BNL 50300
(Reactor Technology - TID-4500)
ENDF 110

ENDF-110
DESCRIPTION OF THE ENDF/B PROCESSING CODES AND RETRIEVAL SUBROUTINES

Edited by Odelli Ozer
NATIONAL NEUTRON CROSS SECTION CENTER
Brookhaven National Laboratory
Upton, New York 11973

September 1967
Revised April 1969
Revised September 1970
Revised June 1971

Contents:

1.	Program	CRECT	5.	"	LISTFC
2.	"	CHECKER	6.	"	DICTION
3.	"	RIGEL	7.	"	SLAVE-3
4.	"	PLOTFB	8.	"	DAMMET

Work performed under the auspices of the U.S. Atomic Energy Commission

NATIONAL NEUTRON CROSS SECTION CENTER

BROOKHAVEN NATIONAL LABORATORY

ASSOCIATED UNIVERSITIES, INC.

under contract with the

UNITED STATES ATOMIC ENERGY COMMISSION

Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22151
Price: Printed Copy $3.00; Microfiche $0.95

1240 copies August 1971

Introduction

The Evaluated Nuclear Data File/B (ENDF/B) is a pool of evaluated nuclear data to be used in reactor or shielding calculations. This data is in general stored on a number of magnetic tapes (or disk files) either in the binary mode or, to facilitate exchange among various centers, in a BCD card image mode. In addition two arrangements of the data (standard or alternate) are possible. A detailed description of the arrangements, the type and ordering of the data as well as the formats involved can be found in the "Data Formats and Procedures Manual for the ENDF/B Neutron Cross Section Library" by M. Drake (BNL-50274(T-601) - ENDF-102, Vol. I).

There are two kinds of programs associated with ENDF/B: those associated with the data file per se (its creation, checking, retrieval, interpretation, display, change of mode or arrangement, etc.) and those which process the data for use in reactor or shielding calculations. This report describes a group of programs falling into the first category and a set of subroutines to be used in other codes which retrieve and process the data from the ENDF/B file for later calculations.

The following programs are described:

Section	Program Name	Function	Approximate Octal Core Requirements
1	CRECT	Correct data on an ENDF/B BCD card image tape.	20
2	CHECKER	Check structure, formats and consistency of data on an ENDF/B BCD card image tape.	54
3	RIGEL	Retrieve, merge, change mode or arrangement of data on ENDF/B tapes. (replaces DAMMET)	45
4	PLOTFB	Interpret data from BCD or Binary ENDF/B tapes and produce edited listings and/or CALCOMP plots.	74
5	LISTFC	Produce interpreted listings of data from an ENDF/B, BCD card image tape.	51

(con't on next page)

Section	Program Name	Function	Approximate Octal Core Requirements
6	DICTION	Produce a new section dictionary for an ENDF/B BCD card image tape.	22
7	SLAVE-3	Retrieval subroutines and selected averages of ENDF/B File 3 data.	101
8	DAMMET	Delete, alter mode and merge ENDF/B tapes (obsolete).	70

The retrieval subroutines included in the listing of the program SLAVE-3 provide a means of transfering ENDF/B records from one storage medium to another as well as the capability of generating, interpolating, combining and integrating one dimensional tabulated functions (TAB1 records).

The codes described in this report can also be used to operate on the ENDF/A libraries since there are no format differences between the "A" and "B" files.

All the codes are written in FORTRAN-IV to provide compatibility with various computers.

This report consists of independent sections each giving a short description of one of the codes with its operating instructions. No program listings are included since these will be part of program documentation to be made available through the Argonne Code Center with the distribution of each code.

Program CRECT

by

Henry C. Honeck
U.S. Atomic Energy Commission
Washington, D. C.

and

Joan Felberbaum
Brookhaven National Laboratory
Upton, New York 11973

March 1967
Revised April 1969

Contents

Program CHECKER

A Program to Check the Data on an ENDF/B BCD Card Image Tape

by

Henry C. Honeck
U.S. Atomic Energy Commission
Washington, D. C.

and

Joan Felderbaum
Brookhaven National Laboratory
Upton, New York 11973

September 1967

Revised August 1970

by

Odelli Ozer
National Neutron Cross Section Center
Brookhaven National Laboratory
Upton, New York 11973

I. Purpose

The Program CHECKER has been written to check the data on an ENDF/B BCD card image format tape. The tape is primarily tested for the correctness of its structure and formats as well as the consistency of the data. No elaborate physics checks are carried out on the data, however some simple tests (such as checks for negative angular distributions, negative probabilities or unreasonable values of $\nu(E)$, etc.) are performed.

The nomenclature used in this description and in the error messages printed by CHECKER follows that used in the ENDF/B Manual.

II. Method

A. General Description:

The input (e.g., card reader) and output (e.g., printer) units NIN = 5 and NOUT = 6 are preset in the main program, the input tape NT* is positioned to the desired section and checked card by card. The cards can either be read in directly or, to avoid fatal program stops due to format errors in the data, the code can be requested to read each card image as 80 Hollerith characters then try to reconstruct the appropriate format.

If an error is detected a message is printed on NOUT. Generally the message follows the listing of the card in error. In some cases however the error cannot be detected until after a complete section has been read in and therefore the message may appear at the end of that section. The user should carefully check the entire section of the data in which such an error was found. Certain errors which do not affect the structure of the tape can be diagnosed and the checking can be continued. Other "major errors" will cause the code to skip over a bad or unrecognizable part and resume the checking at the beginning of the next section.

B. Description of the Checks and Tests Carried Out by the Code:

The checks and tests carried out by the code CHECKER fall into three major categories: (1) checks of the deck structure, (2) checks of the tabulated functions and (3) specialized checks depending on the type of data. These checks will be discussed below.

---

* NT is defined by the input cards.

1. Checking of the Deck Structure

The deck structure is checked with respect to the following:

(a) The fields MAT, MF, and MT are preset and sections, files and materials are properly ordered.

(b) All cards are sequence numbered.

(c) The control cards SEND, FEND, MEND, and TEND, are included and properly ordered.

(d) The fields ZA and AWR on HEAD cards are filled in and are consistent for a given material.

(e) Fields which count items in a list to follow or determine what type of data is to follow are properly filled in and do not exceed certain bounds.

Detection of errors of Type (e) will, in general, cause the code to skip over the section in question.

2. Checking of Tabulated Functions

Most of the data given on the ENDF/B tape is in the form of a one-dimensional tabulated function (TAB1 record) defined by the sequence of numbers:

NBT(M), JNT(M), M=1, NR

X(N), Y(N), N=1, NP

Each time this record is read, regardless of the physical meaning of the data, the following checks are performed:

(a) $1 \leq NR \leq 100$

(b) $2 \leq NP \leq 3000$*

(c) NBT(M) > NBT(M-1), M=2, NR

(d) NBT(NR) = NP

(e) NBT(1) > 1

(f) $1 \leq JNT(M) \leq 5$, M=1, NR

(g) $X(N) \geq X(N-1)$, N=2, NP

(h) X(N) > 0, when lnX(N) is needed for interpolation

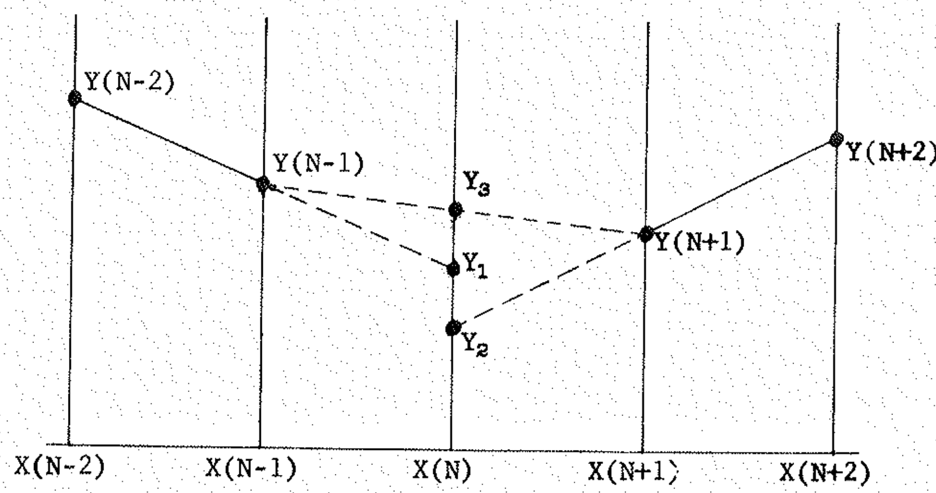
(i) Y(N) > 0, when lnY(N) is needed for interpolation

(j) A crude test to detect mispunched values (deviant point check).

This last check is done in the following manner (which to a certain extent simulates the procedure that the eye uses when checking graphical data):

---

*Although ENDF formats allow arrays of up to 5000 points the present version of CHECKER has been programmed for arrays of dimensions up to 3000.



Shown on the preceding diagram are five sequential points labeled N-2,..., N+2. Point N is the one we wish to check [Y(N) is not shown]. We estimate a value of Y(N) by extrapolation, using the points at N-2 and N-1. This value is called Y_1. Y_2 is obtained by extrapolation, using points at N+1 and N+2. A third point, Y_3, is obtained by interpolating between the points at N-1 and N+1. In this diagram, if $Y_1 < Y(N) < Y_3$ and we are looking at a plot of the function, we would say that Y(N) was a reasonable value and not in error. If Y(N) lay between Y_2 and Y_1, it would also look reasonable, but perhaps not quite as good.

Our purpose here is to detect punching errors which are usually rather gross errors. The philosophy is then to give the points only a crude test and to save the more refined test for later. Accordingly, we define

$$Z_H = 1.10 \text{ Largest of } (Y_1, Y_2, Y_3)$$

$$Z_L = \frac{1}{1.10} \text{ Smallest of } (Y_1, Y_2, Y_3).$$

If the actual value Y(N) lies outside the range (Z_L, Z_H), a message is printed indicating that the point is suspicious. The value of N, X(N), Y(N) are also printed.

This method will obviously not work for the first two and last two points in the table, nor will it work for the two points on the left and two points on the right of a discontinuity. These points are not checked.

3. Specialized Checks Depending on the Type of Data

The following table lists the checks that are made in each of the files of a material, in addition to those described above. The value of n in the column labeled "Test n Subroutine" indicates which of the Test n Subroutines (see p. CH-13) is used to make the test indicated in the column labeled "Type of Test".

(a) Tests for File 1 - General Information - (subroutine CKF1)

Condition	Type of Test	Explanation	Test n Subroutine
	$451 \le MT \le 455$	Acceptable MT numbers.	1
MT = 451 General Information	$0 \le NXC \le 250$	Length of index table.	1
"	$0 \le LRP \le 1$	Flag for resonance parameters.	1
"	$0 \le LFI \le 1$	Flag for fissionable mat'l.	1
"	$0 \le LDD \le 1$	Flag for radioactive decay.	1
"	$0 \le LFP \le 1$	Flag for fission product yield.	1
"	Is index table present?	---	-
	Are file Nos. in table acceptable?	---	1
	Are MT Nos. in table acceptable?	---	1
	Is MF = 1 and MT = 451 in table?	---	10
MT = 452 - # of Neutrons/fission	Was LFI set = 1?	Nuclide fissionable	3
"	Is MF = 1, MT = 452 in table?	---	10
"	$1 \le LNU \le 2$	Permissible representation.	1
LNU = 1	$0.0 \le \nu(E) \le 10.0$	E = 15 MeV.	-
LNU = 2	$0.0 \le \nu(E_i) \le 10.0$	E_i = all energies in table.	-
MT = 453 Decay Data	Was LDD set = 1?	Flag for radioactive decay.	3
"	Is MF = 1, MT = 453 in table?	---	10
"	Is N1 = NRT*6? †	Six quantities per reaction type.	3

(con't on next page)

†Revised formats have been described in ENDF-102, Vol. I. The code will be updated when data with these formats are distributed.

(a) Tests for File 1 (con't)

Condition	Type of Test	Explanation	Test n Subroutine
MT = 454 fission product yield	Was LFP = 1?	Fission product flag.	3
"	Is MF = 1, MT = 454 in index table?	---	10
"	Is N1/2 = NFP?	Two items per fission product.	3
"	Is the sum of all F. P. yields $= 2 \pm 10^{-4}$?	---	-
MT=455, Delayed Neutron data.	Was LFI set = 1?	Nuclide fissionable.	3
	Is MF=1,MT=455 in index tabl.?	---	10
	$1 \le LND \le 2$	Permissible representation of $\bar{\nu}_d(E)$.	1
	Are λ's in decreasing order?	---	5B
LND = 1	$0.0 \le \bar{\nu}_d(E_i) \le 1.0$	E = 15 MeV.	-
LND = 2	$0.0 \le \bar{\nu}_d(E_i) \le 1.0$	E_i = all energies in table.	-

(b) Tests for File 2 - Resonance Parameter Data - (subroutine CKF2)

Condition	Type of Test	Explanation	Test n Subroutine
MT=151, Resonance Data.	MT = 151?	---	3
"	Is MF=2,MT=151 in index tbl.?	---	10
"	$1 \le NIS \le 20$	Number of Isotopes.	1
"	ZAI > 0.0	The (Z,A) designation for isotope.	-
"	$0.0 \le ABN \le 1.0$	Isotopic abundance.	7
"	$0 \le LFW \le 1$	Flag for fission widths.	1
"	$1 \le NER \le 100$	No. of energy ranges.	1
LRU ≠ 0	Was LRP set=1 in File 1?	Flag for resonance parameters.	3
"	Are energy range limits correct?	---	-
"	Are there gaps between E ranges?	---	-
"	$1 \le LRU \le 2$	Flag for resolved or unresolved resonance parameters.	1
LRU = 1	$1 \le LRF \le 4$	Flag for representation type.	1
LRU = 2	$1 \le LRF \le 2$	" " " "	1
LRU=1(Resolved R. P.)	$1 \le NLS \le 100$	No. of ℓ-states.	1
"	Is ℓ = 0, 1, 2,...,etc?	---	3
"	N2*6 = NRS	Six items/resonance.	3
"	Are E_i's increasing in order?	Resonance energies.	5A
"	Do partial widths add up to total?	---	12
"	Is AWRI field set?	Test value calculated from ZAI.	7
LRU=2, LRF=1 (Unresolved R.P.)	$1 \le NLS \le 100$	No. of ℓ-states.	1
"	Is ℓ = 0, 1, 2, ...	---	3
LRU=2,LRF=1, LFW=0	Is N2 = NJS*6?	Six entries for each J-state.	3
"	Is AWRI field set?	Test value calculated from ZAI.	7
LRU=2,LRF=1, LFW=1	$1 \le NJS \le 20$	No. of J-states.	1
LRU=2,LRF=2	$1 \le NLS \le 100$	No. of ℓ-states.	1
"	$1 \le NJS \le 20$	No. o J-states.	1
"	N2 = (N1-6)/6	Six entries at each energy, plus six entries for the degrees of freedom.	3

(con't on next page)

(b) Tests for File 2 Data (con't)

Condition	Type of Test	Explanation	Test n Subroutine
"	Are E_i's increasing?	Resonance energies.	5A
LRU = 1 or 2	Do abundances add up to 1 ± 10^{-3}?	---	-
LRU=0 (No R.P.)	LRF = 0	All materials must have a File 2. If no resonance parameter data given (LRU=0) these tests must be satisfied.	3
"	NER = 0		3
"	NIS = 1		3
"	LRW = 0		3
"	NLS = 0		3

(c) Tests for File 3 - Neutron Cross Section Data - (subroutine CKF3)

any MT	Is MT acceptable?	ENDF/B Version II.	2
"	Is MF=3 and above MT in index table?	---	10

(d) Tests for File 4 - Angular Distributions of Secondary Neutrons - (subroutine CKF4)

Angular distributions (any MT)	Is MT acceptable?	ENDF/B Version II.	2
"	Is MF=4 and above MT in index table?	---	10
"	$0 \leq LVT \leq 1$	Flag for transformation matrix.	1
"	$1 \leq LTT \leq 2$	Flag for representation used.	1
LVT = 0	$1 \leq LCT \leq 2$	Flag for frame of reference.	1
LVT > 0	$1 \leq NM \leq 100$	Max order of Legendre polynomial.	1
"	$1 \leq NE \leq 2000$	No. on incident energy points.	1
LTT=1 (Legendre expansion)	$-1 \leq V_k \leq 1.0$	Transformation matrix elements.	7
	Are angular distributions positive and reasonable?	---	LE6CK
LTT=2 (Tabulated distribution)	$-1.0 \leq \mu_i \leq +1.0$	All μ's in table.	7
"	$\int_{-1}^{+1} p(\mu,E)d\mu = 1 \pm 10^{-4}$	---	-
"	Are E_i's in increasing order?	Energies at which distributions tabulated.	5

(e) Test for File 5 - Energy Distributions of Secondary Neutrons - (subroutine CKF5)

Condition	Type of Test	Explanation	Test n Subroutine
Any MT	Is MT Acceptable?	ENDF/B Version II.	2
"	Is MF=5 and above MT in index table?	---	10
"	$0 \leq LFE \leq 1$	This flag no longer used.	1
"	$1 \leq NK \leq 100$	No. of partial energy distributions.	1
"	$0.0 \leq p_k(E_n) \leq 1.0$	At all E_n.	7
"	$1 \leq LF \leq 10$	Flag for secondary E Distribution law.	1
LF=1 (Arbitrary tabulated function)	$\int g(E \rightarrow E')dE' = 1 \pm 10^{-4}$	---	2
	Are energies in increasing order?	---	5
LF=2,3 (Discrete levels)	$10^3 \leq \theta \leq 2 \times 10^7$	Excitation energy of the level.	7
LF=4 (not used)	$10^4 \leq \theta \leq 10^7$	---	7
LF=5 (Evaporation spectrum)	$10^4 \leq \theta(E) \leq 10^7$	For all E where θ tabulated.	7
"	$\int g(E \rightarrow E')dE' = 1 \pm 10^{-4}$	---	9
LF=6 (not used)	$2 \times 10^5 \leq \theta \leq 5 \times 10^6$	---	7
LF=7 (fission spectrum)	$2 \times 10^5 \leq \theta(E) \leq 5 \times 10^6$	For all E where θ tabulated.	7
LF=8 (not used)	$10^4 \leq \theta \leq 10^7$	---	7
LF=9 (Maxwellian distribution)	$10^4 \leq \theta(E) \leq 10^7$	For all E where θ tabulated.	7

(f) Test for File 6 - Energy - Angular Distributions for Secondary Neutrons - (subroutine CKF6)

Any MT	Is MT acceptable?	ENDF/B Version II	2
"	Is MF=6 and above MT in index table?	---	10
"	$0 \leq LFE \leq 1$	This flag no longer used.	1
"	$1 \leq LTT \leq 2$	Flag for representation type.	1
LTT=1 (Legendre expansion)	$1 \leq LCT \leq 2$	Flag for frame of reference.	1
"	$1 \leq NL \leq 100$	Order of the Legendre expansion.	1
LTT=2 (tabulation)	$1 \leq NA \leq 100$	No. of cos. at which distributions given.	1
	Plus all tests for File 5		

(g) Tests for File 7 - Thermal Neutron Scattering Law Data - (subroutine CKF7)

Condition	Type of Test	Explanation	Test n Subroutine
MT = 4	Is MT = 4?		3
"	MF=7,MT=4 in index table?	---	10
"	$0 \le LAT \le 1$	Flag indicating temperature.	1
"	NS = (NL/6)-1	No. of non-principal scattering atoms.	3
	Are β's in increasing order?		5

(h) Tests for File 14† - Photon Angular Distributions - (subroutine CKF14)

Any MT	Is MT acceptable	ENDF/B Version II	2
"	Is MF=14 and above MT in index table?	---	10
"	$0 \le LI \le 1$	Flag for isotropic distribution.	1
LI = 0	$1 \le ND \le 100$	If not isotropic-number of energies.	1
"	$1 \le LTT \le 2$	Flag for representation used.	1
"	$0 \le NC$	No. of continuous spectra.	1
"	$1 \le NE \le 2000$	No. of neutron energies given.	1
LTT = 1 (Lengendre expansion)	$-1.0 \le f_\ell(E) \le +1.0$	---	7
"	Are energies in increasing order?	Neutron energies.	5
"	Angular dist. everywhere positive?	---	LEGCK
LTT = 2 (tabulation)	Are energies in increasing order?	Neutron energies.	5
	$-1.0 \le \mu_i \le +1.0$	---	
	$\int_{-1}^{+1} p(\mu_i E_i)d\mu = 1 \pm 10^{-4}$		

†A new version of CHECKER with tests for the revised formats of Files 12, 13, 14, 15, etc. is being written.

(i) Tests for File 15 - Photon Production Multiplicities - (subroutine CKF15)

Condition	Type of Test	Explanation	Test n Subroutine
Any MT	Is MT acceptable?	ENDF/B Version II	2
"	Is MF=15 and above MT in index table?	---	10
"	$1 \leq LO \leq 2$	Option parameter.	1
LO=1 (Option 1)	$1 \leq NK \leq 100$	No. of partial energy dist.	1
"	$1 \leq LF \leq 2$	Flag for final energy dist. law.	1
LF=1 (tabulated fn.)	Are energies in increasing order?	---	5
"	$\int g(E_\gamma \leftarrow E_i)dE_\gamma = 1 \pm 10^{-4}$	All E_i	-
LF=2 (discrete final energy)	$1 \leq LG \leq 2$	Flag for simple or complex process.	1
LO=2 (Option 2) LG=1	N1 = N2*2	Two items per state.	3
"	Are ES_i in decreasing order?	Energy of i^{th} state.	5B
LG = 2	N1 = N2*3	Three items per state.	3
"	Are ES_i in increasing order?	Energy of the i^{th} state.	5A

(j) Tests for File 16 - Photon Energy - Angle Distributions (subroutine CK16)

Any MT	Is MT acceptable?	ENDF/B Version II	2
"	Is MF=16 and above MT in index table?	---	10
	$1 \leq LTT \leq 2$	Flag for representation used.	1
	All other tests same as for File 15.	---	-

(k) Tests for File 23 - Smooth Photon Cross Sections - (subroutine CKF23)

Any MT	Is MT acceptable?	See ENDF - 111	2B
"	Is MF=23 and above MT in index table?	---	-

(ℓ) Tests for File 24 - Secondary Angular Distributions for Photons - (subroutine CKD24)

Condition	Type of Test	Explanation	Test n Subroutine
Any MT	Is MT acceptable?	ENDF-111	2B
"	Is MF=24 and above MT in index table?	---	10
"	LVT = 0	No transformation matrix for photons.	-
"	$1 \le LTT \le 2$	Flag for representation used.	1
"	LCT = 1	Data must be in (L) system.	-
"	$1 \le NE \le 2000$	No. of incident energy points.	1
LTT = 1 (Legendre expansion)	$-1.0 \le B_k \le +1.0$	Expansion coefficients.	7
"	Are energies in increasing order?	---	5
"	Are angular distributions positive and reasonable?	---	LEGCK
LTT = 2 (tabulated distribution)	$-1.0 \le \mu_i \le +1.0$	All μ_i's in table.	7
"	$\int_{-1}^{+1} p(E,\mu)d\mu = 1 \pm 10^{-4}$	Normalization	9
"	Are energies in increasing order?	---	5

(m) No tests for File 25 - Secondary Photon Energy Distributions -

(n) Tests for File 26 - Secondary Photon Energy - Angular Distributions - (subroutine CKF26)
Same as tests for File 6.

(o) Tests for File 27 - Form Factors for Coherent and Incoherent Scattering - (subroutine CKF27)
Same as tests for File 7.

4. TEST n Subroutines

Error messages are, for the most part, self-explanatory. Most of the error messages come as a result of tests failed in a set of subroutines called TESTn, where n = 1, 12. The function of each of these subroutines and their error messages are given in this Section. The symbol KXXX is the location of a three-character word, XXX, which indentifies the quantity being tested. The notation (NA) should be read as "the value of NA."

TEST1 (N,NA,NB,KXXX,IERR)

If N < NA or N > NB, print:

"XXX ØUT ØF RANGE (NA) - (NB)"

If IERR=1, return to calling program.

If IERR=2, execution in terminated.

TEST2 (MT)

If MT is not in the range 1-4, 16-26, 28-29, 51-91, 101-109, 151, 251-253, 301-455, 700-799, print:

"MT = (MT) INCØRRECT"

TEST2B (MT)

If MT is not in the range 501-504, 516, 518, 531-533, 602, print:

"MT = (MT) INCØRRECT"

TEST3 (N1,N2,KXXX)

If N1 is not equal to N2, print:

"XXX SHØULD BE SET TØ (N2)"

TEST4 (NBT,JNT,NR,NP)

If any of the following conditions are not met,

NBT(M) > NBT(M-1), M=2, NR

NBT(NR) = NP

NBT(1) > 1

$1 \le$ JNT(M) ≤ 5, M=1, NR   print:

"INTERPØLATIØN TABLE INCØRRECT"

TEST5 (X,NP)

If the array X(N), N=1, NP is not in increasing order at the point K, print:

"X LIST ØUT ØF ØRDER NEAR N=(K)"

TEST5A (X,NP,L)

If every "L" th entry in array X(N), N=1, NP is not in increasing order at the point K print:

"X LIST ØUT ØF ØRDER NEAR N=(K)"

TEST5B (X,NP,L)

Same as TEST5A except that array is tested for decreasing order.

TEST6

If the label fields are incorrect, or a card is out of sequence, print:

"MAT INCØRRECT" or
"MT INCØRRECT" or
"MF INCØRRECT" or
"ØUT ØF SEQUENCE"

TEST7 (Y,NP,A,B,KXXX)

If any of the elements of the array Y(N), N=1, NP lie outside the range A - B, print:

"XXX NØT IN RANGE (A) - (B)"

TEST8 (X,Y,NP,NBT,JNT,NR)

This subroutine provides an overall check on the TAB1 record. If lnX(N) or lnY(N), N=1, NP, is needed for interpolation and the value is ≤ 0 at point K, print:

"NEG ØR ZERØ ARG ØF LØG BELØW PØINT N =(K)
DEVIANT PØINT CHECK ØMITTED"

If X(K) < X(K-1), print:

"LIST ØUT ØF ØRDER NEAR N = (K)
DEVIANT PØINT CHECK ØMITTED"

If the deviant point check described above on p. 3 fails, print:

"CHECK PØINT (N), X = (X), Y = (Y)"

If more than 25 deviant points are found by this method, only the first 25 are printed along with the message:

"AND MAYBE ØTHERS"

TEST9 (X,Y,NP,NBT,JNT,NR)

If the integral of the function Y(X) differs from unity by more than $\pm 10^{-4}$, print:

"NØRMALIZATIØN CHECK, INTEGRAL = (value)"

TEST10 (MF,MT)

If MF or MT are not in INDEX, print:

"SECTION (MF),(MT) NOT IN INDEX"

TEST11 (IREC)

If the data in record type IREC (LIST,TAB1 or TAB2) is $0 < |X_k| < 10^{-20}$ print:

"DATA POINT (K), VALUE (X_K) IS BELOW THE ACCEPTABLE LOWER LIMIT OF ABS (1.0E-20) AND GREATER THAN ZERO"

TEST12 (BX,NPTS,NTOT,NSTEP)

Used for testing if partial resonance widths add up to total. If (TOT), the sum of the three entries in array BX following BX(NTOT) do not add up to BX(NTOT) print:

"SUM OF PARTIALS DOES NOT ADD UP TO TOTAL AT THE FOLLOWING POINTS=ENERGY=(BX(NTOT-1)), GAMMA-TOTAL=(BX(NTOT)), SUM=(TOT)"

Repeat test after NSTEP entries up to BX(NPTS).

III. Limitations and Assumptions

The program CHECKER assumes its input tape to be in ENDF/B Version 2 format. The code will check the general structure and formats of the tape as well as the consistency of the data. No major physics checks are done.

IV. Requirements

The program was written for the CDC-6600 in FORTRAN-IV. It has been converted to the PDP-10 as well as to various IBM computers. It requires less than $54K_8$ (or $23K_{10}$) of core storage and one tape unit for the ENDF/B input tape. In addition input (e.g., card reader) and output (e.g., line printer) units are required.

V. Input

The input data consists of the BCD tape to be checked (symbolic name NT) and two cards read from the system input tape (NIN). The ENDF/B tape to be checked should contain a label card (TPID) as the first card and a tape end card (TEND) as the last card.

The data given on the two cards read from NIN is described in the following table:

Card	Cols.	Format	Symbol	Description
1	1-11	I11	NT	Logical tape assignment of the ENDF/B tape.
	12-22	I11	LABEL	The number given in the MAT field of the TPID card. This will be checked to insure that the proper tape is mounted. If LABEL = 0, checking is ignored.
	23-33	I11	NØPT	If NØPT=0 complete listing of data and errors are generated; if NØPT=1 only errors are listed.
	34-44	I11	IRDHOL	Hollerith read option: If IRDHOL = 0 ignore If IRDHOL = 1 read records in 80 A1 format and try to reconstruct the appropriate format.
2	1-11 12-22 23-33	I11 I11 I11	MAT1 MF1 MT1	The material, file, and section number of the first section to be checked. If MAT1=0, checking starts at the beginning of the tape.
	34-44 45-55 56-66	I11 I11 I11	MAT2 MF2 MT2	The material, file, and section number of the last section to be checked. If MAT2 = 0, checking continues until the end of the tape.

VI. Output

The output consists of either the listing of the ENDF/B tape with diagnosed errors followed by error messages or only of the error messages with the incorrect records referenced by their sequence number, if the short listing option is selected.

Program PLOTFB

by

D. E. Cullen
National Neutron Cross Section Center
Brookhaven National Laboratory
Upton, New York 11973

September 28, 1970

Contents:

Program PLOTFB

by

D. E. Cullen
National Neutron Cross Section Center
Brookhaven National Laboratory
Upton, New York 11973

September 28, 1970

I. Purpose

The purpose of Program PLOTFB is to produce interpreted listings and/or CALCOMP plots of ENDF/B data from BCD card image or binary tapes in the standard arrangement.

II. Method

Plotting and/or listing is performed at the file (MAT,MF) level. Plotting and/or listing is non-selective within a file (all "tables" are plotted). ENDF/B data which may be considered to be a table of X vs. Y is automatically scaled and plotted. All numerical fields are translated to indicate their physical significance.

III. Limitations and Assumptions

The program does not check the ENDF/B format and as such this program will only work on properly structured ENDF/B tapes. The CALCOMP subroutines used in this program are those supplied by CALCOMP to CALCOMP-835 users and locally modified at Brookhaven (for further information contact the Applied Mathematics Department, BNL).

IV. Requirements

The program was written for the CDC-6600 in FORTRAN-IV and it has been converted to other computers. The program requires about $74K_8$ (approximately $31K_{10}$) of core and either two tape units (ENDF/B tapes and plotting tape) or a single tape unit and an on line plotter. In addition INPUT (e.g., card reader) and OUTPUT (e.g., line printer) units are required.

V. Input

The input consists of a single card defining the run parameters followed by up to fifty (50) request cards. The stack of request cards should be followed by a card that is blank in Columns 1-20. If the first request card is blank in Columns 1-20 it is interpreted as applying to all materials uniformly (i.e., is equivalent to requesting all MATs). Each request card defines a MAT and/or ZA (10000*Z+A) and a list of files to plot (1-7, 14-16, 23-27).

Input Card 1

Columns	
1 - 5	ENDF/B data tape unit number (e.g., 10)
6 - 10	Plotting output tape unit number (e.g., 11)
11 - 15	ENDF/B tape MODE = 1 BCD; = 2 Binary
16 - 20	ENDF/B tape number (e.g., 114) = greater than zero - read TPID and check label = equal to zero - read TPID. Do not check label = less than zero - do not read TPID or check label
21 - 25	Listing option = 1 - Produce interpreted listing; = 0 - no interpreted listing
26 - 30	Listing editing option = 1 - one section per page; = 0 - many sections per page
31 - 35	Plotting option = 1 - plot output; = 0 - no ploter output
36 - 40	Grid option = 1 - draw fine structure grid lines; = 0 - draw coarse grid lines only (faster)
41 - 45	= beam/pen compatibility option = 1 generate beam/pen compatible tapes = 0 generate beam compatible tapes (faster)
46 - 50	Starting frame number for output (e.g., 1) = greater than zero - number all frames starting at this number = less than or equal to zero - do not number frames

Input card 2 to N (N = 2 to 51)

Columns		
1 - 5	ENDF/B material number (MAT)	either or both may be used to identify a material.
6 - 15	ENDF/B ZA (i.e., 10000*Z+A)	
21 - 23	List of files (MF) to be plotted.	
24 - 26	List is terminated by a blank.	
27 - 29	All illegal numbers are automatically ignored. Legal MF	
.	ranges are 1-7, 14-16, and 23-27.	
.		
.		
78 - 80		

example

To plot Files 3 and 4 of MAT = 1063 and Files 1 and 3 of MAT ≑ 1059 with grid lines the following input is required.

Col.	5	10	15	20	23	26	30	35	40	45	50
	10	11	1	114				1	1	1	1
	1059				3	4					
	1063				1	3					

---Blank Card--------------------

VI. Output

The output naturally consists of CALCOMP plots (either directly or via tape) and/or interpreted listings. In addition the listed output includes an edited summary of the input cards in the form:

THE FOLLOWING CONDITIONS HAVE BEEN SELECTED FOR PLOTFB RUN NUMBER 1

LOGICAL NUMBER OF INPUT DATA UNIT----------------------------------	10
LOGICAL NUMBER OF PLOTTING OUTPUT UNITS------------------------	11
MODE OF INPUT DATA TAPE---	BCD
REQUESTED TAPE LABEL--	114
SHOULD DATA BE LISTED---	NO
SHOULD DATA BE EDITED TO ONE SECTION PER PAGE------------------	NO
SHOULD DATA BE PLOTTED---	YES
SHOULD OUTPUT CONTAIN GRID LINES*-------------------------------	YES
WILL OUTPUT BE BEAM/PEN COMPATIBLE------------------------------	YES

THE FOLLOWING DATA HAS BEEN REQUESTED:

MAT	ZA	Files	
1059	0.0	3	4
1063	0.0	1	3

---

* Actually fine structure.

Program LISTFC

by

D. E. Cullen
National Neutron Cross Section Center
Brookhaven National Laboratory
Upton, New York 11973

March 23, 1970

Table of Contents

Program LISTFC

by

D. E. Cullen
National Neutron Cross Section Center
Brookhaven National Laboratory
Upton, New York 11973

I. Purpose

The purpose of program LISTFC is to produce interpreted listings of ENDF/B data. This program is an abrieviated version of PLOTFB and will only produce interpreted output from BCD standard arrangement ENDF/B tapes. The plot option from PLOTFB is not available in LISTFC.

II. Method

Listing is performed at the file (MAT,MF) level. Listing is non-selective within a file (all "tables" are plotted). ENDF/B data which may be considered to be a table of X vs. Y is interpreted and listed. All numerical fields are translated to indicate their physical significance.

III. Limitations and Assumptions

The program does not check the ENDF/B format and as such this program will only work on properly structured ENDF/B tapes. CALCOMP plots (ala PLOTFB) are not available from LISTFC.

IV. Requirements

The program was written for the PDP-10 in FORTRAN-IV but should be easily converted to other computers. The program requires less than $60K_8$ (approximately 21 K_{10}) of core and one tape unit (ENDF/B tapes). In addition INPUT (e.g. card reader) and OUTPUT (e.g. line printer) units are required.

V. Input

The input consists of a single card defining the run parameters followed by up to fifty (50) request cards. The stack of request cards should be followed by a card that is blank in columns 1-20. If the first request card is blank in columns 1-20 it is interpreted as applying to all materials uniformly (i.e. is equivalent to requesting all MATs). Each request card defines a MAT and/or ZA (1000*Z+A) and a list of files to edit (1-7, 14-16, 23-27). Several of the fields on the first input card have been maintained for compatibility with PLOTFB even those that are ignored or restricted as indicated below.

Input card 1

Columns	
1 - 5	ENDF/B data tape unit number (e.g. 10)
6 - 10	Plotting output tape unit number (ignored)

(con't on next page)

Input card 1 (con't)

Columns	
11 - 15	ENDF/B tape MODE (must be 1) = 1 BCD
16 -20	ENDF/B tape number e.g. 114) = greater than zero - read TPID and check label = equal to zero - read TPID. Do not check label = less than zero - do not read TPID or check label (e.g. unlabelled tape)
21 - 25	= listing option (must be 1)
26 - 30	= listing editing option = 1 edit output to one ENDF/B section per page = 0 minimize output by multiple sections per page
31 - 35	= plotting option (ignored)
36 - 40	= ticks structure option (ignored)
41 - 45	= beam/pen compatibility option (ignored)
46 - 50	starting frame number for output (ignored)
50 - 55	size of plotting frame in X direction (ignored)
56 - 60	" " " " " Y " (")

Input card 2 to N (N = 2 to 51)

Columns		
1 - 5	ENDF/B material number (MAT)	either or both may be used to identify a material
6 - 15	ENDF/B ZA (i.e. 1000*Z+A)	
21 - 23 24 - 26 27 - 29 . . . 78 - 80	List of files (MF) to be listed. List is terminated by a blank. All illegal numbers are automatically ignored. Legal MF ranges are 1-7, 14-16, and 23,27.	

example

```
Col.    5|   10|   15|    20|  25|   30|  35|  40|  45|  50|  55|  60|
       10|   11|    1|   114|    1|    1|
     1059|               3|    4|
     1063|               1|    3|
---------|-----|-----|------|---|---|----Blank Card-----|----|----|----
Col.                       |23    |26
```

To LIST Files 3 and 4 of MAT = 1063 and Files 1 and 3 of MAT = 1059 with output edited to one section per page the input shown above is required.

VI. Output

The output naturally consists of interpreted listings (see: Figure 1). The only additional listed output is an edited summary of the input cards in the form:

```
THE FOLLOWING CONDITIONS HAVE BEEN SELECTED FOR PLOTFB† RUN NUMBER 1

LOGICAL NUMBER OF INPUT DATA UNIT ---------------------------------- 10
LOGICAL NUMBER OF PLOTTING OUTPUT UNITS ---------------------------- 11
MODE OF INPUT DATA TAPE ------------------------------------------- BCD
REQUESTED TAPE LABEL ---------------------------------------------- 114
SHOULD DATA BE LISTED --------------------------------------------- YES
SHOULD DATA BE EDITED TO ONE SECTION PER PAGE ---------------------- YES
SHOULD DATA BE PLOTTED --------------------------------------------- NO
SHOULD OUTPUT CONTAIN GRID†† LINES --------------------------------- NO
WILL OUTPUT BE BEAM/PEN COMPATIBLE --------------------------------- NO

THE FOLLOWING DATA HAS BEEN REQUESTED:

MAT     ZA      FILES
1059    0.0     3  4
1063    0.0     1  3
```

† Note: Title not changed to LISTFC

†† Actually fine structure

NEPTUNIUM-237 | ELASTIC NEUTRON CROSS SECTIONS | ENDF/B MATERIAL NO. 1048

INTERPOLATION LAW BETWEEN ENERGIES

RANGE	DESCRIPTION
1 TO 5	Y LINEAR IN X
5 TO 47	LN Y LINEAR IN LN X

NEUTRON CROSS SECTIONS

INDEX	ENERGY	CROSS SECTION	ENERGY	CROSS SECTION	ENERGY	CROSS SECTION	ENERGY	CROSS SECTION	ENERGY	CROSS SECTION
1	1.0000E-04	1.2000E+01	1.0000E-01	1.2000E+01	1.0000E-01	0.	1.2000E+03	0.	1.2000E+03	1.2000E+01
6	1.5000E+03	1.0728E+01	2.0000E+03	9.8400E+00	3.0000E+03	9.5020E+00	5.0000E+03	1.0165E+01	7.0000E+03	1.0271E+01
11	1.0000E+04	1.0678E+01	2.0000E+04	1.0887E+01	2.4000E+04	1.1500E+01	4.0000E+04	1.0800E+01	6.0000E+04	1.0350E+01
16	8.0000E+04	9.9000E+00	1.0000E+05	9.6000E+00	1.5000E+05	8.8000E+00	2.0000E+05	8.2000E+00	2.5000E+05	7.8000E+00
21	3.0000E+05	7.3500E+00	3.5000E+05	7.0000E+00	4.0000E+05	6.6500E+00	4.5000E+05	6.4000E+00	5.0000E+05	6.1000E+00
26	5.5000E+05	5.9000E+00	6.0000E+05	5.7000E+00	7.0000E+05	5.3000E+00	8.0000E+05	5.0000E+00	9.0000E+05	4.7000E+00
31	1.0000E+06	4.4000E+00	1.2000E+06	4.0000E+00	1.5000E+06	3.8000E+00	2.0000E+06	3.8000E+00	2.5000E+06	4.4000E+00
36	3.0000E+06	4.6000E+00	3.5000E+06	4.5000E+00	4.0000E+06	4.4000E+00	5.0000E+06	4.4000E+00	6.0000E+06	4.4000E+00
41	6.5000E+06	4.3000E+00	7.0000E+06	4.1000E+00	8.0000E+06	3.6000E+00	9.0000E+06	3.4000E+00	1.0000E+07	3.2000E+00
46	1.2000E+07	3.1000E+00	1.5000E+07	3.0500E+00						

NEPTUNIUM-237 | TOTAL NEUTRON CROSS SECTIONS | ENDF/B MATERIAL NO. 1048

INTERPOLATION LAW BETWEEN ENERGIES

RANGE	DESCRIPTION
1 TO 53	LN Y LINEAR IN LN X

NEUTRON CROSS SECTIONS

INDEX	ENERGY	CROSS SECTION	ENERGY	CROSS SECTION	ENERGY	CROSS SECTION	ENERGY	CROSS SECTION	ENERGY	CROSS SECTION
1	1.0000E-04	2.7020E+03	1.0000E-03	8.6300E+02	1.0000E-02	2.8100E+02	2.5300E-02	1.8100E+02	4.0000E-02	1.4640E+02
6	6.0000E-02	1.2180E+02	9.0000E-02	1.0170E+02	1.0000E-01	9.7100E+01	1.0000E-01	7.4500E+01	1.2000E+03	6.8000E-01
11	1.2000E+03	2.2860E+01	1.5000E+03	2.0000E+01	2.0000E+03	1.8000E+01	3.0000E+03	1.6000E+01	5.0000E+03	1.5000E+01
16	7.0000E+03	1.4300E+01	1.0000E+04	1.4000E+01	2.0000E+04	1.3500E+01	2.4000E+04	1.4012E+01	4.0000E+04	1.3381E+01
21	6.0000E+04	1.2805E+01	8.0000E+04	1.2568E+01	1.0000E+05	1.2201E+01	1.5000E+05	1.1359E+01	2.0000E+05	1.0658E+01
26	2.5000E+05	1.0194E+01	3.0000E+05	9.7360E+00	3.5000E+05	9.4300E+00	4.0000E+05	9.2100E+00	4.5000E+05	9.1300E+00
31	5.0000E+05	9.0950E+00	5.5000E+05	9.0730E+00	6.0000E+05	9.1210E+00	7.0000E+05	8.9320E+00	8.0000E+05	8.8700E+00
36	9.0000E+05	8.8050E+00	1.0000E+06	8.6600E+00	1.2000E+06	8.3900E+00	1.5000E+06	8.3640E+00	2.0000E+06	8.4640E+00
41	2.5000E+06	8.9700E+00	3.0000E+06	8.9590E+00	3.5000E+06	8.6900E+00	4.0000E+06	8.5270E+00	5.0000E+06	8.5450E+00
46	6.0000E+06	8.6900E+00	6.5000E+06	8.7450E+00	7.0000E+06	8.7301E+00	8.0000E+06	8.4550E+00	9.0000E+06	8.3380E+00
51	1.0000E+07	8.1830E+00	1.2000E+07	7.9028E+00	1.5000E+07	7.4462E+00				

Figure 1: Sample Listing From LISTFC

Program DICTION

by

D. E. Cullen
National Neutron Cross Section Center
Brookhaven National Laboratory
Upton, New York 11973

April 20, 1970

Contents

Program DICTION

by

D. E. Cullen
National Neutron Cross Section Center
Brookhaven National Laboratory
Upton, New York 11973

April 20, 1970

I. Purpose

The purpose of Program DICTION is to construct a new section dictionary (File 1, Section 451) for an entire ENDF/B tape. If a section dictionary is already present it is replaced.

II. Method

During the first pass, the ORIGINAL ENDF/B tape is read and an entry consisting of

(1) material identification (MAT)
(2) file identification (MF)
(3) reaction identification (MT)
(4) number of cards in the section (excluding the SEND card)

is created for each section of the tape.

During the second pass, the ORIGINAL ENDF/B tape is copied to the FINAL ENDF/B tape, the dictionary entries are inserted (the original dictionary, if any, is deleted), and the cards are re-numbered sequentially.

III. Limitations and Assumptions

The program does not check the ENDF/B format; it merely creates a dictionary entry for every (MAT,MF,MT) couple found on the tape. Therefore, tapes in the incorrect format may generate nonsense dictionary entries. The ENDF/B tape must contain less than 1000 sections. If it contains more than 1000 sections an error message is printed and the program will stop.

The ORIGINAL and FINAL tapes must properly execute REWIND instructions. In addition the FINAL tape must properly execute an ENDFILE command.

The program will only work on BCD standard arrangement ENDF/B tapes.

IV. Requirements

The program is written for the CDC-6600 and PDP-10 and requires 12^{*}_{10}($14K_8$) of core and two tape units (ORIGINAL and FINAL).

---

*PDP-10 core requirement

V. Input

The input to the program consists of a single card defining the ORIGINAL and FINAL tape unit numbers. All fields are in the ENDF/B convention of eleven (11) columns each.

Columns	
1 - 11	ORIGINAL tape unit number (e.g. 10)
12 - 22	FINAL tape unit number (e.g. 11)

Example

To read an ENDF/B tape from unit 10 and write an ENDF/B tape with correct section dictionary on unit 11 the following input card should be used.

Column:	11	22
	10	11

VI. Output

Output consists of an interpretation of the input card in the following form:

DICTION RUN PARAMETERS

ORIGINAL TAPE UNIT ------- 10
FINAL TAPE UNIT ---------- 11

In addition a short listing describing the dictionary for each material is also included in the otuput.

Example

------------(MAT = 1005)------------

MF	MT	CARDS
1	451	93
3	1	43
3	2	43
3	3	43
.	.	.
.	.	.
.	.	.

Retrieval Subroutines for the ENDF/B System (SLAVE-3)

by

Henry C. Honeck
U.S. Atomic Energy Commission
Washington, D. C.

March 1967
Revised April 1969

Contents

Contents (con't)

1. Introduction

This document is a description of many subroutines for retrieving and processing data from the ENDF/B system. When a programmer writes a code to retrieve and process ENDF/B data for his specific application, there will be many operations he must perform which are common to any ENDF/B retrieval code. The intent is to collect and document here those subroutines which are common to all retrieval codes so that the preparation of retrieval codes will be much easier, and the resulting codes will be more understandable to other users of the code.

To illustrate the utility of the retrieval routines, the routines are included in a listing of the SLAVE-3 program. This program determines spectrum averages for File 3 (pointwise data) and in the process interpolates, multiplies, and averages data to a predetermined accuracy.

This document is intended to be a "programmer's manual" and is written from the point of view that the ENDF/B is a collection of data in a certain format which is to be manipulated. That these data have certain physical meanings is incidental here, and the physical meanings are not discussed. It is assumed that ASA Standard Fortran (Fortran IV) is the only language used, and that the subroutines are to work on any computer of reasonable size.

A certain philosophy has been used in preparing the subroutines described here. In brief, this philosophy is summed up by the phrases "get the job done" and "be flexible." The basis for this philosophy comes from a supposition on how retrieval codes will be written. The first round of retrieval codes will be experimental in nature to gain experience in using the ENDF/B data. Speed and optimization of the code are not the main concerns at this stage, and it is for this first round of retrieval codes that this document is intended. A second round of codes will then follow which take advantage of the peculiarities of a particular computer and will be faster and more economical to use. These two classes of codes might be described by the words "experimental" and "production."

The routines described here are only a few of the many which will be developed. This document contains those routines which have been checked out and which have been available for use since March 1967. It is assumed that the subroutines and the document are <u>not</u> satisfactory as they stand. The only way this situation can be changed is if each member of the CSEWG prepares comments on why they are unsatisfactory for his purpose, and presents these comments to the Codes and Formats Subcommittee of the CSEWG.

The remainder of this document is divided into the following sections:

<u>Section 2</u> - A review of the ENDF/B specifications is given with emphasis on the logical structure of the file.

<u>Section 3</u> - The allocation of storage is discussed and the labeled common blocks /RECS/ and /DENS/ defined.

<u>Section 4</u> - The error stops are described. (Subroutine ERRØR)

<u>Section 5</u> - Interpolation methods are described. (Subroutines TERP1, TERP2)

<u>Section 6</u> - The subroutines for transmitting data between the various storage areas are described. (Subroutines RREC, WREC, STØRE, FETCH, DELETE, RBS, WBS, CRØP, LRIDS, FPDS, IPDS)

<u>Section 7</u> - The representation and generation of TAB1 functions from analytic functions is described. (Subroutine GENT1)

<u>Section 8</u> - The integration of TAB1 functions is described. (Subroutines ECSI, GRATE)

<u>Section 9</u> - The methods for combining (adding, subtracting, multiplying, dividing, and interpolating) two TAB1 functions are described. (Subroutines CØMBP, CØMB)

2. Review of the ENDF/B Structure

A brief review of the ENDF/B structure is given in this section. The emphasis is on the logical and mathematical nature of the file.

2.1 Definitions and Conventions

The notation used throughout is that of the Fortran IV programming language. The following additional rules have been used:

1. Symbols starting with I, J, K, L, M, or N are integers. All other symbols are floating point numbers or Hollerith information.

2. Letters J, K, L, M, or N, when used alone, are indices.

3. A symbol starting with M is a control number. Examples; MAT, MF, MT.

4. A symbol starting with L is a test number. Examples: L1, L2.

5. A symbol starting with N is a count of items. Examples: N1, N2.

6. Brackets, [ ], denote a record.

7. Brackets, { }, denote a group of records.

A record is a group of numbers. The symbols and definitions of the numbers used in a record are:

MAT - Material number (integer from 1 to 9999).

MF - File number (integer from 1 to 99).

MT - Reaction type number (integer from 1 to 999).

C1 - A constant (floating point). In most cases, this constant will be the temperature (^{o}K).

C2 - A constant (floating point).

L1 - A test (integer). In most cases, this test will be used to indicate whether temperature dependence is given, and the temperature interpolation code.

L2 - A test (integer).

N1 - A count of items in a list to follow (integer).

N2 - A count of items in a second list to follow (integer).

X(N) - A table of x values of a tabulated function (floating point). N=1, N2.

Y(N) - A table of y values of a tabulated function (floating point). N=1, N2.

B(N) - A list of numbers (floating point). N=1, N1.

H(N) - A list of Hollerith information, 4 characters/word (Hollerith). N=1, N1.

NBT(N) - A break point table (integer). N=1, N1.

JNT(N) - A table of interpolation codes (integer). N=1, N1.

The first 9 numbers (MAT-N2) always appear in a record. The remaining numbers may be used in the particular record types discussed later. The definitions given above are general definitions to indicate the position of a number in the record. When a record is processed, a physical meaning and new symbol will be associated with each particular number. For example, the X values may be identified as energies, and the Y values as cross sections or probabilities.

Two changes from the original ENDF/B specifications should be noted. First, the array INT(N) has been changed to JNT(N) because INT specifies an internal function on some computers. Second, Hollerith information is limited to 4 characters/word since this is the least common denomination for all computers. Since there is little Hollerith information in the file, this should not be a severe restriction.

2.2 Arrangement of the Data

The smallest unit of data considered is a record, which is a group of numbers. A record would be a deck of cards or a binary record on tape as read or written with a single Fortran statement. A section is a group of records which give all of the data for a particular reaction type, a particular class of data, and a particular material. For example, a section might contain the fission cross section for U-235, the elastic angular distribution for iron, or the secondary energy distribution for inelastic scatterring in lead.

There are two possible ways to arrange sections in the ENDF/B system. In the standard arrangement, a group of sections containing the same class of data for one material is called a file. For example, all cross sections for iron, all angular distributions for aluminum, or all secondary energy distributions for U-235 are each contained in a file. A material is a collection of all files for that material. In the alternate arrangement, the file and material are interchanged. A material is a group of sections containing the same class of data for one material, and a file is a group of materials. These arrangements will be illustrated in the next section.

The data may be stored on cards (assumed to be on magnetic tape as BCD card images) or on a binary tape. The various combinations of arrangement and representation on tape are specified by a tape mode. Four modes are used and specified by MØDE.

MØDE = 1, Binary tape, standard arrangement
= 2, Binary tape, alternate arrangement
= 3, BCD card image tape, standard arrangement
= 4, Expanded and interpreted tape for printing, standard arrangement.

MØDE = 1, 2, 3 may be used as either input or output.

MØDE = 4 is only used to provide an interpreted output tape for printing.

2.2.1 Standard Arrangement (Mode 1)

The structure of an ENDF/B tape in the standard arrangement (Mode 1) is shown schematically in Figure 2.1. The tape contains a single record at the beginning (TPID) which identifies the tape, and a record at the end (TEND) signalling the end of the tape. The remainder of the tape is first divided into materials. The data for a material is divided into files, each containing certain classes of data. A file is divided into sections, each containing data for a particular reaction type. Finally, each section is divided into records, each of which correspond to a logical binary record on tape.

Associated with each of these major subdivisions is a number. MAT is the material number, MF is the file number, and MT is the reaction type number. Every record on the tape contains these three numbers as the first three numbers of the record. The numbers are always in increasing order, and the hierarchy is MAT, MF, and MT.

There is no count of records in a section, sections in a file, or files in a material. Sections and files (except for file 1, first section) which are not used are omitted from the tape. The end of a section is signalled by a SEND record, the end of a file by a FEND record, and the end of a material by a MEND record.

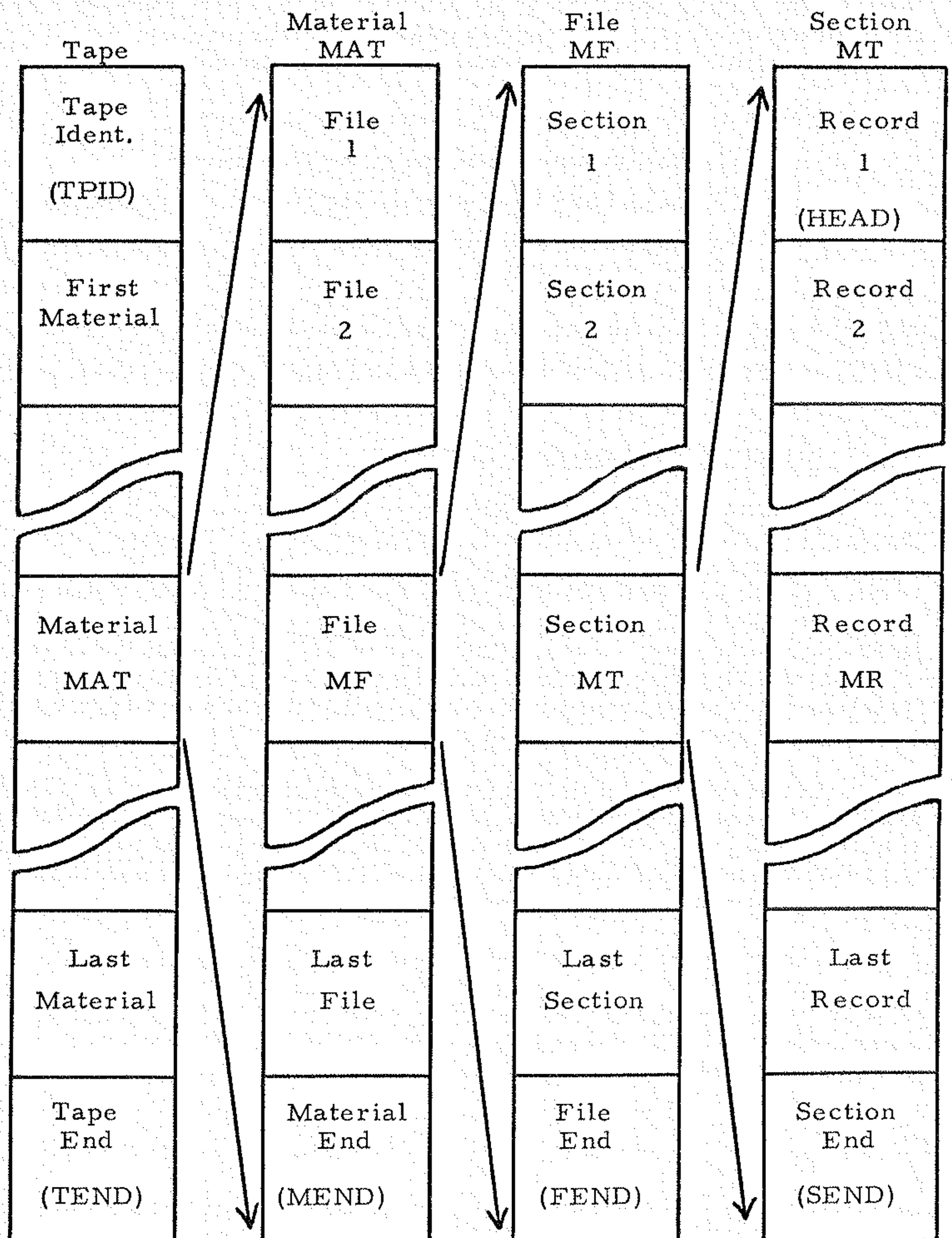


Figure 2.1 Standard Arrangement of an ENDF/B Tape

2.2.2 Alternate Arrangement (Mode 2)

The structure given in the preceding section is well suited for card decks and binary tapes for many processing programs. It is desirable to indicate an alternate arrangement for a binary tape more suited to segmented types of processing programs. This alternate arrangement is illustrated in Figure 2.2 and is simply an interchange of materials and files. The hierarchy is now MF, MAT, and MT, and the first three numbers in each record should conform to this order.

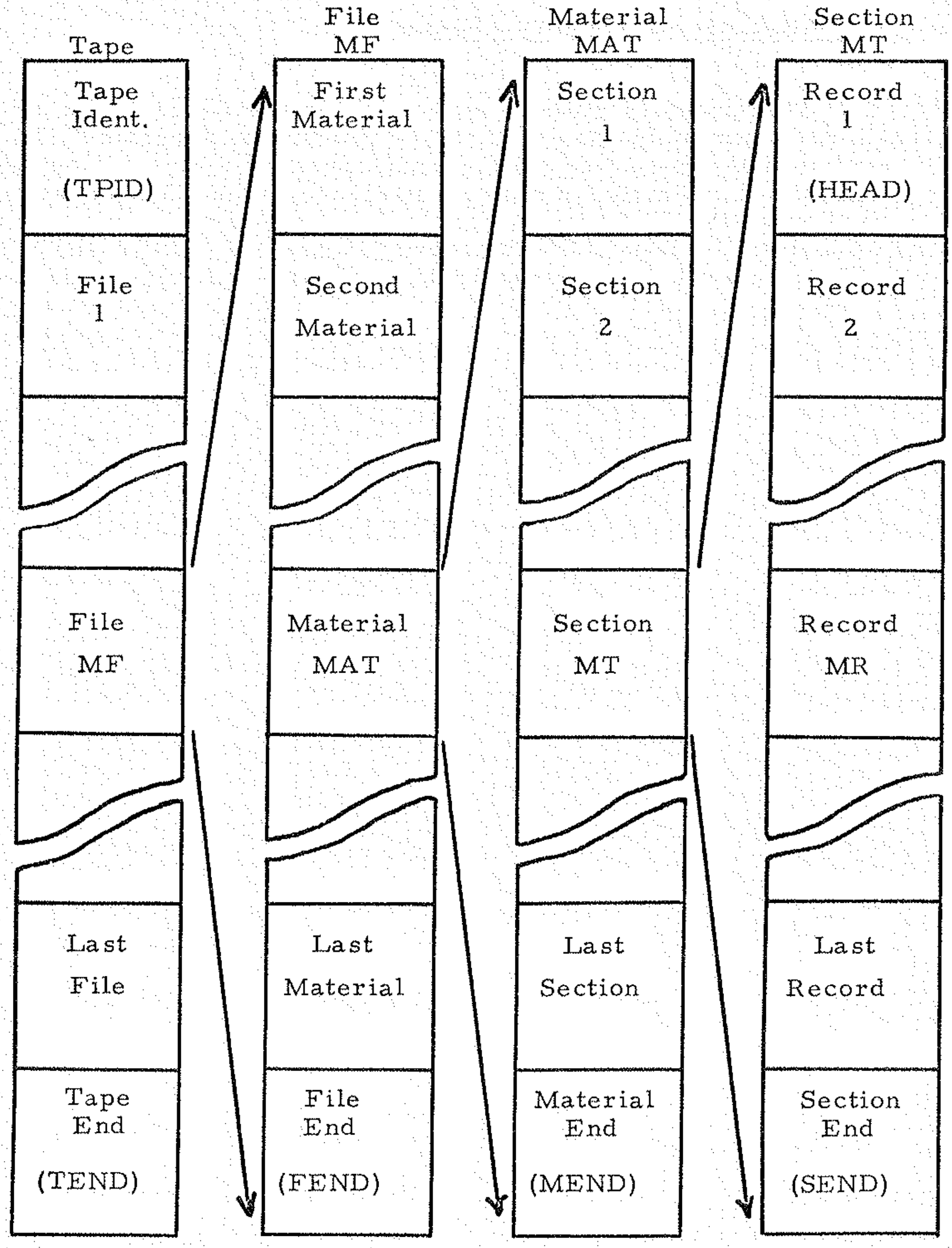


Figure 2.2 Alternate Arrangement of an ENDF/B Tape

2.2.3 Standard Arrangement (Mode 3)

When data is present as BCD card images on tape, the data is arranged as shown in Figure 2.1 with the following change. The records shown in Figure 2.1 are no longer logical records on tape, but are a group of one or more card images each of which forms a logical tape record.

The general format of a punched card is given below.

Field	Columns	Description
1	1-11	Datum
2	12-22	"
3	23-33	"
4	34-44	"
5	45-55	"
6	56-66	"
7	67-70	Material number (MAT)
8	71-72	File number (MF)
9	73-75	Reaction type (MT)
10	76-80	Sequence number starting with 1 for the first card of a material.

2.2.4 Standard Arrangement (Mode 4)

For convenience in reading the data, a special print output mode is provided. The order and arrangement is identicial to the BCD card image tape (Mode 3). The numbers are spread out on the printed page and headings printed.

2.3 Record Types

All data is presented in one of six possible record types. The symbol JT is used to denote the record type.

JT = 1, CØNT Record
 = 2, LIST Record
 = 3, TAB1 Record
 = 4, TAB2 Record
 = 5, HØLL Record
 = 6, TPID Record

The first four types contain numerical data and Control information. The last two types are used to handle Hollerith information.

2.3.1 CØNT Record

The control (CØNT) record consists of nine numbers, MAT, MF, MT, C1, C2, L1, L2, N1, and N2. The Fortran IV read statements for a CØNT record are:

Mode 1

```
      READ(LIB)MAT, MF, MT, C1, C2, L1, L2, N1, N2
```

Mode 2

```
      READ(LIB)MF, MAT, MT, C1, C2, L1, L2, N1, N2
```

Mode 3

```
      READ(LIB, 10)C1, C2, L1, L2, N1, N2, MAT, MF, MT
   10 FØRMAT(2E11.4, 4I11, I4, I2, I3)
```

For convenience we will denote this record by:

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

There are several special cases of the CØNT record denoted by HEAD, SEND, FEND, MEND, and TEND. The HEAD record is the first record of a section, and the number C1 is interpreted as ZA, the (Z,A) designation for the material, and C2 is interpreted as AWR, the ratio of the atomic weight of the material to that of the neutron.

The SEND, FEND, MEND, and TEND records signal the end of a section, file, material, and tape respectively. Only the MAT, MF, and MT fields are used, and C1=C2=L1=L2=N1=N2=0. The following table summarizes the use of these records.

Record Type	Modes 1, 3, 4			Mode 2		
	MAT	MF	MT	MF	MAT	MT
SEND	MAT	MF	0	MF	MAT	0
FEND	MAT	0	0	0	0	0
MEND	0	0	0	MF	0	0
TEND	-1	0	0	-1	0	0

2.3.2 LIST Record

A LIST record is used to list a string of floating point numbers, B_1, B_2, B_3, etc. We assume that these numbers are in the array B(N) and that there are N1 of them. The Fortran IV read statements for a LIST record are:

Mode 1

```
READ(LIB)MAT, MF, MT, C1, C2, L1, L2, N1, N2, (B(N), N=1, N1)
```

Mode 2

```
READ(LIB)MF, MAT, MT, C1, C2, L1, L2, N1, N2, (B(N), N=1, N1)
```

Mode 3

```
    READ(LIB, 10)C1, C2, L1, L2, N1, N2, MAT, MF, MT
10  FØRMAT(2E11.4, 4I11, I4, I2, I3)
    READ(LIB, 20)(B(N), N=1, N1)
20  FØRMAT(6E11.4)
```

While the READ statements could be combined in Mode 3, it was presented as two statements to illustrate that the array B(N) starts on a separate card.

For convenience we will denote the LIST record by:

$[MAT, MF, MT/C1, C2; L1, L2; N1, N2/B_n]$ LIST

2.3.3 TAB1 Record

The TAB1 record contains the data needed to specify a one-dimensional tabulated function. Such a function is illustrated in Figure 2.3.

Define:

X(N)	-	N^{th} value of x, the values in increasing order.
Y(N)	-	N^{th} value of y.
N2	-	The number of values of x (and y) given.
N1	-	The number of regions having different interpolation schemes.
JNT(M)	-	The interpolation scheme used in the M^{th} region.
NBT(M)	-	The value of N separating the M^{th} and $M+1^{st}$ interpolation regions.

Permissible interpolation schemes are given in the following table.

INT	Description
1	constant
2	y linear in x
3	y linear in lnx
4	lny linear in x
5	lny linear in lnx

Interpolation code 1 (constant) implies that the function is constant and equal to the value given at the lower limit of the interval. In the case where a function is discontinuous (for example, when resonance parameters are used to specify the cross section in one range), the value of x is repeated and a pair (x, y) given for each of the two values at the discontinuity (see Figure 2.3).

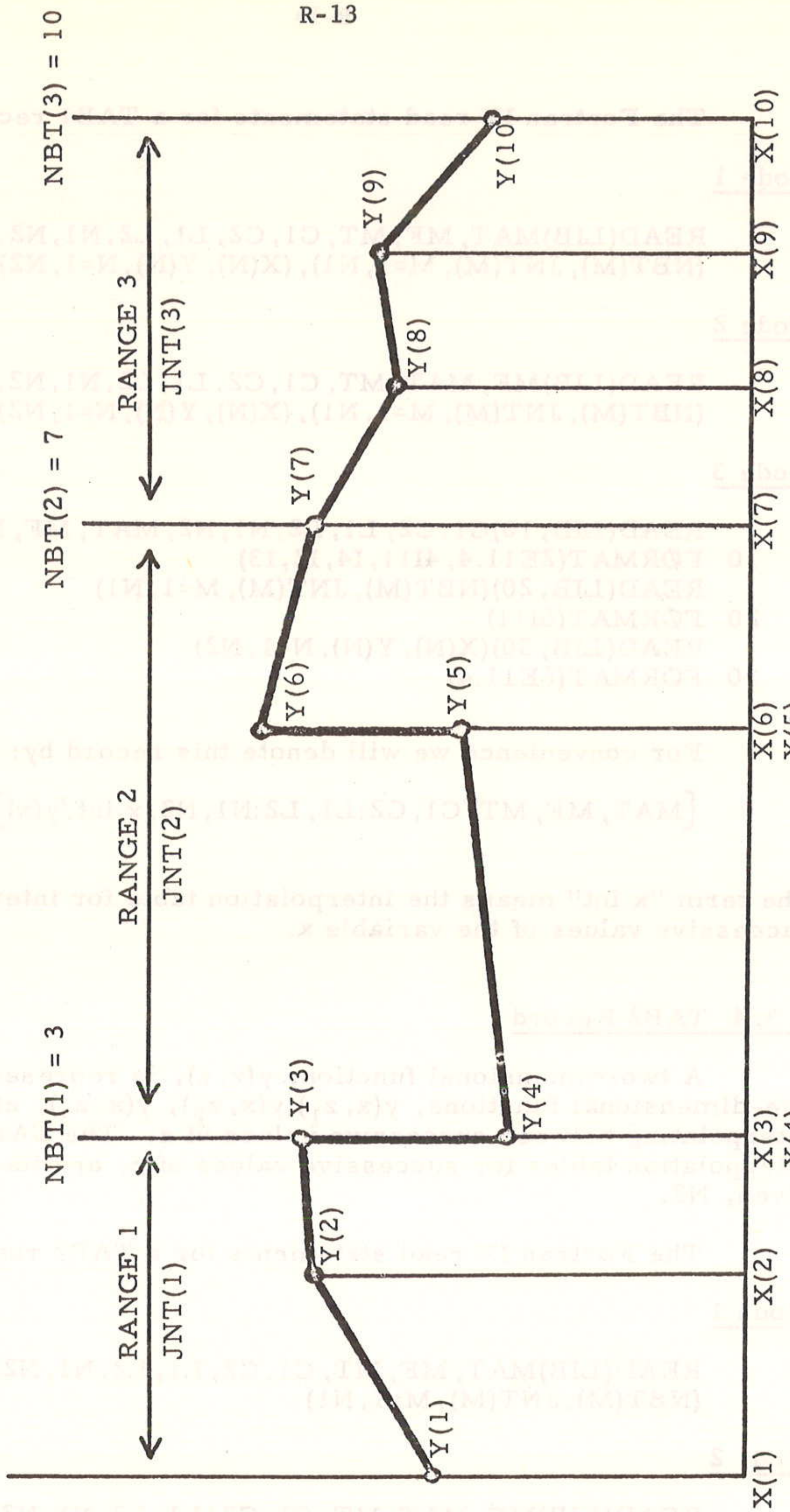


Figure 2.3 Tabulated one dimensional function illustrated for the case N2=10, N1=3

The Fortran IV read statements for a TAB1 record are:

Mode 1

```
READ(LIB)MAT, MF, MT, C1, C2, L1, L2, N1, N2,
(NBT(M), JNT(M), M=1, N1), (X(N), Y(N), N=1, N2)
```

Mode 2

```
READ(LIB)MF, MAT, MT, C1, C2, L1, L2, N1, N2,
(NBT(M), JNT(M), M=1, N1), (X(N), Y(N), N=1, N2)
```

Mode 3

```
   READ(LIB, 10)C1, C2, L1, L2, N1, N2, MAT, MF, MT
10 FØRMAT(2E11.4, 4I11, I4, I2, I3)
   READ(LIB, 20)(NBT(M), JNT(M), M=1, N1)
20 FØRMAT(6I11)
   READ(LIB, 30)(X(N), Y(N), N=1, N2)
30 FØRMAT(6E11.4)
```

For convenience we will denote this record by:

[MAT, MF, MT/C1, C2;L1, L2;N1, N2/x int/y(x)] TAB1

The term "x int" means the interpolation table for interpolating between successive values of the variable x.

2.3.4 TAB2 Record

A two-dimensional function, $y(x, z)$, is represented as a series of one-dimensional functions, $y(x, z_1)$, $y(x, z_2)$, $y(x, z_3)$, etc., plus rules for interpolating between successive values of z. The TAB2 record gives the interpolation tables for successive values of z, and the number of z values given, N2.

The Fortran IV read statements for a TAB2 record are:

Mode 1

```
READ(LIB)MAT, MF, MT, C1, C2, L1, L2, N1, N2,
(NBT(M), JNT(M), M=1, N1)
```

Mode 2

```
READ(LIB)MF, MAT, MT, C1, C2, L1, L2, N1, N2,
(NBT(M), JNT(M), M=1, N1)
```

Mode 3

```
    READ(LIB,10)C1,C2,L1,L2,N1,N2,MAT,MF,MT
 10 FØRMAT(2E11.4,4I11,I4,I2,I3)
    READ(LIB,20)(NBT(M),JNT(M),M=1,N1)
 20 FØRMAT(6I11)
```

For convenience we will denote this record by:

[MAT, MF, MT/C1, C2;L1, L2;N1, N2/z int] TAB2

The TAB2 record would normally be followed by N2 TAB1 or LIST records with the appropriate values of z in the field C2.

2.3.5 HØLL Record

Hollerith comment data is used in File 1 and is contained in a HØLL Record. The Fortran IV read statements for a HØLL record are:

Mode 1

```
    READ(LIB)MAT,MF,MT,C1,C2,L1,L2,N1,N2,(H(N),N=1,N1)
```

Mode 2

```
    READ(LIB)MF,MAT,MT,C1,C2,L1,L2,N1,N2,(H(N),N=1,N1)
```

Mode 3

```
    READ(LIB,10)C1,C2,L1,L2,NCD,N2,MAT,MF,MT
 10 FØRMAT(2E11.4,4I11,I4,I2,I3)
    N1=17*NCD
    READ(LIB,20)(H(N),N=1,N1)
 20 FØRMAT(16A4,A2)
```

For convenience we will denote this record by:

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

Note that in Modes 1 and 2 it is identical to a LIST record. The A4 format was selected because it will work on all computers, whereas wider fields will not work on an IBM/360.

Note also that NCD (the number of comment cards) is given in the N1 field of the first card of the record in Mode 3.

2.3.6 TPID Record

In the original ENDF/B specifications, the TPID record was a special case of the CØNT record. Rather than putting zeros in fields C1, C2, ... N2 of a TPID card, it is frequently convenient to use Hollerith information to further identify a tape. Trouble occurs if this Hollerith information is read under an E or I format.

In Mode 3 (card image), the TPID card may have Hollerith remarks in columns 1-66, and a tape ident. number (NTID) in columns 67-70. The Fortran IV read statements for a TPID record are:

Mode 1, 2

```
      READ(LIB)NTID, z, z, (H(N), N=1, 17)
```

Mode 3

```
      READ(LIB, 10)(H(N), N=1, 17), NTID
   10 FØRMAT(16A4, A2, I4)
```

z is a dummy location. A positive value of NTID should be used for Modes 1 and 3 (Standard arrangement), and a negative value for Mode 2 (Alternate arrangement).

2.4 Temperature Dependence

Any of the data in files 3-7 may have a temperature dependence (where physically realistic) specified by repeating the data for each temperature given and indicating how to interpolate between tabulated temperatures. Since the data will always be given in a LIST or TAB1 record, we consider a TAB1 record for the function y(x). In this case, we must write y(x, T). We constrain this function in the following way. The set of x values and the interpolation between successive x values must be the same at all temperatures. Define:

T_m - Temperature (^oK). These must be listed in increasing order.

LT - A test for temperature dependence.

LT=0, no temperature dependence

LT>0, the function is given at LT+1 temperatures.

I_m - Interpolation scheme used between T_{m-1} and T_m.

The function at the first temperature, $y(x, T_1)$, is given in a TAB1 record. The function at the remaining temperatures is given in LIST records.

$[MAT, MF, MT/ T_1, C2; LT, L2; NR, NP/ x\ int/ y(x, T_1)]$ TAB1

$[MAT, MF, MT/ T_2, C2; I_2, L2; NP, 0/ y_n(T_2)]$ LIST

$[MAT, MF, MT/ T_3, C2; I_3, L2; NP, 0/ y_n(T_3)]$ LIST

---- --- --- --- -- -- --- --- ---- ----- ----

There will be a total of LT records of the LIST type, each containing only the list of y values.

If the temperature dependence refers to data already in a LIST record, all records are of the LIST type.

$[MAT, MF, MT/ T_1, C2; LT, L2; N1, 0/ B_n(T_1)]$ LIST

$[MAT, MF, MT/ T_2, C2; I_2, L2; N1, 0/ B_{n(T_2)}]$ LIST

$[MAT, MF, MT/ T_3, C2; I_3, L2; N1, 0/ B_n(T_3)]$ LIST

---- --- --- -- -- -- -- -- -- ------ ----

3. Storage Allocation

Four types of storage areas are used by the ENDF/B retrieval subroutines.

1. Input and output tapes
2. An area in fast core storage where a single record can be stored and each item of the record separately identified.
3. An area in fast core storage where many records can be stored in a dense manner.
4. Bulk storage such as disks or tapes.

These storage areas are illustrated in Figure 3.1. The symbols alongside of the arrows denote subroutines used to transmit data between storage areas.

3.1 Expanded Record Storage, /RECS/

The main storage area used to transmit to and from input and output tapes is denoted by /RECS/. It is defined by the statements

```
      COMMON/RECS/MAT, MF, MT, C1, C2, L1, L2, N1, N2, NBT(100), JNT(100),
     1             X(2000), Y(2000), B(2000), N1X, N2X, NS
```

Most of the symbols have been previously defined, the remaining ones are defined as

N1X	-	The length of the arrays NBT and JNT (=100)
N2X	-	The length of the arrays X, Y, and B (=2000)
NS	-	Sequence number for card punching

The quantities N1X and N2X should be defined in the main program using /RECS/.

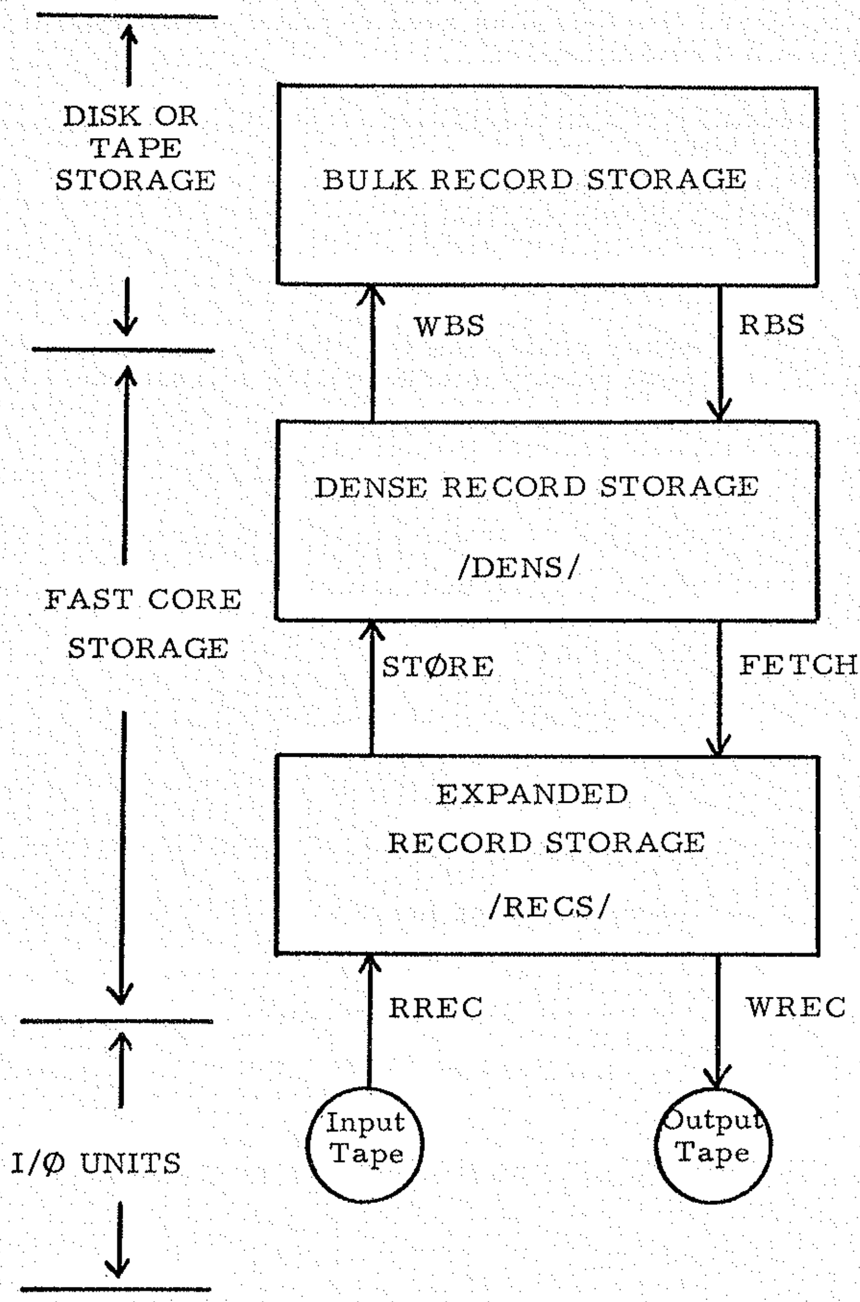


Figure 3.1 Schematic of Storage Allocation

3.2 Dense Record Storage, /DENS/

A LIST record may have up to 2009 entries, a TAB1 record may have up to 4209 entries, but the average number of entries may be only a few hundred. We define an area of fast core storage to store these records in a dense manner.

```
COMMON/DENS/JMT(100),JAT(100),JTT(100),JLT(100),A(5000),
JNS, MNS, JX, MX
DIMENSION LA(1)
EQUIVALENCE (A, LA)
```

where

A(J) - Array containing the ENDF/B records. The data within each record is ordered the same as on a binary tape, standard arrangement (Mode 1). LA(J) is equivalent to A(J).

JMT(M) - An arbitrary identification of the M^{th} record in A.

JAT(M) - Starting location (J value) in the array A of the M^{th} record.

JTT(M) - Type of the M^{th} record.

JT=1, CØNT	JT=4, TAB2
=2, LIST	=5, HØLL
=3, TAB1	=6, TPID

JLT(M) - Number of locations in array A used by the M^{th} record.

JNS - An index giving the next available location in the array A.

MNS - An index giving the next available location in the JMT, JAT, JTT, and JLT tables.

JX - The length of the array A (=5000).

MX - The length of the JMT, JAT, JTT, and JLT tables (=100).

The following tables indicate how the records are stored in the array A.

CØNT Record

JT = JAT(M) = 1		Record Type
JA = JAT(M)		Starting location
JL = JLT(M) = 9		Record length

MAT = LA (JA)
MF = LA (JA+1)
MT = LA (JA+2)
C1 = A (JA+3)
C2 = A (JA+4)
L1 = LA (JA+5)
L2 = LA (JA+6)
N1 = LA (JA+7)
N2 = LA (JA+8)

LIST Record

JT = JTT(M) = 2	Record type
JA = JAT(M)	Starting location
JL = JLT(M) = N1+9	Record length

MAT = LA (JA)
MF = LA (JA+1)
MT = LA (JA+2)
C1 = A (JA+3)
C2 = A (JA+4)
L1 = LA (JA+5)
L2 = LA (JA+6)
N1 = LA (JA+7)
N2 = LA (JA+8)
B(1) = A (JA+9)
B(2) = A (JA+10)

B(N) = A(JA+N+8)

B(N1) = A(JA+N1+8)

TAB1 Record

```
JT     = JTT(M) = 3                  Record type
JA     = JAT(M)                      Starting location
JL     = JLT(M) = 2*N1 +2*N2 +9      Record length

MAT    = LA(JA)
MF     = LA(JA+1)
MT     = LA(JA+2)
C1     =  A(JA+3)
C2     =  A(JA+4)
L1     = LA(JA+5)
L2     = LA(JA+6)
N1     = LA(JA+7
N2     = LA(JA+8)
NBT(1) = LA(JA+9)
JNT(1) = LA(JA+10)
------  ---------
NBT(N)= LA(JA+2*N+7)
JNT(N) = LA(JA+2*N+8)
------  ------------
X(1)   =  A(JA+2*N1+9)
Y(1)   =  A(JA+2*N1+10)
----      --------------
X(N)   =  A(JA+2*N1+2*N+7)
Y(N)   =  A(JA+2*N1+2*N+8)
----      ----------------
X(N2)  =  A(JA+2*N1+2*N2+7)
Y(N2)  =  A(JA+2*N1+2*N2+8)
```

TAB2 Record

JT	= JTT(M) = 4	Record type
JA	= JAT(M)	Starting location
JL	= JLT(M) = 2*N1+9	Record length

MAT = LA(JA)
MF = LA(JA+1)
MT = LA(JA+2)
C1 = A(JA+3)
C2 = A(JA+4)
L1 = LA(JA+5)
L2 = LA(JA+6)
N1 = LA(JA+7)
N2 = LA(JA+8)
NBT(1) = LA(JA+9)
JNT(1) = LA)JA+10)

NBT(N) = LA(JA+2*N+7)
JNT(N) = LA(JA+2*N+8)

NBT(N1) = LA(JA+2*N1+7)
JNT(N1) = LA(JA+2*N1+8)

HØLL Record

JT	= JTT(M) = 5	Record Type
JA	= JAT(M)	Starting Location
JL	= JLT(M) = N1+9	Record Length

MAT	= LA(JA)
MF	= LA(JA+1)
MT	= LA(JA+2)
C1	= A(JA+3)
C2	= A(JA+4)
L1	= LA(JA+5)
L2	= LA(JA+6)
N1	= LA(JA+7)
N2	= LA(JA+8)
B(1)	= LA(JA+9)
B(2)	= LA(JA+10)

B(N)	= A(JA+N+8)

B(N1)	= A(JA+N1+8)

TPID Record

JT	= JAT(M) = 6	Record type
JA	= JAT(M)	Starting location
JL	= JLT(M) = 20	Record length

NTID	= LA(JA)
	= LA(JA+1)
	= LA(JA+2)
B(1)	= A(JA+3)
B(2)	= A(JA+4)
----	--------
B(N)	= A(JA+N+2)
----	-------
B(17)	= A(JA+19)

3.3 Bulk Storage

It is anticipated that subroutines will be needed to transfer records between /DENS/ and bulk storage such as disks, drums, or tapes. These routines have not yet been specified or written.

4. Error Stops

In many cases, subroutines are provided with an error flag in their calling sequences. In cases where the error cannot be corrected and no further computation is possible, the program stops by calling subroutine ERRØR.

4.1 Subroutine ERRØR

The subroutine ERRØR is called with the statement

```
CALL ERRØR (N)
```

where N is the error stop number. The subroutine prints the message "ERRØR STØP (N)" and terminates the run. The current values of MAT, MF, MT contained in /RECS/ are also printed. More extensive diagnostics can be printed if desired.

4.2 List of Error Stops

The list below is a resume of the error stops thus far assigned. Consult the appropriate subroutine description for the detailed explanation of the stop.

Error Stop Number	Subroutine
99-107	RREC
108-109	WREC
110	ECSI
130-132	TERP2
133-135	TERP1
300-301	STØRE
310-311	CØMB
312-313	CRØP
314	IPDS
315	GRATE

5. Interpolation

The data presented in TAB1 (or following a TAB2) record are discrete values. Intermediate points are obtained by interpolation between the nearest two adjacent points. Five interpolation formulas are allowed and specified by an interpolation code, I.

I	Interpolation scheme
1	y constant and equal to the value at the lower end of the interval
2	y linear in x
3	y linear in ln x
4	ln y linear in x
5	ln y linear in ln x

5.1 Subroutine TERP1

The subroutine TERP1 computes an interpolated value Y given the desired X, the end points of the line (X1, Y1) and (X2, Y2), and the interpolation code I to be used. The calling sequence is

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

The interpolation equations used are:

I	Interpolation equation
1	Y = Y1
2	Y = Y1 + (X-X1)*(Y2-Y1)/(X2-X1)
3	Y = Y1 + ln (X/X1)*(Y2-Y1)/ln (X2/X1)
4	Y = Y1*exp [(X-X1) * ln (Y2/Y1)/(X2-X1)]
5	Y = Y1*exp [ln (X/X1)*ln(Y2/Y1)/ln (X2/X1)]

Input checking routines will assure that the original data does not contain inconsistancies such as specifying I=5 and having one or more zero or negative values of x and y. It is conceivable that subsequent manipulations of the data may introduce such inconsistancies. TERP1 automatically investigates the end points X1 and X2 and, if either are negative or zero and ln x is used in the interpolation formula (I=3, 5), the interpolation formula is changed to I=2 or 4. Similarly, if Y1 or Y2 are negative or zero and I=4 or 5, I is changed to 2 or 3. If X is negative or zero, and I=3 or 5, an error stop occurs.

The following error stops are used.

Number	Meaning
133	Interpolation code not in range 1-5
134	Zero or negative X cannot be interpolated by logs.
135	X1 = X2, discontinuity

The following program example illustrates the use of TERP1 and the interpolation tables in computing a value of YP for a given XP from a TAB1 function stored in /RECS/.

```
      YP=0.0
      IF (XP.LT.X(1)) GØ TØ 50
      DØ 10 N=2,N2
      IF (XP.LT.X(N)) GØ TØ 20
   10 CONTINUE
      IF (XP.GT.X(N2)) GØ TØ 50
   20 DØ 30 M=1,N1
      IF (N.LE.NBT(M)) GØ TØ 40
   30 CØNTINUE
      (Error in interpolation table)
   40 I=JNT(M)
      CALL TERP1 (X(N-1),Y(N-1),X(N),Y(N),XP,YP,I)
   50 CØNTINUE
```

5.2 Subroutine TERP2

Given a table XP(N), N=1, NX, the subroutine TERP2 will compute the corresponding YP(N), N=1, NX by interpolating in the TAB1 record in /RECS/. This would be a straightforward task using TERP1, except that discontinuities must be dealt with.

Consider the case where YP(N) is to be determined. The TAB1 record is searched to find the X(M) such that X(M-1) < XP(N) < X(M). If a suitable M can be found, the value YP(N) is obtained by interpolation between Y(M-1) and Y(M). If, however, an M is found such that X(M) = XP(N), there are several possibilities. For example, if X(M-1) = X(M), there is a discontinuity. If XP(N) = XP(N-1), there is also a discontinuity in the XP mesh and YP(N) should be set equal to Y(N). If XP(N) = XP(N+1), then YP(N) should be set equal to Y(N+1). The TERP2 subroutine senses these discontinuities and, when they occur in both the X and XP meshes, lines them up properly to obtain the correct values of YP. If a discontinuity is sensed in the X mesh but not in the XP mesh, there is no unique procedure, and TERP2 simply computes the average of the values on either side of the discontinuity.

The following error stops are used:

Number	Explanation
130	X(N) not in increasing order
131	XP(N) not in increasing order
132	Interpolation table incorrect

6. Data Transmission

The subroutines used to transmit data between I/Ø units and storage areas are described in this section. They are:

RREC and WREC

Subroutines to transmit records between I/Ø units and storage area /RECS/.

STORE, FETCH, and DELETE

Subroutines to transmit records between storage areas /RECS/ and /DENS/.

RBS and WBS

Subroutines to transmit records between /DENS/ and bulk storage.

CRØP

Subroutine to crop and compress a TAB1 record in /RECS/ and eliminate data outside of the desired range.

LRIDS

Subroutine to locate a record in /DENS/.

FPDS and IPDS

Subroutines to fetch or interpolate points from a TAB1 record in /DENS/.

6.1 Subroutines RREC and WREC

Data is transmitted between the I/Ø tapes and storage area /RECS/ with subroutines RREC (Read RECord) and WREC (Write RECord). The appropriate Fortran statements are:

```
CALL RREC (JT, NT, MØDE, T)
CALL WREC (JT, NT, MØDE)
```

where

JT = 1, CØNT record
   = 2, LIST record
   = 3, TAB1 record
   = 4, TAB2 record
   = 5, HØLL record
   = 6, TPID record

NT = Input or output tape number

MØDE = 1, Binary tape, standard arrangement
   = 2, Binary tape, alternate arrangement
   = 3, BCD card image tape, standard arranement
   = 4, Expanded and interpreted tape for printing, standard arrangement.

T = Temperature (^{o}K). If T $\geq$ 0, and if the record specified has a temperature dependence, the record will be evaluated at T. If T < 0, temperature dependence will be ignored and only the first record read in.

After the RREC subroutine is called, the data appears in /RECS/. The detailed read statements are given in section 2. Table 3.1 indicates where items defined in the original ENDF/B specifications will be found in /RECS/.

Temperature dependence is treated in the following manner. Assume that the data in TAB1 (or LIST) record has a temperature dependence, in which case it is followed by LIST records containing the data at progressively higher temperatures. If the data is wanted at a specific temperature, T, for calculational purposes, the statement

```
CALL RREC (3,NT,MØDE, T)
```

would read in the TAB1 record and all the following LIST records. When the proper records had been read in bracket T, the data would be automatically interpolated, and a TAB1 record at temperature T would appear to /RECS/. If, on the other hand, the data for all temperatures were to be transmitted from tape NTA to tape NTB, the following statements would be used.

```
100 CALL RREC(3,NTA,MØDE,-1.0)
    CALL WREC(3,NTB,MØDE)
    IF (MF.EQ.2)GØ TØ 120
    IF(L1.EQ.0) GØ TØ 120
    DØ 110 L=1,L1
    CALL RREC (2,NTA,MØDE,-1.0)
110 CALL WREC (2,NTB,MØDE)
120 CØNTINUE
```

Note that temperature dependent LIST and TAB1 records are not allowed in File 2 since the field L1 is used for other purposes.

Subroutine WREC uses a set of punching and printing subroutines in Modes 3 and 4. The required routines are:

PRCØNT	PUCØNT
PRLIST	PULIST
PRTAB1	PUTAB1
PRTAB2	PUTAB2
PRHØLL	PUHØLL
PRTPID	PUTPID
CXFP	

The following error stops are used.

Number	Subroutine	Error
99	RREC	NT not defined
100	"	JT not in range 1-6
101	"	MØDE not in range 1-3
102	"	T not in range given by data
103	"	Interpolation table too long or 0
104	"	List too long or 0
105	"	Tabulation too long or less than 2
106	"	Improper temperature dependence
107	"	MAT, MF, MT incorrect for JT=2, 3, 4, 5
108	WREC	JT not in range 1-6
109	"	MØDE not in range 1-4
133	TERP1	Interpolation code not in range 1-5
134	"	Zero or negative value cannot be interpolated with logs.
135	"	Discontinuity encountered.

TABLE 3.1

Correspondance between symbols used in /RECS/

and those used in ENDF/B specs

Symbols in /RECS/	Record Types				
	CØNT	LIST	TAB1	TAB2	HØLL
MAT	MAT	MAT	MAT	MAT	MAT
MF	MF	MF	MF	MF	MF
MT	MT	MT	MT	MT	MT
C1	C1	C1	C1	C1	C1
C2	C2	C2	C2	C2	C2
L1	L1	L1	L1	L1	L1
L2	L2	L2	L2	L2	L2
N1	N1	N1	NR	NR	NH
N2	N2	N2	NP	NZ	N2
NBT			NBT	NBT	
JNT			INT	INT	
X			X		
Y		**	Y		
B		B	**		H
N1X					
N2X					
NS					

** May be used as temporary storage if record depends on temperature.

6.2 Subroutines STØRE, FETCH, and DELETE

Records are transmitted between the /RECS/ and /DENS/ storage areas with subroutines STØRE and FETCH. Records in /DENS/ no longer required can be deleted with DELETE.

The subroutine DELETE serves two purposes: clear /DENS/ storage and initialize counters JNS and MNS, and delete a record no longer needed. The Fortran IV statement needed to use the subroutine is:

```
CALL DELETE (MA)
```

where MA is the identification number of the record to be deleted. If MA=0, all records in /DENS/ are set to zero, and MNS and JNS are set to unity. The subroutine should be entered with MA=0 at the beginning of the main program using /DENS/ storage. After a record has been deleted, all tables in /DENS/ are closed up so that there are never any gaps in /DENS/.

A record in /RECS/ can be stored in /DENS/ with the instruction

```
CALL STØRE (JT, MA, LØF)
```

where JT = Record Type
- = 1, CØNT record
- = 2, LIST record
- = 3, TAB1 record
- = 4, TAB2 record
- = 5, HØLL record
- = 6, TPID record

MA= record identification

LØF= overflow test
- = 0, normal return, record was stored
- = 1, overflow, record will not fit into /DENS/

The record identification (MA) is arbitrarily assigned to the record when the store instruction is given. As long as the same MA is used in later DELETE or FETCH instructions, no trouble will occur. If a record with the same MA is already in /DENS/, the old record will be deleted, the tables closed up, and the new record stored. Thus, it is not possible to have two records in /DENS/ with the same MA. MA is normally an integer. Care has been taken throughout the codes to use only logical IF statements to test MA, so that a string of Hollerith characters may also be used for MA.

The overflow test could have been made an error stop. However, it was felt more desirable to provide it as a program option so that, if bulk storage is available, selected records can be dumped from /DENS/ to make room for the record which currently will not fit.

A record in /DENS/ can be moved back into /RECS/ with the instruction

CALL FETCH (MA, LNT)

where MA = record identification
LNT = 0, normal return, record transferred
= 1, record MA not in /DENS/

After the transfer, the record is in both /RECS/ and /DENS/. If the record is no longer needed, a DELETE (MA) instruction should be given.

The following error stops are used.

Number	Subroutine	Error
300	STØRE	JT not in the range 1-6
301	STØRE	MA=0 not allowed

6.3 Subroutines RBS and WBS

These routines have not yet been specified.

6.4 Subroutine CRØP

Subroutine CRØP operates on a TAB1 record in /RECS/ and does the following three things:

1. Given a range of XL $\leq$ x $\leq$ XH, those portions of the TAB1 record outside the range are discarded.

2. If the TAB1 record is not defined in some part of the range XL $\leq$ x $\leq$ XH, the record is completed by adding points with y=0.

3. The TAB1 record is compressed by eliminating points which can be interpolated from adjacent points to a relative accuracy of EPS.

Thus, CRØP is used to reduce the size of a TAB1 function by compression and by eliminating unnecessary regions. If the TAB1 record is to be moved into /DENS/ for later use, considerable storage may be saved by first using CRØP.

The following statement is used to crop a TAB1 record in /RECS/.

CALL CRØP (XL,XH, EPS, LØF)

XL and XH are the low and high limits of the range of x to be kept. EPS is the relative error criterion used for compression. EPS=0 bypasses compression. LØF is an overflow test which is normally zero. It is possible that adding extra zero points to complete the range may cause the record to overflow /RECS/. This overflow is signalled with LØF=1.

6.5 Subroutine LRIDS

Subroutine LRIDS is a short program to locate the starting location of a record in /DENS/. The calling sequence is

CALL LRIDS (MA, JA, LNT)

MA is the identification number of the desired record in /DENS/. JA is computed by the subroutine and is the starting index in array A(J) in /DENS/ for the record whose identification number is MA. LNT will be zero if the record MA was located, and LNT=1 if record MA is not in /DENS/.

6.6 Subroutines FPDS and IPDS

Assume that a TAB1 record is stored in /DENS/ starting at location JA (as obtained from LRIDS). Denote this TAB1 record by X(N), Y(N), N=1, N2 and NBT(N), JNT(N), N=1, N1. Subroutine FPDS is used to fetch one of these tabulated points (the one denoted by N=NP) and the interpolation code IP used in the panel between X(NP) and X(NP+1). The calling sequence is:

CALL FPDS (JA, NP, XP, YP, IP)

Note that JA and NP are input to FPDS, and XP, YP, and IP are output from FPDS. If NP is outside of the range of the table, appropriate values are supplied according to the following table.

NP	XP	YP	IP
$NP \leq 0$	-10^{+20}	0.0	1
$0 < NP < N2$	X(NP)	Y(NP)	IP
NP=N2	X(N2)	Y(N2)	2
NP=N2+1	X(N2)	0.0	2
$NP > N2+1$	10^{+20}	0.0	2

Subroutine IPDS is used to obtain an interpolated point from the TAB1 record. The calling sequence is:

CALL IPDS (JA, NP, XP, YP, IP)

Here, JA, NP, and XP (the desired x value) are input to IPDS, and NP, YP (the interpolated value of y at x=XP), and IP (interpolation code used) are output from IPDS. On input NP is used to start the search for the appropriate panel, on output it is such that $X(NP) \leq XP < X(NP+1)$. If $XP < X(1)$, the subroutine returns with YP=0.0, NP=0, IP=1. If $XP > X(N2)$, the subroutine returns with YP=0.0, NP=N2+1, IP=1.

Note that in all cases the subroutines supply zero values when the tabulated range is exceeded.

7. Representation and Generation of Functions

There are many algebraic manipulations required to process ENDF/B data. For example,

1. add partial cross sections to obtain the total cross section

2. multiply a cross section by a probability distribution to obtain a secondary angular or energy distribution.

3. Divide captive cross sections by fission cross sections to obtain alpha.

4. Multiply cross sections by weighting spectra.

On the ENDF/B tape, functions may be expressed as either tabulated (TAB1) or analytic (Maxwellian, fission spectrum, evaporation, etc.) functions. The number of analytic expressions needed to perform algebraic manipulations on any combination of these functions would be enormous. Matters are greatly simplified if only one functional representation is used. The TAB1 function is chosen as the common representation since it is the most general and flexible representation. A later section will describe subroutines which algebraically combine TAB1 functions. Before that, the methods used to generate suitable TAB1 functions from analytic functions will be described.

7.1 General Method, Subroutine GENT1

As an example, assume that we wish to calculate the Maxwellian average value of a cross section. The cross section is given as a TAB1 function, and the Maxwellian is given as an analytic formula. Since subroutines will be available for multiplying and integrating TAB1 functions, we need to convert the analytic formula to a TAB1 function. This can be done with the statement

CALL GENT1 (FUNC, CØN, XL, XH, EPS, LØF)

where

FUNC - The name of a Fortran IV function subroutine which will calculate a value Y of the function at a given value of X. It is called by the statement Y=FUNC(X, CØN)

CØN(N) - A list of auxiliary constants needed to evaluate the function.

XL - The lower limit of X to be used.

XH	-	The upper limit of X to be used
EPS	-	A relative error criterion, ε
LØF	=	0, normal return
	=	1, TAB1 record to large

The resulting TAB1 function is found in /RECS/.

In brief, the problem is stated:

<u>given</u>: A continuous analytic function, y(x)

<u>Find</u>: A set of x values (and corresponding y values) in the range $XL \leq x \leq XH$ such that interpolation between successive x values will yield y values which are accurate to a relative error of ε. An arbitrary interpolation code may be used.

The procedure used is as follows (see Figure 7.1).

1. Initialize counters and compute $\Delta = (XH-XL)/100$

2. Compute values of y at 5 equally spaced (by Δ) values of x. Call these values $x_1, x_2, \ldots x_5$, and $y_1, y_2, \ldots y_5$.

3. Using points 1 and 5 as the end points, interpolate values at points 2, 3, and 4 using all five interpolation codes. Calculate the difference between the exact values and interpolated values for each interpolation code.

4. If no one of the interpolation codes predicts values at points 2, 3, and 4 which are all accurate to ε, halve the spacing and return to step 2.

5. If one or more of the interpolation codes predict values at points 2, 3, and 4 which are all accurate to ε, add another point 6 at a spacing of Δ from point 5. Repeat steps 3 and 4 testing intermediate points and adding new points until either point 9 was the last successful point, or the error criterion was not satisfied.

6. If point 9 was the last point generated, and points 2-8 could all be successfully interpolated by at least one code, double the spacing and return to step 2.

7. If the error criterion fails at point n (n=6, 7, 8, or 9) point n-1 is accepted and stored in /RECS/. This will also be stored as point 1 for use when we later return to step 2.

8. The interpolation code is chosen as either the last interpolation code stored if it produced an acceptable error, or that interpolation code (different from the last one used) which resulted in the smallest interpolation error.

9. Return to step 2 until x exceeds XH.

This hunting method will work well for smoothly varying functions with broad structure. It will not work for discontinuous functions and may miss features such as sharp resonances. The method has the advantage that only the function (no derivatives) is required.

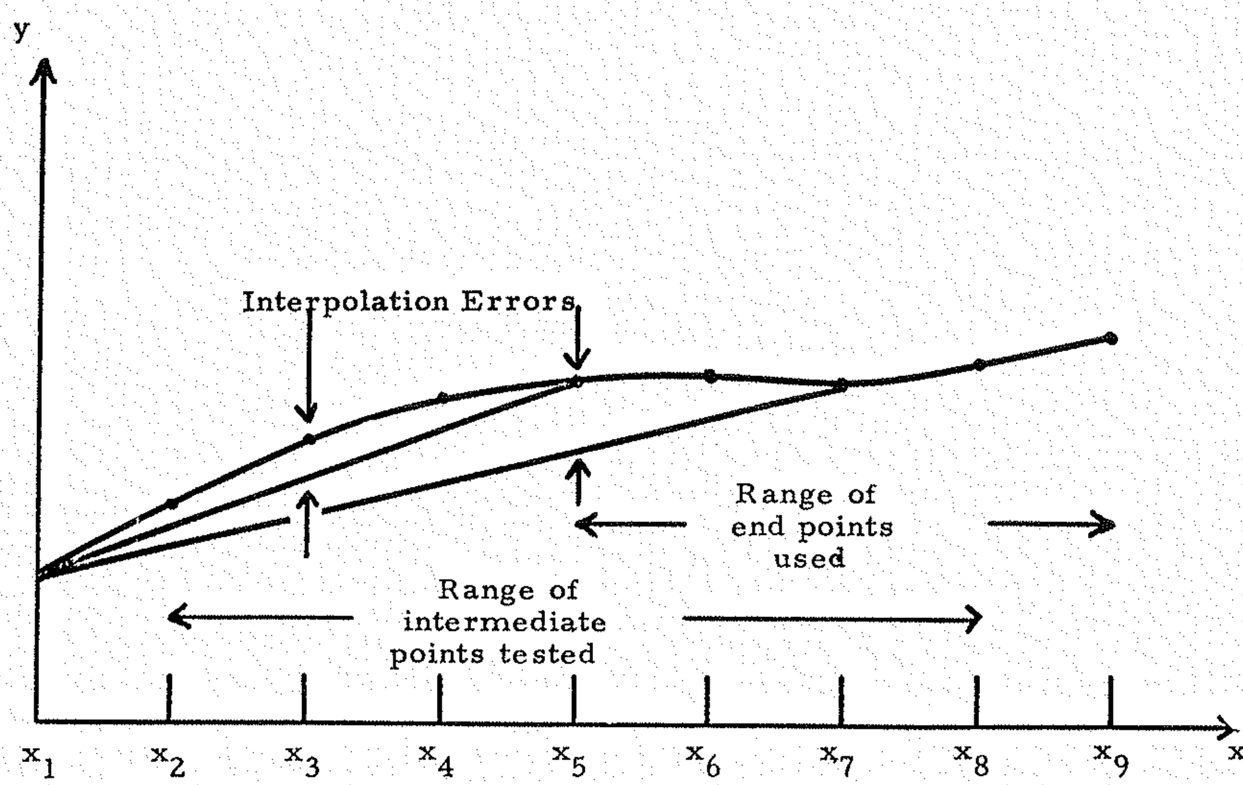


Figure 7.1 Illustration of method used in GENT1

7.2 Some Function Subroutines Used by GENT1

The following is a list of available function subroutines which may be used with GENT1. In every case the calling sequence will be

y = FUNC (x, CØN)

where FUNC is the function name, x is the argument, and CØN(N) is a list of constants set up by the calling program.

1. Maxwellian Distribution (MAXW)

$$y = (x/x_o^2)\, e^{-x/x_o}$$

CØN(1) = x_o

2. Simple Fission Spectrum (FISS)

$$y = \sqrt{\frac{4x}{\pi x_o^3}}\, e^{-x/x_o}$$

CØN(1) = x_o

3. Watt Spectrum (WATT)

$$y = \sqrt{\frac{4}{\pi b a^3}}\, e^{-ab/4}\, e^{-x/a} \quad \sinh\sqrt{bx}$$

CØN(1) = a
CØN(2) = b

4. Power of x (AXPC)

$$y = ax^c$$

CØN(1) = a
CØN(2) = c

5. Polynomial (PØLY)

$$y = \sum_{n=0}^{NT-1} A_n x^n$$

CØN(1) = NT (floating point)
CØN(2) = A_0
CØN(3) = A_1

CØN(NT+1) = A_{NT-1}

6. Legendre Polynomial (LEGP)

$$y = \sum_{n=0}^{NT-1} \left(\frac{2n+1}{2}\right) A_n P_n(x)$$

CØN(1) = NT
CØN(2) = A_0
CØN(3) = A_1
....
CØN(NT+1) = A_{NT-1}

8. Integration of TAB1 Functions

A TAB1 function in /RECS/ can be integrated using subroutine GRATE. The GRATE subroutine uses subroutine ECSI to compute integrals over a single panel.

8.1 Subroutine ECSI

Consider a panel whose end points are (x_3, y_3) and (x_4, y_4). Let x_1 and x_2 be the integration limits. The integral over this range is given by

$$B = \int_{x_1}^{x_2} y(x)dx$$

and the detailed expression depends on the interpolation code I applicable to the panel.

I=1

$$B = (x_2 - x_1) y_3$$

I=2

$$B = (x_2 - x_1)\left[a + \frac{1}{2} b\,(x_2 + x_1)\right]$$

$$b = (y_4 - y_3)/(x_4 - x_3)$$

$$a = y_3 - b x_3$$

I=3

$$B = (x_2 - x_1)\left[y_3 + b \ln\left(\frac{x_1}{x_3}\right)\right] + b x_1 \left[1 + \frac{x_2}{x_1}\left(\ln\frac{x_2}{x_1} - 1\right)\right]$$

$$b = (y_4 - y_3)/\ln(x_4/x_3)$$

$$\left[1 + \frac{x_2}{x_1}\left(\ln\frac{x_2}{x_1} - 1\right)\right] = \frac{1}{2} z^2 \left[1 - \frac{1}{3} z + \frac{1}{6} z^2 - \frac{1}{10} z^3\right], \quad |z| \leq 0.1$$

$$z = (x_2 - x_1)/x_1$$

I=4

$$B = e^{a + b x_1} (e^z - 1)/b, \qquad |z| > 0.1$$

$$= e^{a + b x_1} (x_2 - x_1)\left(1 + \frac{1}{2} z + \frac{1}{6} z^2\right), \qquad |z| \leq 0.1$$

$$b = \ln(y_4/y_3)/(x_4-x_3)$$

$$a = \ln y_3 - bx_3$$

$$z = (x_2-x_1)b$$

I=5

$$B = y_3 x_1 \left(\frac{x_1}{x_3}\right)^b \frac{1}{b+1} \left[\left(\frac{x_2}{x_1}\right)^{b+1} - 1\right]$$

$$b = \ln(y_4/y_3) / \ln(x_4/x_3)$$

$$\frac{1}{b+1} \left[\left(\frac{x_2}{x_1}\right)^{b+1} - 1\right] = \ln\left(\frac{x_2}{x_1}\right)\left(1 + \frac{1}{2}z + \frac{1}{6}z^2\right), \quad |z| \leq 0.1$$

$$z = (b+1)\ln(x_2/x_1)$$

The subroutine is called with the statement:

CALL ECSI (X3, Y3, X4, Y4, X1, X2, I, B)

where (X3, Y3) and (X4, Y4) are the end points of the panel, X1 and X2 are the integration limits, I is the interpolation code, and B is the computed integral.

8.2 Subroutine GRATE

Let XL and XH be the required integration limits for a TAB1 record in /RECS/. The integration is accomplished with the statement

CALL GRATE (XL, XH, ANS)

where ANS is the computed integral.

GRATE operates by finding the panels in the TAB1 record in which XL and XH occur. Indices NL and NH are defined so that $X(NL) \leq XL < X(NL+1)$ and $X(NH) < XH \leq X(NH+1)$. Subroutine ECSI is then used to compute the integral in each panel for $NL \leq N \leq NH$ and the results summed to obtain the total integral.

The following error stop is used:

Number	Description
315	Interpolation table incorrect

9. Combining Two TAB1 Functions

There will be many places in an ENDF/B processing code where two functions expressed as TAB1 records will be combined to form a third TAB1 record. If A and B denote the original TAB1 records, and C denotes the TAB1 record after combination, then we denote the operation by

$$y_A(x) \;\theta\; y_B(x) = y_C(x)$$

The symbol θ denotes the specific operation and may be addition, subtraction, multiplication, division, interpolation, etc. The general method is to write function subroutines to perform each of the specific operations, use these function subroutines in a subroutine (CØMBP) to perform the combination in a single panel, and finally contruct a subroutine (CØMB) to break the original TAB1 functions into single panels.

9.1 Subroutine CØMBP

Consider a single panel whose end points are X1 and X2. Let YA1 and YA2 be the values of function A at the end points, and YB1 and YB2 be the values of function B at the end points. Then YC1 = YA1 θ YB1, and YC2 = YA2 θ YB2. Intermediate values in the panel can be obtained by interpolation using codes IA and IB for functions A and B. Thus, the function $y_C(x)$ is completely specified in the panel and the problem is to define a set of x values in the panel and interpolation codes such that the resulting tabulation of $y_C(x)$ is accurate to a relative error EPS. The method proceeds as follows:

Compute an X3 = 0.5*(X1+X2), and compute YA3 and YB3 at X3 by interpolation. Then YC3 = YA3 θ YB3 is the "exact value" at X3. We want to determine if this value can be predicted to within a relative error EPS by interpolating between YC1 and YC2. If we can, then we assume that all intermediate values can be interpolated within an accuracy of EPS and the TAB1 function need only contain the values YC1 and YC2. If we cannot, intermediate values are needed, we move X2 to X3 and repeat the procedure until a satisfactory X2 is found. This value is accepted and becomes the X1 for the next pass. X3 is reset to its original value, and the process continues until the entire original panel is completed. Note that X1 refers to the last point accepted, X2 refers to the next point under consideration for acceptance, and X3 = 0.5*(X1+X2) refers to the intermediate point used to determine whether X2 is acceptable or not.

We next describe the criterion for accepting a value at X2. Again we have the exact values YC1, YC2, and YC3 at the points X1, X2, and X3. We first test to see if YC1 or YC2 is less than or equal to zero, and if one of them is, we restrict the interpolation code to be linear. Let YC4 denote the interpolated value, and EPS4 the relative error.

YC4 = 0.5*(YC1+YC2)

$$EPS4 = 3\ |YC4-YC3|\ /(|YC1| + |YC2| + |YC3|)$$

This definition of EPS4 avoids problems when YC3 is near zero.

If YC1 and YC2 are positive, but X1 or X2 is negative or zero, we again restrict the interpolation to being linear and compute

YC4 = 0.5*(YC1+YC2)

$$EPS4 = |YC4-YC3|/YC3$$

If YC1, YC2, X1, and X2 are all positive, we try the interpolation code (IC) used for the last point accepted (in an attempt to keep the interpolation tables NBT and JNT to a reasonable size). The interpolated value YC4 is obtained from subroutine TERP1.

CALL TERP1(X1,YC1,X2,YC2,X3,YC4,IC)

$$EPS4 = |YC4-YC3|/YC3$$

If the EPS4 computed from one of the three methods given above is less than or equal to the input criterion EPS, the point (X2, YC2) is accepted and stored in /RECS/ along with the interpolation code which was used.

If the EPS4 computed using the last interpolation code used is not satisfactory, we try the remaining allowed interpolation codes. Thus, a value YC4 and relative error EPS4 is computed for I=2, 3, 4, and 5, and the one giving the smallest EPS4 is selected. If this EPS4 is less than or equal to EPS, the point (X2, YC2) is accepted along with the interpolation code corresponding to the smallest EPS4.

When a value is accepted, it is stored in the TAB1 area in /RECS/. Let X(N), Y(N) denote the accepted point which has just been stored. We would like to keep the total number of points to a minimum so that we will not overflow storage. If the interpolation code used between X(N-2) and X(N-1) is the same as used between X(N-1) and X(N), it may be possible to interpolate the point at X(N-1) to the required accuracy. A value YC6 is computed at X(N-1) by interpolating between Y(N-2) and Y(N). If YC6 agrees with Y(N-1) to within an error EPS, the point at X(N-1) is eliminated and the tables compacted.

A discontinuity (X1=X2) is sensed and treated properly.

The methods described above are contained in subroutine CØMBP whose calling sequence is:

CALL COMBP(X1,YA1,YB1,X2,YA2,YB2,IA,IB,ØPER,CØN,EPS,LØF)

where ØPER is the name of a function subroutine to perform the required operation, and CØN(N) is a list of constants which may be required by ØPER.

This function subroutine is called with a statement

YC = ØPER(YA, YB, CØN)

LØF is an overflow indicator (normally 0) which is turned on if the tables in /RECS/ overflow.

9.2 Subroutine CØMB

Two TAB1 records can be combined to form a third TAB1 record with subroutine CØMB. The original two TAB1 records must be in /DENS/, the resulting TAB1 record will be in /RECS/. The calling sequence is:

CALL CØMB(ØPER, CØN, MA, MB, XL, XH, EPS, LØF, LNT)

ØPER is the name of a function subroutine to perform the specific desired operation, and may use a set of constants CØN(N). MA and MB are the identification numbers of functions A and B in /DENS/. XL and XH are the lower and upper limits of the range of X over which the combination is to be performed. EPS is the relative accuracy desired in resulting TAB1 function. LØF is an indicator (normally 0) which is turned on if the resulting TAB1 function overflows /RECS/. LNT is an indicator (normally 0) which is turned on if either record MA or MB is not in /DENS/.

CØMB operates in the following manner. First, storage areas in /RECS/ are initialized and records MA and MB are located in /DENS/. The union of the X meshes for records MA and MB is formed and the resulting X mesh defines the panels required by subroutine CØMBP. CØMB uses subroutines LRIDS, FPDS, and IPDS to obtain points from /DENS/ for records MA and MB.

9.3 Function Subroutines Used by CØMB

A function will be called by CØMB (actually by CØMBP) with the statement

YC = ØPER (YA, YB, CØN)

where ØPER is the name of a function subroutine, and CØN(N) is a list of constants which may be required. The following function subroutines have been written.

ØPER	Description of Operation
ADD	YC = YA + YB. No constants needed
SUB	YC = YA - YB. " " "

ØPER	Description of Operation
MULT	YC = YA * YB. No constants needed
DIV	YC = YA/YB " " "
TERP	YC is obtained by interpolating between YA and YB. ZA, ZB, ZC are the Z values associated with YA, YB, YC (y=y(z)), and IZ is the interpolation code. These values are contained in CØN(N).
	ZA = CØN(1) ZC = CØN(3)
	ZB = CØN(2) IC = CØN(4)

More exotic combinations may be added as needed. No limit is imposed on the number of constants used.

9.4 An Example

As an example of the use of CØMB, consider the problem of determining the average cross section

$$\bar{\sigma} = \int_{EL}^{EH} dE\,\sigma(E)\,M(E) / \int_{EL}^{EH} dE\,M(E)$$

where M(E) is the Maxwellian distribution at T=0.0253ev. Assume that the TAB1 record for σ(E) has been read in and stored in /DENS/ with MA = 169. The following statements would be used to compute the average value, SIGB, to an accuracy of 0.1 %.

```
EPS = 0.001
CALL GENT1(MAXW, 0.0253,EL, EH, EPS, LØF)
CALL GRATE(EL, EH, ANS1)
CALL STØRE(3, 16, LØF)
CALL CØMB(MULT, 0.0, 169, 16,EL,EH,EPS, LØF, LNT)
CALL GRATE(EL, EH, ANS2)
SIGB = ANS2/ANS1
```

The first statement defines EPS. The next statement generates a TAB1 record in /RECS/ for a Maxwellian at T=0.0253ev for $EL \leq E \leq EH$. The next statement integrates the Maxwellian and stores the integral as ANS1. The Maxwellian is then stored in /DENS/ and given an identification number of 16. The cross section (ident 169) and the Maxwellian (ident 16) are then multiplied (function MULT, no constants needed) and the resulting TAB1 record stored in /RECS/. The product is integrated and the integral stored in ANS2. Finally, the answer is computed as the ratio of ANS2 and ANS1.

For simplicity, the flags LØF and LNT were not tested. In an actual program these flags should be tested. If LØF is turned on by GENT1 or CØMB, the EPS may have to be made larger. If LØF is turned on by STØRE, some records in /DENS/ will have to be moved to bulk storage.

DAMMET

A Program to Delete, Alter Mode, and Merge ENDF/B Tapes

by

Henry C. Honeck
U.S. Atomic Energy Commission
Washington, D. C.

and

Joan Félberbaum
Brookhaven National Laboratory
Upton, New York

March 1967
Revised April 1969

Contents

1. Introduction

DAMMET is a service routine to Delete, Alter Mode, and Merge ENDF/B Tapes. It is written in Fortran IV for a computer with 20K of available storage. The code requires three tapes plus the system input, output, and scratch (2 required) tapes.

DAMMET does not do any computation or processing of the ENDF/B data. It simply transfers data from tape to tape. The logic is simple and straightforward. As a consequence, certain operations may be highly inefficient and involve excessive tape handling time. If available, discs should be used in place of the tapes, particularly the input tapes.

2. General Operation of the Code

The following sections describe the overall operation of the code, input data, and error stops.

2.1 Operation of the Code

An ENDF/B tape may exist in one of the following four modes:

Mode	Description
1	Binary tape, standard arrangement
2	Binary tape, alternate arrangement
3	BCD card image tape, standard arrangement
4	BCD tape for printing, standard arrangement

The DAMMET code accepts an ENDF/B tape in modes 1-3 and will convert it to another tape in modes 1-4. Sections of the data can be deleted during this operation. In addition, two tapes can be merged to form a third tape. Two methods of operation are possible.

1. Mode Alteration Only

Input tape NTA in mode MØDEA will be read, unwanted sections deleted, and written on output tape NTC in mode MØDEC.

2. Mode Alteration and Merge

Input tape NTA in mode MØDEA and tape NTB in mode MØDEB are to be merged to form tape NTC in mode MØDEC. If the arrangement specified by MØDEA is not the same as that specified by MØDEC, tape NTA is transferred to a scratch tape NTSA with a mode alteration from MØDEA to MØDEC. A similar operation is done if MØDEB and MØDEC do not correspond to the same arrangement. The resulting tapes NTA (or NTSA) and NTB (or NTSB) are then merged onto NTC.

2.2 Tape Assignments

The following tape assignments are made at the beginning of the DAMMET main program.

Symbol	Value	Usage
NIN	5	System BCD input
NØUT	6	System print output
NTSA	3	Scratch
NTSB	4	Scratch

All other tapes are assigned on input cards.

2.3 Description of the Input Data Cards

Item	Cols.	Format	Symbol	Description
1	1-11	I11	NTA	First (tape A) input tape number
	12-22	I11	MØDEA	Mode (1, 2, 3) of tape A
	23-33	I11	NTPIDA	Ident. of tape A. If zero, ident. will not be checked.
	34-44	I11	NIDA	Number of deletion cards ($\leq$100) of type 2 for tape A.

2 Repeat the following items for tape A until NIDA cards have been given If NIDA=0, omit this item.

Item	Cols.	Format	Symbol	Description
	1-11	I11	MATDT	Material number of section to be deleted.
	12-22	I11	MFDT	File number of section to be deleted.
	23-33	I11	MTDT	Section number of section to be deleted.

If a zero appears in any of these fields, it is read as "all". Thus, 0, MF, 0 means delete all sections in file MF for all materials.

Item	Cols.	Format	Symbol	Description
3	Repeat item 1 above for tape B (i. e., NTB, MODEB, etc.) If no merge is to be done and tape B not needed, set NTB=0.			
4	Repeat item 3 above for tape B giving NIDB cards.			
5	1-11	I11	NTC	Output (tape C) tape number
	12-22	I11	MØDEC	Mode (1, 2, 3, 4) of tape C.
6	1-66	16A4, A2	(BP(N), N=1, 17)	66 character tape label to be used on NTC.
	67-70	I4	BP(18)	Tape ident. to be used on NTC.

2.4 Printed Output

The printed output consists simply of a listing of the input data and the words "Run Completed".

2.5 Error Stops

If an error occurs, the code exits with a statement

CALL ERRØR(xxx)

where xxx is an integer. The ERRØR subroutine prints the message "ERRØR STØP xxx, MAT=xxxx, MF=xx, MT=xxx". The following error stops are used. Consult individual subroutines for detailed explanation.

Stop Numbers	Subroutine
99-107	RREC
108-109	WREC
133-135	TERP1
200-205	TSEC
206-207	MTAPE
208-211	DAMMET (main program)

3. Detailed Description of the Code

The following sections describe in some detail the construction and operation of DAMMET.

3.1 Reading, Writing, and Transferring Records

The basic unit in the ENDF/B system is a record of which there are six types.

JT = 1,	CØNT	JT = 4,	TAB2
= 2,	LIST	= 5,	HØLL
= 3,	TAB1	= 6,	TPID

A record can be read from tape NT in mode MØDE and stored in CØMMØN/RECS/ with the statement

CALL RREC(JT,NT,MØDE,T)

where if $T \geq 0$, and the record is followed by other records giving the temperature dependence, all of these records will be read and interpolated at the desired temperature T. If $T < 0$, only the first record will be read.

A record in /RECS/ can be written on tape NT in mode MØDE with the statement

CALL WREC(JT,NT,MØDE)

These subroutines and CØMMØN/RECS/ are fully described in a document "Retrieval Subroutines for the ENDF/B System," which is included with this Report (ENDF-110).

A record can be transferred from tape NT1 in mode MØDE1 to tape NT2 in MØDE2 using these two subroutines. The only complication is that a record may be temperature dependent and followed by LIST records containing data at higher temperatures. These are considered to be an extension of the original record and are transferred along with it. The subroutine TREC is used to accomplish these transfer. The calling sequence is

CALL TREC(JT)

and the numbers NT1, MØDE1, NT2, MØDE2 are transmitted to the subroutine through CØMMØN/IØTM/ (Input Output Tapes and Modes).

An index should be specified in File 1 to describe what sections are present. (If this index is not specified, it will be added by DAMMET.) This index will be contained in the first section (MF=1, MT=451) following the comment cards. The structure of this section is changed as follows:

(MAT,1,451/ZA,AWR;LRP	,LFI	;0	,NXC	)HEAD
(MAT,1,451/b , b ;LDD	,LFP	;NWD	,b/H_n	)HØLL
(MAT,1,451/b , b ;MF_1	,MT_1	;NC_1	,b	)CØNT
(MAT,1,451/b , b ;MF_2	,MF_2	;NC_2	,b	)CØNT

(MAT,1,451/b , b ;MF_{NXC}	,MT_{NXC}	;NC_{NXC}	,b	)CØNT
(MAT,1, 0/0.0,0.0; 0	, 0	;	0,0	)SEND

The index is given on the CØNT cards and there are NXC of them. The number NXC is punched in the last field of the HEAD card and may be zero. One CØNT card is given for each section present in this material (including MF=1, MT=451). The appropriate MF, MT numbers are punched in the third and fourth fields of the CØNT card. The CØNT cards are ordered in the same way as are the sections, that is, first by increasing values of MT for a given MF, and then by increasing MF.

The fifth field on the CØNT card (NC) is a count of the number of cards in the section excluding the SEND card. The NC then count only data cards.

The sixth field on the CØNT card is left blank at the time data is prepared. The field is reserved for later use by processing codes to binary records (or numbers) associated with the section.

3.2 Transferring Sections

A section on an ENDF/B tape is composed of a Head record of the CØNT type, one or more intermediate records, and a section end (SEND) record. A section can be transferred from tape NT1 in mode MØDE1 to tape NT2 in mode MØDE2 with the statement

CALL TSEC(NT1,MØDE1,NT2,MØDE2)

On entry to TSEC, the Head record must have been read (but not written) from NT1 in CØMMØN/RECS/. The numbers MAT, MF, MT are then available to TSEC to determine the number and type of the intermediate records. The Head record is written on NT2 and the intermediate records and the SEND record are transferred from NT1 to NT2.

3.3 Deletion Tables

During a tape transfer or merge operation, sections not wanted may be deleted. These sections are specified by the following arrays in CØMMØN/DELT/.

MATDT(NI,IDT)	-	Material number (MAT) to be deleted.
MFDT(NI,IDT)	-	File number (MF) to be deleted.
MTDT(NI,IDT)	-	Section number (MT) to be deleted.
NIDT(IDT)	-	Number of entries (NID) in above tables.
NI	-	Index to particular combination of (MAT, MF,MT) to be deleted.
IDT	-	Index to appropriate deletion table.
NIDX	=	100, maximum value of NI.
IDXX	=	2, maximum value of IDT.

These tables are loaded from input cards (see input data description). If two input tapes are specified (NTA and NTB), then IDT=1 will refer to deletions to be made from tape NTA, and IDT=2 to tape NTB.

A deletion request is the set (MAT,MF,MT) and says delete the section characterized by these numbers. A zero value is read as "all." Thus, (MAT,0,0) means delete all sections and all files of material MAT. (0,MF,0) means delete all sections in file MF for all materials.

When the Head record of a section has been read into /RECS/, the following statement can be used to decide whether the section is to be transferred or deleted.

CALL DELETE(IDT,JD)

IDT (=1 or 2) indicates what table is to be used. The number of entries in that table is NID = NIDT(IDT). The subroutine then compares the (MAT,MF,MT) in /RECS/ for the section in question with the MATDT, MFDT, and MTDT tables. If no correspondence is found, JD is set to zero. If a correspondence is found, JD is set to unity.

3.4 Transferring Data from Tape to Tape

We can now think of an ENDF/B tape as composed of sections (which can be transferred with TSEC) and control cards of the TPID, FEND, MEND, and TEND types. If the output tape NT2 is to be arranged in the same way as the input tape NT1, it is then a simple matter to transfer sections from one tape to the other, and only one pass through tape NT1 is required. Unwanted sections can be deleted during this transfer operation.

If the output tape NT2 is to be arranged differently than the input tape NT1, the transfer is more complicated. For example, suppose MØDE1 = 3 (BCD, standard arrangement) and MØDE2 = 2 (Binary, alternate arrangement). We must first read the entire tape NT1 selecting and writing on NT2 all of the Files of Type 1. Tape NT1 must be rewound and completely read again to pick up all of the File 2, etc. In general, it may be necessary to read tape NT1 as many as 7 times. Obviously, considerable time can be saved by initially transferring tape NT1 to a disc.

The other complicated case is where, for example, MØDE1 = 2 and MØDE2 = 1 (or 3, or 4). Here the tape NT1 must be completely read once for each material present of NT1.

The methods used here are straightforward and admittedly inefficient. Hopefully, these transfers and rearrangements of data will not need to be done often, so that the inefficiency will not be too great a penalty. If these operations begin to consume excessive amounts of computer time, a new code will have to be written.

It should be noted that during transfer from tape to tape, the appropriate control cards and sequence numbers (for MØDE2 = 3) are automatically included on the output tape.

Data can be transferred from tape NT1 in mode MØDE1 to tape NT2 in MØDE2 with the statement

CALL TTAPE(NT1,MØDE1,NT2,MØDE2,IDT,BP,LTC)

where IDT is the index (1 or 2) to the appropriate deletion table (see previous section), LTC is a label transfer control, and BP(N) is a tape ident array. If LTC=0, the TPID record on NT1 is transferred to NT2. If LTC=1, the TPID record on NT1 is ignored, and a new TPID record is constructed from BP(N) and written on NT2.

3.5 Merging Tapes

The merging of two ENDF/B tapes is a simple operation if both tapes are in the same arrangement. Where necessary, subroutine TTAPE is used to put both input tapes into the arrangement specified for the output tape.

The merge proceeds as follows. The two tapes to be merged (NTA and NTB) are positioned behind the TPID record. Output tape NTC is rewound and a TPID record (originally read from input cards) written. A Head record is then read from both NTA and NTB. The corresponding (MAT,MF,MT) are compared and the one with the lower values selected, the complete section transferred to NTC, and another Head record read. This process is repeated until all sections have been transferred (or deleted). Appropriate control cards are automatically inserted, and if MØDEC=3, sequence numbers added.

The merge operation is performed with the statement

CALL MTAPE(NTA,MØDEA,IDTA,NTB,MØDEB,IDTB,NTC,MØDEC,BP),

where NTA and NTB are the tapes to be merged, MØDEA and MØDEB are their modes, IDTA and IDTB are the indices to their deletion tables (zero if not used), NTC and MØDEC are the output tape and mode, and BP(N) is the TPID record to be written on NTC.

3.6 The Control Program

Having written subroutines to perform the required transferring and merging operations, it is a simple matter to write the control program, DAMMET. DAMMET performs the following steps:

1. Read input data and initialize.
2. If no merging is to be done (NTB=0), tape NTA is transferred to NTB using subroutine TTAPE.

(con't on next page)