear interpolation is within a specified fractional tolerance (TOL) of the exact cross section at every point. This procedure is repeated for every panel from the first Bragg edge to the specified maximum energy for the thermal treatment (EMAX).

The code actually computes and writes out the average over μ of Eq. (5) (that is, the P_0 cross section). Subsequent codes can deduce the correct discrete scattering angles $\mu_0[\tau]$ from the location of the Bragg edges and the factors $f(\tau)$ from the cross-section step at the Bragg edge (see GROUPE). A typical coherent elastic cross section is shown in Fig. 1.

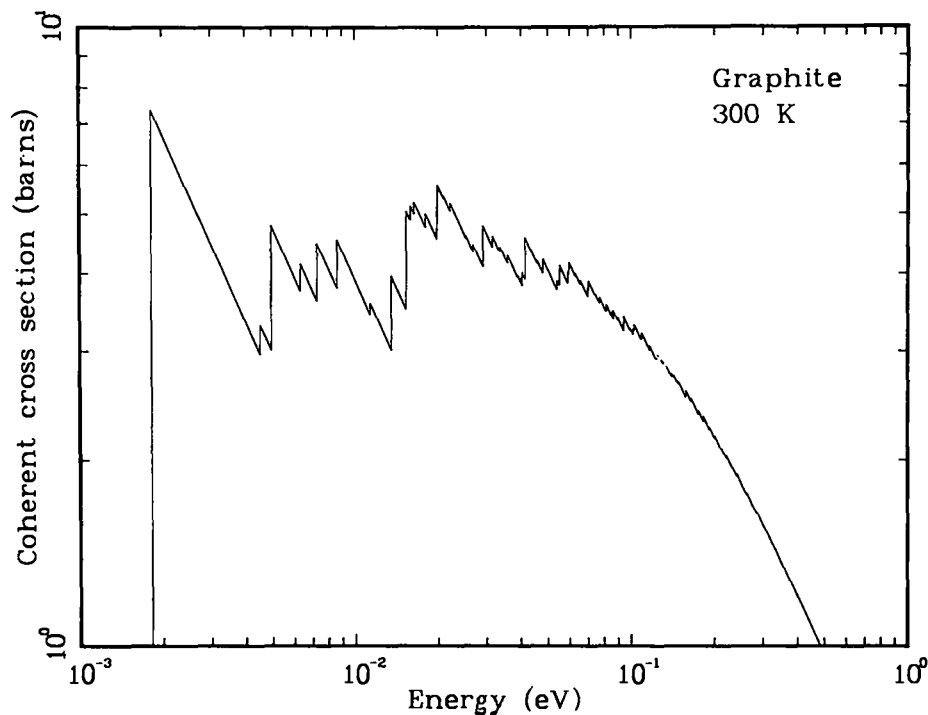


Fig. 1.
Typical behavior of the coherent elastic scattering from a crystalline material as computed by THERMR.

B. Incoherent Inelastic Scattering

In ENDF/B notation, the thermal incoherent scattering cross section is given by

$$\sigma^{inc}(E \rightarrow E', \mu) = \sigma_b \frac{1}{4\pi T} \sqrt{\frac{E'}{E}} e^{-\beta/2} S(\alpha, \beta) \quad , \quad (5)$$

where E is the initial neutron energy, E' is the energy of the scattered neutron, μ is the scattering cosine in the laboratory system, σ_b is the characteristic bound incoherent scattering cross section for the nuclide, T is the Kelvin temperature, β is the dimensionless energy transfer

$$\beta = \frac{E' - E}{kT} \quad , \quad (6)$$

α is the dimensionless momentum transfer

$$\alpha = \frac{E' + E - 2\mu\sqrt{EE'}}{AkT} \quad , \quad (7)$$

k is Boltzmann's constant, and A is the ratio of the scatterer mass to the neutron mass. The bound scattering cross section is usually given in terms of the characteristic free cross section, σ_f ,

$$\sigma_b = \sigma_f \frac{(A+1)^2}{A^2} \quad . \quad (8)$$

The scattering law $S(\alpha, \beta)$ describes the binding of the scattering atom in a material. For a free gas of scatterers with no internal structure,

$$S(\alpha, \beta) = \frac{1}{\sqrt{4\pi\alpha}} e^{-(\alpha^2 + \beta^2)/4\alpha} \quad (9)$$

For binding in solids and liquids, $S(\alpha, \beta)$ for several materials has been computed³ and written in ENDF/B File 7 format.² The scattering law is given as tables of S versus α for various values of β . Any desired values of S can be obtained by interpolation.

If the α or β required is outside of the range of the table in File 7, the differential scattering cross section can be computed using the SCT approximation

$$\sigma_{SCT}^{(E \rightarrow E', \mu)} = \frac{\sigma_b}{4\pi T} \sqrt{\frac{E'}{E}} \frac{e^{-\left\{ \frac{(|\alpha| - \beta)^2 T}{4|\alpha| T_{eff}} + \frac{\beta + |\beta|}{2} \right\}}}{3.544908 \sqrt{|\alpha| \frac{T_{eff}}{T}}}, \quad (10)$$

where T_{eff} is the effective temperature for the SCT approximation. These temperatures are given in Ref. 3; they are usually somewhat larger than the corresponding Maxwellian temperature T . For the convenience of the user, the values of T_{eff} for the common moderators are included as defaults (see input instructions).

THERMR expects the requested temperature T to be one of the temperatures included on the ENDF/B thermal file, or within a few degrees of that value (296 K is used if 300 K is requested). Intermediate temperatures should be obtained by interpolating between the resulting cross sections and not by interpolating $S(\alpha, \beta)$.

The secondary energy grid for incoherent scattering is obtained adaptively (see CALCEM). A stack is first primed with four points; a point at the kinematic down-scattering limit $E(A-1)^2/(A+1)^2$, a point near $E' = E$ or near the expected peak ($E + kT$ if $E < kT$), and a point far out on each wing. These intervals are then subdivided by successive halving until the cross section obtained by linear interpolation is within the specified tolerance of the correct cross

section (from SIGL). The result is easily integrated by the trapezoid rule to find the incoherent cross section at energy E. In this way, all the extreme energy dependence of this function is accurately represented.

The cross section for one particular $E \rightarrow E'$ is the integral over the angular variable of Eq. (5). The angular dependence is obtained by adaptively subdividing the cosine range until the actual angular function (see SIG) is represented by linear interpolation to within a specified tolerance. The integral under this curve is used in calculating the secondary-energy dependence as described above. Rather than providing the traditional Legendre coefficients, THERMR divides the angular range into equally probable cosine bins and then selects the single cosine in each bin, which preserves the average cosine in the bin. These equally probable cosines can be converted to Legendre coefficients easily when producing group constants, and they are suitable for direct use in Monte Carlo codes. For strongly peaked functions such as scattering for $E \gg kT$ when the result begins to look "elastic", all the discrete angles will be bunched together near the scattering angle defined by ordinary kinematics. This behavior cannot be obtained with ordinary P_3 Legendre coefficients. Conversely, if such angles are converted to Legendre form, very high orders can be used. If a direct calculation of Legendre components is desired, reverse the sign of NNL in CALCEM.

The incident energy grid is currently stored directly in the code (see EGRID in CALCEM). The choice of grid for $\sigma^{inc}(E)$ is not critical since the cross section is a slowly varying function of E. Of course, $\sigma^{inc}(E \rightarrow E')$ for one value of E' is a very strongly energy-dependent function. However, this strong energy dependence is partially an artifact of looking at a given E' . The shape of the secondary energy distribution changes slowly whereas the peak tends to follow the line $E' = E$. This behavior implies that a relatively coarse incident energy grid might prove adequate if a suitable method is used to interpolate between adjacent E values. One such interpolation scheme is implemented in GROUPE. The use of discrete angles is especially suitable for this interpolation scheme.

The scattering law for free gas scattering given in Eq. (9) is strictly applicable to scatterers with no internal structure. However, many materials of interest in reactor physics have strong scattering resonances in the thermal

range (for example, ^{240}Pu and ^{135}Xe). The Doppler broadened elastic cross section produced by BROADR is formally correct for a gas of resonant scatterers, but the cross section resulting from Eq. (9) is not. In order to allow for the resonance scattering in a way that at least provides the correct total cross section, HEATR renormalizes the free scattering to the broadened elastic cross section. The secondary energy distribution will still be incorrect.

C. Incoherent Elastic Scattering

Materials such as polyethylene and zirconium hydride exhibit a component of elastic scattering (that is, $E' = E$) that can be treated in the incoherent approximation³

$$\sigma(E, \mu) = \frac{\sigma_b}{2} e^{-\frac{2EW}{A}(1-\mu)}, \quad (11)$$

where σ_b is the characteristic bound cross section and W is the Debye-Waller integral. The energy grid of the elastic cross section is used for E , and the average cross section and equally probable angles are computed using

$$\sigma(E) = \frac{\sigma_b}{2} \frac{1}{2EW} e^{-4EW}, \quad (12)$$

where W is interpolated from tables given in reference 3, and

$$\bar{\mu}_i = \frac{N}{2EW} [e^{-2EW(1-\mu_{i-1})} (2EW\mu_{i-1} - 1) - e^{-2EW(1-\mu_{i-1})} (2EW\mu_{i-1} - 1)] / (1 - E^{-4EW}),$$

where

$$\mu_i = 1 + \frac{1}{2EW} \ln \left[\frac{1 - e^{-4EW}}{N} + e^{-2EW(1-\mu_{i-1})} \right] \quad (14)$$

is the upper limit of one equal probability bin and $\bar{\mu}_i$ is the selected discrete cosine in this bin. Here N is the number of bins and μ_0 is -1.

D. Coding Details

The procedure begins in THERMR with the reading of the user's input. The required ENDF tape (NENDF) is only used for MF7 data;^{*} it can be set to zero if only free scattering is needed. Similarly, MATDE is the material number on the MF7 tape and can be set to zero for free problems. The ENDF MF7 format only gives " $M_0 \sigma_{f_0}$ ", the product of the free scattering cross section for the principle scatterer and the number of principle scatterer atoms in the molecule. As a result, THERMR needs the parameter NATOM to obtain the effective microscopic cross section (for example, for H in H₂O, use NATOM = 2).

THERMR then finds the desired material on the input PENDF and ENDF tapes. It will automatically loop over NTEMP materials on NIN. The input tape must have been through BROADR. The elastic cross section at the current temperature is saved on a LOADA/FINDA scratch file to be used for normalizing free scattering if necessary. On option, THERMR computes elastic and/or inelastic cross sections by calls to COH, IEL, and CALCEM. The results are written onto the output PENDF tape by PEND.

Some alteration of ENDF/B formats and conventions was required to accommodate thermal cross sections. The incoherent inelastic cross sections fit well into MF=3 using MTREF (see user input). The coherent or incoherent elastic cross section (if present) uses MTREF+1. Other modules of NJOY expect that thermal MT numbers will be between 200 and 250. The incoherent energy-to-energy matrix is stored in MF6 (coupled angle-energy distributions). The original ENDF formats are not well-suited to this application because secondary angle and energy are not tightly coupled as required by the physics of the

* These are tapes in the 320 series available from the National Nuclear Data Center at Brookhaven National Laboratory.

problem. Three new options have been defined: LTT=5 for reordered discrete-angle inelastic transfer cross sections, LTT=6 for discrete-angle elastic data, and LTT=7 for coherent elastic reactions.

The format for LTT=5 is (in standard ENDF/B notation)

```
[MAT,6,MT/ZA,AWR,0,LTT,0,0]HEAD
[MAT,6,MT/TEMP,0.,0,NNR,NNE/Eint]TAB2
<subsections for each of the NNE values of incident energy E>
[MAT,6,0/0. 0. 0 0 0 0 ]SEND
```

The structure of a subsection for LTT=5 is

```
[MAT,6,MT/0.,E,0,0,(NL+2)*NEP NL+2/
E1',f1,μ1,μ2,...μNL,
E2',f2...
ENEP',fNEP...]LIST
```

where E is the incident energy, NL is the number of discrete angles (currently 8), and NEP is the number of secondary energy (E') values. For each E' value, the normalized scattering function is given, where

$$\int f(E \rightarrow E') dE' = 1 \quad . \quad (15)$$

This is followed by the NL discrete cosines. The table continues for each of the other E' values. This format also works for Legendre coefficients (set NNL positive in CALCEM) except that in this case NL is the Legendre order (that is, 3 for P₃ distributions), f₁ is 1, and the μ_i are replaced by the P₁, P₂, P₃... Legendre coefficients.

The format for LTT=6 is the same as above except NEP = 1 because E' = E for elastic scattering. The normalized distribution reduces to f₁ = 1.

The format for LTT=7 is just provided to hold a position in File 6 because all the necessary information is implicit in File 3. The structure used is

```
[MAT,6,MT/ZA,AWR,0,LTT,0,0]HEAD  
[MAT,6,MT/ZA,AWR,0,0,0,NBRAGG]CONT,
```

where NBRAGG is the number of Bragg edges used in the cross-section calculation.

In subroutine COH, the energy grid is determined adaptively and stored onto the same LOADA/FINDA scratch file used for the elastic cross section. The elastic cross section is converted to the coherent grid using Lagrange interpolation (see TERP). The structure of the record stored on the scratch file is

[energy / static elastic / incoherent inelastic / coherent elastic].

Coherent cross sections at a given energy E are computed by SIGC. If this is the first entry ($E=0$), the appropriate lattice constants are selected. Then the reciprocal lattice wave vectors and structure factors are computed, sorted, and stored for later use. On a normal entry ($E>0$), the stored list is used to accumulate the sums of Eq. (4).

Incoherent elastic cross sections are computed in subroutine IEL. The appropriate Debye-Waller integrals are given in data statements and adjusted to the specified temperature using TERP. The bound cross sections are also set in the coding. The angle-integrated cross section is computed analytically on the grid of the static elastic cross section and written back onto the LOADA/FINDA scratch file in the same slot used for coherent elastic as described (both never occur in the same material). The discrete equally probable cosines are cast into LTT=7 format and written onto a scratch tape for use by TPEND.

Incoherent cross sections and matrices are generated in CALCEM. On the first entry, the ENDF/B scattering law is read in or parameters are set for free scattering. On subsequent entries, the adaptive loop to determine the secondary energy grid is carried out. The required cross sections and discrete cosines are returned by SIGL, which uses SIG to compute the differential cross sections. As each σ versus E' curve is computed, it is put directly into the modified MF6 format and written onto a scratch file. At the same time, the incoherent cross section is accumulated by trapezoidal integration, interpolated onto the energy grid of the LOADA/FINDA scratch file, and stored. If free scattering has been selected, the elastic cross section is stored in the incoherent slot. This process is repeated for each energy in the incident energy grid (EGRID).

Incoherent inelastic scattering cross sections and discrete cosines are computed in SIGL. The stack for the adaptive reconstruction of the angular distribution for a given $E \rightarrow E'$ is primed with $\mu = -1$, $\mu = +1$, and the angle for static (that is, $T = 0$) scattering. The top interval on the stack is subdivided by halving until the actual cross section computed by SIG is within a specified tolerance of a linear interpolate. As each panel is converged, its area is added to the accumulating cross section. On convergence, the fraction of the cross section corresponding to each equal-probability bin is computed, and the linearization process is repeated to find the bin boundaries and discrete cosines. Note that Legendre coefficients can be computed in this routine from the discrete cosines.

Finally, PEND is called to prepare the output tape. The dictionary is updated to account for the new sections that are being added. File 3 is located, and the cross sections stored on the LOADA/FINDA scratch file are retrieved, formatted, and written to the output tape. Note that the elastic cross section in MT2 and the total cross section in MT1 are not changed from their static values, nor is the union grid updated. As a result, MT200-250 must be considered supplemental. Subsequent modules could ignore them or use them in place of the static values. Also note that it is possible to run THERMR several times with different values of MTREF. The result would be one PENDF tape containing static cross sections and several different binding states that can be selected at will (for example, MT2 = static hydrogen, MT201 = free hydrogen, MT202 = hydrogen in water, and MT203 = hydrogen in polyethylene, all on one PENDF tape).

File 6 matrices are read from a scratch tape (NSCR) in ENDF format, normalized, and written back onto the final tape. Since free incoherent scattering was set equal to elastic scattering in CALCEM, the approximate resonance correction of the matrix is now complete.

E. Input Instructions

The following input instructions have been copied from the comment cards in HEATR (see also Vol. I: User's Manual).

```

C *---INPUT SPECIFICATIONS (FREE FORMAT)-----*
C *
C * CARD 1
C * NENDF ENDF/B TAPE FOR MF7 DATA
C * NIN OLD PENDF TAPE
C * NOUT NEW PENDF TAPE
C * CARD 2
C * MATDE MATERIAL DESIRED ON ENDF TAPE
C * MATDP MATERIAL DESIRED ON PENDF TAPE
C * NBIN NUMBER OF EQUI-PROBABLE ANGLES
C * NTEMP NUMBER OF TEMPERATURES
C * IINC INELASTIC OPTIONS
C * 0 NONE
C * 1 COMPUTE AS FREE GAS
C * 2 RESERVED
C * 3 COMPUTE S(A,B) AND MATRIX
C * 4 READ S(A,B) AND COMPUTE MATRIX
C * ICOH ELASTIC OPTIONS
C * 0 NONE
C * 1 GRAPHITE
C * 2 BERYLLIUM
C * 3 BERYLLIUM OXIDE
C * 11 POLYETHYLENE
C * 12 H(ZRH)
C * 13 ZR(ZRH)
C * NATOM NUMBER OF PRINCIPAL ATOMS
C * MTREF MT FOR INELASTIC REACTION (201-250 ONLY)
C * IPRINT PRINT OPTION (0=MINIMUM. 1=MAXIMUM.
C * 2=MAX. NORMAL + INTERMEDIATE RESULTS)
C * (DEFAULT=0)
C * CARD 3
C * TEMPR TEMPERATURES (KELVIN)
C * CARD 4 FOR IINC=4 ONLY
C * EFTEMP EFFECTIVE TEMPERATURES FOR SHORT COLLISION TIME
C * (DEFAULT FOR EACH TEMPERATURE IS STANDARD VALUE
C * FROM GENERAL ATOMIC REPORT IF AVAILABLE.
C * OTHERWISE MATERIAL TEMPERATURE)
C * CARD 5
C * TOL TOLERANCE
C * EMAX MAXIMUM ENERGY FOR THERMAL TREATMENT
C *

```

The following sample problem illustrates producing thermal cross sections for hydrogen in water. It assumes that a previous RECONR/BROADR run prepared a three-temperature PENDF tape on unit 23 in blocked binary mode. ENDF/B-III tape 320 was mounted on unit 26.

```

*THERMR*
26 -23 -24
1002 1269 8 3 4 0 2 201 0
300. 500. 600.
/
.01 2.0

```

Note that default effective temperatures are used. The tape on unit 24 will contain MF3/MT201 and MF6/MT201, which can be requested in GROUPR.

A calculation of both free and graphite cross sections for ENDF/B-IV carbon would go as follows:

```

*THERMR*
0 -23 -24
0 1274 8 3 1 0 1 201 0
300. 900. 1200.
.02 3.7
*THERMR*
26 -24 -25
1065 1274 3 3 4 1 1 209 0
300. 900. 1200.
/
.02 3.7

```

The output tape on unit 25 will contain the following new sections:

```

MF3/MT201  free carbon incoherent xsec
3 209  graphite incoherent xsec
3 210  graphite P0 coherent cross section
6 201  free carbon incoherent matrix
6 209  graphite incoherent matrix
6 210  graphite coherent MF6

```

These reactions could all be averaged using GROUPR. Subsequent formatting modules could then be used to select the desired scatterer and merge it with the static data.

F. Error Messages

THERMR***IINC = 2 or 3 NOT PROGRAMMED

These are future options.

THERMR***MODE CONVERSION NOT ALLOWED

NIN and NOUT must both be binary or both be coded.

THERMR***NIN = 0

An input PENDF tape is required.

THERMR***ILLEGAL REFERENCE MT

Restricted to MT201-250.

THERMR***MAT AND TEMP NOT ON TAPE

Check input instructions against contents of thermal tape.

COH***TOO MANY LEGENDRE ORDERS

The code currently computes only P_0 , but $NL = 1$ in COH can be changed if desired. Code is currently limited to 6 (P_5). If more coefficients are desired, increase NLMAX and the dimensions of the variables S, EJ, and EX in COH, CALCEM, and PEND.

SIGCOH***STORAGE EXCEEDED

Not enough room for lattice factors. Increase /STORE/ and NAMAX in THERMR.

SIGCOH***ILLEGAL LAT

Only three lattices are coded so far. To add others, insert the constants in SIGC and form factor formulas in FORM.

IEL***UNKNOWN MATERIAL IDENTIFIER

Only three options are coded so far. To add others, insert DATA statements for the Debye-Waller integrals and values for the bound cross sections.

CALCEM***NL TOO LARGE FOR BINNING

Increase NLMAX (now 17) and the dimensions of Y and YT.

CALCEM***DESIRED TEMPERATURE NOT FOUND

Requested temperatures do not agree with those on NIN from a previous BROADR run.

CALCEM***STORAGE EXCEEDED

Increase NWSCR in THERMR. This may cause a STORAG error that requires /STORE/ and NAMAX to be increased.

SIG***ILLEGAL OPTION

Only tabulated $S(\alpha, \beta)$ and free gas are coded at this time.

SIGL***NEGATIVE DISCRIMINANT
SIGL***NO LEGAL SOLUTION

Having trouble solving equation for the boundary of a bin.

TPEND***DID NOT FIND TEMP ON NIN

Temperatures requested for THERMR are not consistent with those on the input PENDF tape.

TPEND***STORAGE EXCEEDED

Increase NWSCR in THERMR.

TPEND***CROSS SECTION = 0

Thermal cross section of zero can not be used to normalize the distribution.

G. Input/Output Units

The following logical units are used.

- 10/11 IOLD/INEW in THERMR. Also used in COH, READEM, CALCEM, and TPEND. Used for the LOADA/FINDA scratch file that saves the energy grid and reaction cross sections.
- 12 NSCR in THERMR. Also used in CALCEM and TPEND. Contains the scattering matrix before normalization.
- 13 NSCR2 in THERMR and TPEND. Contains data from NIN that are to be simply copied to NOUT.
- 20-99 User's choice for NENDF, NIN, NOUT, and NREAD (IINC = 2 only) to link with other modules. No mode conversion between NIN and NOUT allowed.

Units 10 and 11 are always binary. Units 12 and 13 have the same mode as NIN and NOUT. The user can choose the modes for NENDF, NIN, and NOUT, except NIN and NOUT must have the same mode.

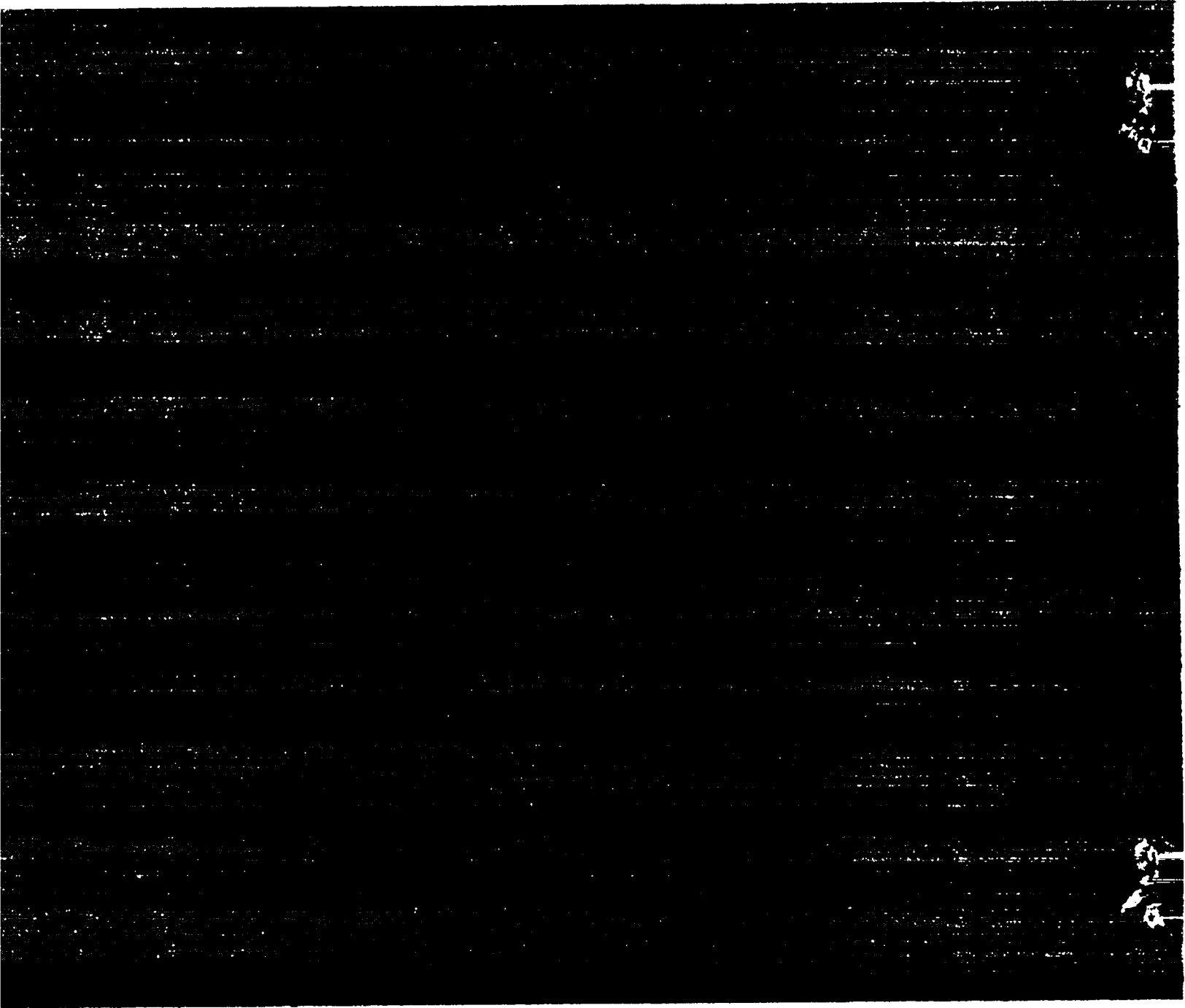
H. Storage Allocation

The storage allocated in THERMR is for the LOADA/FINDA buffers and a scratch array. NBUF may be changed at will; larger values increase I/O efficiency. NWSCR controls the maximum size of the TAB1 records of $\sigma(E \rightarrow E')$ versus E' for incoherent scattering. Hence, it interacts with TOL. The linearization stack (STK) in COH is controlled by IMAX and the number of Legendre components requested (always 1 in the standard version). The current value of IMAX (20)

is sufficient to divide each panel into parts as small as one-millionth of the panel size. The length of the list of lattice factors (FL) in SIGCOH is controlled by the size of the ENDF/B File 7 and /STORE/ must be big enough for the problem.

I. References for THERMR

1. Y. D. Naliboff and J. U. Koppel, "HEXSCAT: Coherent Scattering of Neutrons by Hexagonal Lattices," General Atomic report GA-6026 (1964).
2. R. Kinsey, Ed., "ENDF-102, Data Formats and Procedures for the Evaluated Nuclear Data File, ENDF," Brookhaven National Laboratory report BNL-NCS-50496 (ENDF-102) (1979).
3. J. U. Koppel and D. H. Houston, "Reference Manual for ENDF Thermal Neutron Scattering Data," General Atomic report GA-8774 revised and reissued as ENDF-269 by the National Nuclear Data Center, Brookhaven National Laboratory (1978).
4. H. C. Honeck and D. R. Finch, "FLANGEII (Version 71-1), A Code to Process Thermal Neutron Data From an ENDF/B Tape," Savannah River Laboratory report DP-1278 (ENDF-152) (1971).
5. M. M. R. Williams, The Slowing Down on Thermalization of Neutrons (John Wiley and Sons, New York, 1966).



Los Alamos