KfK 2387/VI
März 1978

The KEDAK Program Compendium Part VI Mechanized Transfer of Nuclear Data from ENDF/B to KEDAK and vice versa

Compiler: E. Stein
Institut für Neutronenphysik und Reaktortechnik
Projekt Schneller Brüter

Kernforschungszentrum Karlsruhe

KERNFORSCHUNGSZENTRUM KARLSRUHE GMBH

KERNFORSCHUNGSZENTRUM KARLSRUHE

Institut für Neutronenphysik und Reaktortechnik
Projekt Schneller Brüter

KfK 2387/VI

The KEDAK Program Compendium

Part VI

Mechanized Transfer of Nuclear Data from ENDF/B to KEDAK and vice versa

Compiled by

E. Stein

with contributions from
I. Langner, R. Meyer*, G.C. Panini**, E. Stein

* Present address: Software AG, Darmstadt
** C.N.E.N., C.D.C., Bologna

Kernforschungszentrum Karlsruhe GmbH., Karlsruhe

Abstract

Translation of nuclear data from one representation into another one permit programs, which are linked to a specified data base, to use also data out of another data base.

The programs BRIGITTE and COPEND, used for translating from ENDF/B to KEDAK, will be described in detail. The description includes source program listings. Translation from KEDAK to ENDF/B can be done with the program KTOE. The report contains a copy of the NEA (Nuclear Energy Agency) CPL (Computer Programme Library) abstract. The programs BRIGITTE (abstract no. NEA 438) and KTOE (abstract no. NEA 342) can be obtained from the NEA CPL.

Das KEDAK Programm Compendium
Teil VI

Mechanisierte Übertragung von Nuklearen Daten von ENDF/B nach KEDAK und umgekehrt

Zusammenfassung

Die Übertragung von nuklearen Daten aus einer Darstellung in eine andere erlaubt es in Programmen, die an eine Datenbasis angeschlossen sind, auch Daten aus einer anderen Datenbasis zu verwenden.

Die zur Übertragung von ENDF/B nach KEDAK verwendeten Programme BRIGITTE und COPEND werden ausführlich beschrieben. Die Beschreibung enthält auch Quellprogrammlisten.Die Übertragung von KEDAK nach ENDF/B kann mit dem Programm KTOE erfolgen. Der Bericht enthält eine Kopie der NEA (Nuclear Energy Agency) CPL (Computer Programme Library) Kurzbeschreibung. Die Programme BRIGITTE (abstract no. NEA 438) und KTOE (abstract no. NEA 342) können von der NEA CPL bezogen werden.

Acknowledgement

The authors of this report are grateful to Mrs. K. Mayer for typing this report with great carefulness and patience. Besides typing Mrs. K. Mayer established reproducable graphical representations out of sketches of flowcharts for programs.

The authors are also indebted to Dr. H. Küsters for his critical reading of this report.

Contents

Zum Druck am: Jan. 1978

Section 1.

Some remarks about the exchange of nuclear data

E. Stein

Normally the representation of nuclear data,used to exchange these data,is different from that one used to work with these data. This must be done, because using the exchange data set, which is in nearly all cases sequentially written on a magnetic tape, is ineffective, if a lot of random accesses are made. It is evident, that the translation from the exchange representation of the data to the local library and vice versa, should be rather simple. E.g. the translation program must not be changed, if new data types are defined.

Unfortunately the described state is not at all realized. There are existing - due to historical reasons - different nuclear neutron data libraries. The data of these libraries can only be exchanged easily, if the receiving library is of the same type. The difficulties result from the different conventions used when storing the same physical data. One solution of this problem could be to use world wide one set of conventions when representing the same data, that means to use one library. But this can be done only with a very great effort of work, because all programs, which are working with the data of a specified library, are using the special conventions of this library. That means, that all these programs then had to be changed.

A second solution of the problem is to translate the data from one representation into another and vice versa. This does not only mean to represent the data in such a manner that they can be read by the retrieval routines of the other library, but also to use all conventions of the other library in the translation of the data.

So far we have made only general considerations about the problem. In the following we will restrict to the exchange of data between the libraries ENDF/B (see /2/ and /3/) and KEDAK (see /4/, /5/, and part II of this KEDAK Program Compendium). Because the data in KEDAK are mainly intended to be used for the calculation of fast reactors, only a subset of the ENDF/B data types must be exchanged. To enable the programs, which are working with KEDAK, an access to this subset of ENDF/B-data, the program BRIGITTE - described in section 2. - has been written. A change of all the programs would have been much more difficulty.

If - in the future - KEDAK users will be interested in datatypes, which are not processed by BRIGITTE until now, the data can be used directly from

ENDF/B, or the program BRIGITTE has to be extended. An extension of the program BRIGITTE should be made, if the existing programs can work with the translated data after performing only minor program changes. If processing of the new data types requires new programs or extensive program changes, the data should be directly read from ENDF/B. In this case the data should not be read from the card image form of ENDF/B, but from a dataset to be created on a direct access device, which allows random access, without sequential processing.

Until now only one way - ENDF/B to KEDAK - of data exchange has been treated. The tools for the other direction are described in section 4. Until now, we have not done any translation from KEDAK to ENDF/B. But it is planned in the near future to offer KEDAK materials, which have been evaluated in Karlsruhe, also in an ENDF-representation. This will be done by using the program, which is described in section 4., or, if this is not possible, by changing this program, or by developing own methods.

But it should be kept in mind, that a card image representation of the KEDAK library exists. This card image representation is used to exchange the data between KEDAK libraries on different installations. But it can also be used to print the data, or if retrieval programs exist[+)], to process the data. However, due to the sequential representation, this is not an optimal way.

[+)] A retrieval program package for the KEDAK card image representation, which will have the same subroutine calls as the LDFPAC (see part III), will be developed in the near future.

This page has been left internally blank.

Section 2.

BRIGITTE

A program for translating nuclear data in ENDF/B-representation to nuclear data in KEDAK-representation

J.C.Schepers[+], E. Stein, P.Vandeplas[++]
described by
E. Stein

[+] Formerly employed by CEN MOL

[++] Working at CEN MOL

Table of contents

page

2.1 Introduction

The program BRIGITTE translates a selection (see 2.2.2) of ENDF/B (see /2/ and /3/) data types into the KEDAK-representation (see /4/, /5/ and II.1 of this compendium). The necessity to do this has been explained in section 1. The present version of the program solves the problem in a nearly automatic way. But because the translated data must be controlled and in few cases changes must be made, we prefer to use the term "mechanized transfer" instead of automatic transfer.

Several people have developed the program BRIGITTE. The first version was written by J.C. Schepers (see /1/) from CEN at MOL (Belgium). The present running version of the program at GfK in Karlsruhe has been developed out of the program witten by J.C. Schepers by making corrections, improvements and extensions. This work has been influenced by some important proposals and ideas from P. Vandeplas, from the CEN at MOL. P. Vandeplas improved some parts of the program in the version of BRIGITTE, running at MOL.

This report describes the Karlsruhe version of BRIGITTE only. To understand this report, the user should be familiar to some extent with ENDF/B and KEDAK. For ENDF/B, one may read the reports /2/ or /3/; for KEDAK, the preceding parts of this Compendium (mainly II.1 and sections of III) or the reports /4/ and /5/ can be used.

The following information about the program BRIGITTE starts with a guide for the program user. The recommendations given in 2.2 should be obeyed, when reading this section. The user's guide is followed by a programmer's guide (2.3), where one can find all information, which is necessary for program changes or for the location of program errors. After this section one can find the reference listings (2.4) followed by the source program. Section 2.5 contains a list of the source programs together with some aids to find one's way in this listing.

At the end of this introduction, we would like to point to section 2.2.4, where the normal way of inserting data into a public KEDAK library is described. This section will explain the necessity of the program COPEND, which is described in 3.

2.2 User's guide

If one likes to run the program BRIGITTE, one may read the following sections until 2.2.10 and additionally 2.3.1 and 2.3.3, and then start the program. But if one wants to have a BRIGITTE output at once and are only reading 2.2.10, we must give a warning for using a KEDAK-library produced by or processed with KEMA (see part II of this Compendium or /4/) as direct acceptor (data-set linked to FT01F001) for the converted data. Before running the program, one should in any case read 2.2.4.

2.2.1 Differences between the ENDF and the KEDAK representation

In this section the expense of work, which must be done by the program BRIGITTE, will be shown. This work results from the differences between the ENDF and the KEDAK representations of the same data.

(1) ENDF-conventions (see /2/ and /3/)

(α) ENDF is built up of mainly four record types: CONT, LIST, TAB1, and TAB2. To represent a specified cross section type these records are composed in a defined manner. But this composition can depend from the value of a quantity in a first record. The type and sequence of records can differ from datatype to datatype and must be given for all datatypes in a written description.

The data in ENDF can be retrieved in units of records. A TAB1-record can contain e.g. the pointwise representation of the total cross section in the whole energy range.

(β) ENDF uses five interpolation schemes. By this the number of stored data can be reduced, but problems occur when combining the ENDF-data (e.g. addition of two pointwise cross sections).

(γ) ENDF did not contain compound cross sections like the transport cross section .

(δ) Often a parametric representation is chosen in ENDF instead of a pointwise representation; e.g. in the resonance region only parameters plus a correction cross section is given.

(2) KEDAK conventions (see /4/, /5/ and part II of this Compendium):

(α) KEDAK is built up of data items. A data item can have N arguments ($N \geq 0$) and M functional values ($M \geq 0$). An example for a data item is with N = 1 and M = 1 an energy value together with a cross section value, or with N = 3 and M = 8 an item describing a resolved resonance. The data in KEDAK are retrieved in units of data items. A number of data items is composed to a data type, e.g. a cross section in the whole energy range or a set of all resonance information for a specified material. This datatype can have one or more so called "further names". In the case of the inelastic excitation cross sections,the excitation energy is a further name. To each further name or each combination of further names a set of data items, which have all the same number N of arguments and the same number M of functional values, is associated.

(β) KEDAK uses only linear interpolation.

(γ) Programs, which are using KEDAK, expect to find a number of compound cross sections, like the transport or the absorption cross section.

(δ) Some cross sections must not be given only in parametric form on KEDAK, but also in a pointwise representation, because users of KEDAK and above all, programs working with KEDAK are expecting these data.

By (1)(α) and (2)(α) we can see that each datatype of ENDF with a different structure must be processed with a different method. But this is not only a problem of the program BRIGITTE but also a problem in a normal retrieval of data from ENDF/B. Points (1)(β) and (2)(β) shows that converting from

ENDF/B to KEDAK means the introduction of only linear interpolation. This can be done by generating enough points between the existing ones.

Due to (1)(γ) and (2)(γ) the compound cross sections must be produced in BRIGITTE. This is only done for the transport and the absorption cross section.

The differences mentioned in (1)(δ) and (2)(δ) give rise, in the case of the resonance parameters, to a lot of computer work, because the calculation of pointwise cross sections out of resolved resonance parameters is very CPU-time consuming in the case of many resonances.

2.2.2 Datatypes converted by the program

In the following table all datatypes, that can be converted by the version of the program described here, are listed. The ENDF identification (see /2/,/3/) is followed by the associated KEDAK alphamerical name (see /4/, /5/, II). Some data types, which are given in ENDF/B by different formulations, as in the case of energy distributions of secondary neutrons, can only be converted for some formulations. This is also mentioned in the following table. If the ENDF MT-number is enclosed in "(", that datatype is no longer used in ENDF/B. That means this datatpye must be generated during conversion. A "-" sign in the MT-column means: this KEDAK-datatype did not exist in ENDF/B. Remember: In the case of SGT,SGN,SGF and SGG the resolved and unresolved resonance part - if existant - is added to the appropriate pointwise cross sections of ENDF/B MF = 3.

ENDF MF	ENDF MT	KEDAK alphamerical name	remarks
1	452	NUE	average total number of neutrons per fission (If given in a polynomial representation, the pointwise representation is calculated in conversion.)
2	151	RANGRES	see /4/, /5/, II
		ISOT1	" "
		ISOT2	" "

ENDF MF	ENDF MT	KEDAK alphamerical name	remarks
2	151	ISOT3	generated in the case of more than 1 isotope. See /4/, /5/, II.
		RES	resolved resonance parameter set
		ST	unresolved resonance parameter set
		STGF	generated in the case of fissionable materials. Contents see /4/, /5/, II
			pointwise cross sections are produced out of the resonance parameters and added to the appropriate data of ENDF MF = 3. Conversion and generation of pointwise cross sections is only done for single and multilevel Breit-Wigner parameter sets
3	1	SGT	total cross section
3	2	SGN	elastic scattering cross section
3	3	SGX	non elastic cross section [1]
3	4	SGI	total inelastic cross section
3	16	SG2N	(n,2n) cross section
3	17	SG3N	(n,3n) cross section
3	18	SGF	total fission cross section
3	22	SGIA	(n,n'α) cross section
3	23	SGI3A	(n,n'3α) cross section
3	24	SG2NA	(n,2nα) cross section
3	25	SG3NA	(n,3nα) cross section
3	(27)	SGA	absorption cross section. Produced in the program. SGA = SGG + SGF + SGP + SGD + SGH3 + SGHE3 + SGALP + SG2HE
3	28	SGIP	(n,n'p) cross section
3	29	SGI2A	(n,n'2α) cross section

[1] The translated cross section SGX does not contain the resonance parts.

ENDF MF	ENDF MT	KEDAK alphamerical name	remarks
3	51 . . 90	SGIZ	inelastic cross section for excitation of rest nucleus. (If MT = 51 did not exist, but higher MT-numbers can be found, see 2.2.6, namelist INPUT, variable SALINE.)
3	91	SGIZC	inelastic scattering cross section to the continuum
3	102	SGG	(n,γ) radiative capture cross section
3	103	SGP	(n,p) cross section
3	104	SGD	(n,d) cross section
3	105	SGH3	(n,t) cross section
3	106	SGHE3	(n,He^3) cross section
3	107	SGALP	(n,α) cross section
3	108	SG2HE	(n,2α) cross section
3	-	SGTR	transport cross section SGTR = SGT - MUEL * SGN Always calculated in conversion (if MUEL exists)
3	-	ETA	average number of fission neutrons per neutron absorption. (Not produced in the running version of the program but foreseen in later versions.)
3	-	ALPHA	ratio of capture (SGG) to fission (SGF) cross section. (Foreseen to be produced in later program versions.)
3	251	MUEL	average cosine of the elastic scattering angle in the laboratory system
4	2	SGNC / SGNL / LEGNG / LEGNL	angular distributions of secondary neutrons for elastic scattering. SGNC and SGNL are tabulated distributions in the center of mass resp. laboratory system. LEGNC and LEGNL are sets of Legendre coefficients. If the ENDF/B data are given in tabulated form, only SGNC resp. SGNL is generated. In the case of Legendre coefficients, the tabulated data are generated.
5	16	SED2N	energy distributions of secondary neutrons emitted by the (n,2n) process
5	17	SED3N	energy distributions of secondary neutrons emitted by the (n,3n) process

ENDF MF	ENDF MT	KEDAK alphamerical name	remarks
5	18	SEDF	energy distributions of secondary neutrons emitted in the fission process
5	91	SEDIC	energy distributions of secondary neutrons emitted in the inelastic scattering to the continuum
			Only the following ENDF energy distribution laws can now be converted: LF = 3 in K = 4 (excitation of discrete levels) LF = 7 in K = 2 (Maxwellian spectrum) LF = 9 in K = 1 (evaporation spectrum) LF = 10 in K = 2 (Watt spectrum)

2.2.3 General description of the program (options, region, time, restrictions)

As described in the preceding sections, the program BRIGITTE translates nuclear data from the ENDF- into the KEDAK-representation. Only a selection of the data types is processed (see 2.2.2).

Provided that a load module of the program exists, one can use the JCL-cards, described in 2.2.9, and the input control cards, described in 2.2.6, to start the program. The construction of a complete job will be much simplified, when using the examples given in 2.2.10.

An exact description of the program flow can be found in 2.3.1. For this section it is sufficient to know, that the translation of the ENDF/B data is made filewise. First some data of MF = 1 are processed and stored in an auxiliary file, then the resonance data (MF = 2) are read, stored and used for the calculation of pointwise cross sections, which are also stored in an auxiliary file. These pointwise data are then combined with the MF = 3 data. The result of this process is then adapted to KEDAK conventions and stored in an auxiliary file. After this some data of MF = 4 and afterwards MF = 5 are processed in a similar mode. In the last section of the program the data, stored on the auxiliary file, are included into a KEDAK-library*)

*) This KEDAK-library must not be created by or processed with KEMA (see 2.2.4)

There are some options in the program, which can be selected by using the quantities in the namelist control input. Processing of data of MF = 4 and/or MF = 5 can be suppressed, or processing only MF = 4 or MF = 5 data can be selected (see IF4 and IF5 in the namelist INPUT, 2.2.6). Only due to historical reasons the data translation can be made with respect to a not longer used KEDAK convention: all inelastic cross sections have the same energy mesh (see KEDAVE in the namelist INPUT, 2.2.6). Until now, only two options, controlled by quantities in the namelist INPUT, have described. To get a total knowledge of all options, one must read section 2.2.6.

Some important options are controlled by variables in the namelist LIB (also described in 2.2.6). One can select, whether the program has to built up a new KEDAK-library, or can use an existing one. It is also possible, to delete in an existing library a material, if it has the same name, as the converted one. But remember: all possible program options are described in 2.2.6.

The region used by the program itself is about 102 k-byte. Some region is needed by the system, for read-input and print-output buffers, and for buffers for other datasets, used by the program. As a mean value we can take about 150 k-byte necessary for the program, for system use, and for buffers. The rest of the region, which is given in the JOB-card, is used for data arrays. The size of the data field region mainly depends on the number of resolved resonance parameters. If there are no such or only a few parameters, 300 k-byte are normally sufficient to run the program. In the case of many resolved resonance parameters, 300 k-byte will not be enough. But we recommend, to start with 300 k-byte. If the program has not enough region, an error message will be produced. This error message will help to decide, how much additional region is necessary.

The CPU (central processing unit)-time needed by the program, also depends on the number of the resolved resonance parameters. This time will be given for an IBM/370 model 168. If there are no parameters, 1 minute is sufficient. For about 80 single level Breit Wigner resonances about 11 minutes are used. The CPU-times will increase up to hours, if there are many multilevel resolved Breit Wigner resonance parameters.

Although the data storage used in the program is variably dimensioned, there are some arrays of a fixed lenght, which causes some restrictions. These are:

- One (natural) element may have up to 10 isotopes.

- Five different l-states (l = 0,1,2,2,4) are allowed in the resolved Breit Wigner resonance parameter set.

- Three different l-states and 6 different j-states for each l-state are allowed in the unresolved Breit Wigner resonance parameter set.

- One hundred points are allowed as primary energy grid for energy distributions of secondary neutrons.

2.2.4 Insertion of the translated data into a local KEDAK library

The program BRIGITTE produces a KEDAK-library, which differs from the KEDAK-libraries produced by the program KEMA (see II, /4/ and /5/). The BRIGITTE created libraries contain in word 880 of the first record (the word is the last one) the number of the next available record in the library.

This word is used, when inserting new data in the subroutine SICKY (see 2.3.5.86). In the library created by BRIGITTE also the data of each material start on a new record. By this, deleting materials and compressing the library in the direct access form is much simpler than for the KEDAK-libraries created by KEMA (see /4/ resp. II). Because of the differences described here, it is not possible to use a KEDAK-library, which has been created by or processed with KEMA, as direct acceptor for the data converted by the program BRIGITTE. But there are no difficulties for the KEDAK retrieval codes, as described in III, to use the BRIGITTE created libraries. Because all programs, which are working with KEDAK, use these routines, all these programs can work with the "BRIGITTE-created" libraries.

In most cases using of a KEMA created (or processed) library will cause the program to do no insertion of the translated data. But because the present version of the program only tests word 880 of the first record, the decision whether the library is BRIGITTE- or KEMA-created (or processed) may be

incorrect. Therefore we recommend never to use a KEMA created (or processed) library as direct acceptor for the data translated by BRIGITTE.

The only way to insert the translated data in a public KEMA-created (or processed) library is to produce KEMA-input out of the BRIGITTE-created libraries. This problem is solved by the program COPEND, described in 3.

One may ask, why BRIGITTE itself does not produce this KEMA-input instead of a KEDAK-library. The answer is very simple: in many cases it is not necessary to insert the data into the public KEDAK-library. As stated before all retrieval routines, described in III, can work with the BRIGITTE-created libraries. Why then insert the data into one library? There are several reasons for this action:

(1) If the data should be kept for a longer time interval, it is much simpler to manage one library (e.g. produce back-up copies), than a number of different libraries.

(2) If a BRIGITTE-created library has been processed by KEMA (e.g. to correct some data), it can not be longer used by BRIGITTE.

(3) There are programs, which are processing several materials in one run. This materials must be contained in one library, because these programs use the retrieval program III.3, which permits only one KEDAK-library.

A last question may be left. Why BRIGITTE does not write directly in public KEMA-created KEDAK-libraries. There are three answers to this question:

(1) To have a control about changes in KEDAK, only one program should be allowed to update the library. This is the program KEMA.

(2) The program KEMA performs some formal checks before inserting the data.

(3) The program BRIGITTE can delete materials, with names identical to the translated ones, and compress the library after the deletion. This would be very difficult and expensive with KEMA-created (or processed) libraries.

The contents of this section can be summarized in the following statement. Never use a KEMA created (or processed) KEDAK-library in a BRIGITTE run as acceptor for the translated data. If the checks in BRIGITTE will fortuitously fail, the KEMA-created (or processed) library may be destroyed.

2.2.5 External data sets

The programm BRIGITTE uses 7(8) data sets represented by the following ddnames (see /10/):

FT06F001
FT05F001
FT01F001
FT09F001
FT10F001
FT11F001
FT12F001
(FT15F001) optional

The datasets, which the program expects to be allocated to this ddnames, will be described in the following.

FT06F001

Description:

Print output.

Actions taken:

The program makes only WRITE-statements on this dataset. This is done in the subroutines TITEL, PRIEIN, GSPACE, INITNL, INGRID, REGINE, MARY, MARY01, LUCY, RESERR, MILLI, MELA, ANNICK, ANNI01, DENISE, MARION, SOPHIE, SUZY, ADDPNT, SUPNEG, ISABEL, FILE5, PRIZEI, ERROR, PRCONT, PRLIST, PRTAB1, PRTAB2, PRHOL, PRTPID, WDA, WRECS, SICKY,LISA, TABPRT, PAGE and A8FORM.

DD-parameters:

The printer must be able to produce lines with a maximum of 132 characters.

Possible errors:

Maximal line length of the printer is smaller than 132 characters.

FT05F001

Description:

Input unit for (card) control input.

Actions taken:

The data are read from FT05F001 in the subroutines INITNL, PRIEIN and SICKY. A REWIND is made in PRIEIN.

DD-parameters:

The record format (RECFM) must be fixed (F) or fixed blocked (FB) with a logical record length (LRECL) of 80 bytes.

Possible errors:

The control input does not fulfil the conventions necessary for namelist input (see 2.2.6).

FT01F001

Description:

KEDAK-library * (created in the job or old one).

Actions taken:

The converted material is added to the KEDAK-library. If the library does not exist, it is created. If the library exists, READ and WRITE is made. If it is created, only WRITE-statements are executed.

DD-parameters:

A SPACE parameter must also be given for datasets with DISP=OLD, because the values in this parameter are used in the program in the subroutine DEFI (see 2.3.5.93). Because the BLKSIZE must be equal 3520, the parameter can look like SPACE=(3520,(600)).
The DCB parameter must be
DCB=(RECFM=F,BLKSIZE=3520,LRECL=3520,DSORG=DA).

Possible errors:

No SPACE parameter given for a dataset with DISP=OLD.
BLKSIZE not equal 3520.
Number of blocks not sufficient to receive the converted material (use another or a new KEDAK-library * where the material can be included).

---

* The KEDAK-library must not be created by or processed with KEMA; it must be created by BRIGITTE itself (see 2.2.4).

FT09F001

Description:

ENDF/B-data.

Actions taken:

The data are processed by READ-statements. No writing is done. First a REWIND is made in the subroutine INGRID and finally in the subroutine PART1. Reading is done - only in forward direction - in the subroutines SEARCH and PREC. BACKSPACE can be done in the subroutines REGINE, SOHPIE, ISABEL and FILE5.

DD-parameters:

Because normally ENDF/B is used in the MODE=3 representation (see /3/ chapter R-2.2) the record format (RECFM) must be fixed (F) or fixed blocked (FB) and the logical record length (LRECL) must be equal 80 bytes. The data must be in EBCDIC or BCD representation.

Possible errors:

Wrong tape used.

In the case of NL-types: incorrect DCB-specifications.

FT10F001

Description:

Auxiliary dataset, where temporarily ENDF/B-data are stored (and read) in the MODE=2 (see /3/ Chapter R-2.2).

Actions taken:

The data set is used in the subroutines SOPHIE (only if KEDAVE=0. See 2.2.6 namelist INPUT), ISABEL and FILE5. The data are written by using the subroutine WREC and reread by using the subroutine RREC.

DD-parameters:

Because writing is done, using no FORMAT-statements, the record format (RECFM) must be variable (VS) or variable blocked (VBS). The block size can be chosen with respect to an optimal buffer size. We recommend the values BLKSIZE=1016 or BLKSIZE=2040 or BLKSIZE=3064. Normally a SPACE of about 50 blocks of the size 1016 is sufficient. The dataset can reside on a disk, a tape, a drum, or something like that.

Possible errors:

Incorrect record format (RECFM).

Block size (BLKSIZE) too small when using RECFM=VS.

Space in the SPACE-parameter not sufficient.

FT11F001

Description:

Auxiliary dataset. Contains all data to be inserted in a KEDAK-library * by the subroutine SICKY. The structure of the dataset is described in the subroutine WDA (see 2.3.5.76).

Actions taken:

The data are written on the dataset by the subroutine WDA (see 2.3.5.76) and read by the subroutine RDAB (see 2.3.5.77). Only the subroutine ISABEL writes on FT11F001 without using these subroutines. To get the table of contents and addresses, the subroutines WDA and ISABEL are also reading from FT11F001 (if initialized). Writing of data on the dataset is done (directly or by the subroutine WDA) in the subroutines PART1 (initialisation), REGINE, MARY01, ANNI01, SUZY, ISABEL, and FILE5. WDA is also called by the subroutine PART1 to print the table of contents and addresses of file 11. Reading and processing (insertion in a KEDAK-library *) the data of file 11 is only done by the subroutine SICKY.

DD-parameters:

The allocated dataset must reside on a direct access storage (e.g. a disk). The blocksize (BLKSIZE) must be 1600 bytes, the record format (RECFM) must be fixed (F). You can give the DCB-subparameter DSORG=DA. In any case the space must be given in units of the blocksize 1600, e.g. SPACE=(1600,(400),CONTIG) (CONTIG means, that the space must be contiguous.) The contents of the SPACE-parameter are used in the subroutine DEFI (see 2.3.5.93).

Possible errors:

BLKSIZE not equal 1600 bytes.

SPACE not given in units of 1600 bytes.

Not enough SPACE given (Error produced is of the form "Record number out of range for .....").

---

* The KEDAK library must not be processed with or created by KEMA; it must be created by BRIGITTE itself (see 2.2.4).

FT12F001

Description:

Auxiliary dataset. Used to store ENDF/B TAB1- and LIST-records, which have been read from the ENDF/B file, but cannot be processed at the moment. Also data calculated in the program, which will be used later, are stored on FT12F001. The structure of the dataset is described in 2.3.5.78.

Actions taken:

The data are written on the dataset by the subroutine WRECS (see 2.3.5.78) and read from it by the subroutine RRECS (see 2.3.5.79). Writing of data is done in the subroutines PART1 (initialisation), INGRID, MELA, ANNIO1, DENISE and SOPHIE.

The data are read in the subroutines MYRIAM, MARION, SOPHIE and SUZY.

DD-parameters:

The allocated dataset must reside on a direct access storage (e.g. a disk). The blocksize (BLKSIZE) must be 1600 bytes, the record format (RECFM) must be fixed (F). The DCB-subparameter DSORG=DA can be given. Because the contents of the SPACE-parameter are used in the subroutine DEFI (see 2.3.5.93), the space must be given in units of the block-size 1600, e.g.

SPACE=(1600,(400),,CONTIG).

(CONTIG means, that the space must be contiguous)

Possible errors:

BLKSIZE not equal 1600 bytes.
SPACE not given in units of 1600 bytes.
Not enough SPACE given (Error produced is of the form "Record number out of range for ...").

If ITAPE (see 2.2.6 namelist INPUT) is equal 1, the following ddname must exist:

FT15F001

Description:

Datatypes of ENDF/B MF=3 with added resonance parts are written in ENDF representation on the dataset.

Actions taken:

Writing (including HEAD- and SEND-records) is done in the subroutine SUZY (see 2.3.5.37) by using the subroutines WCONT (see 2.3.5.50) and WREC (see 2.3.5.51).

DD-parameters:

Because writing is done with MODE=3, the record format (RECFM) must be fixed (F) or fixed blocked (FB), the logical record length (LRECL) must be equal 80 bytes. The dataset can reside on a disk, a tape, a drum, or something like that. The data can also be punched.

Possible errors:

Completion code D37, because not enough SPACE has been given

2.2.6 Input description

The program BRIGITTE is controlled by the contents of three (plus one) namelists. All three namelists must be given in the input, but some can be empty. If more than one material is to be converted in one job, these three namelists must be given for each material.
The namelist input is read from unit FT05F001 respectively SYSIN.
The sequence of the namelists is as following:

```
&SPACE   data-list or empty  &END
&INPUT   data-list           &END
{Only for NISO greater 1:          }
{&ISO    data-list           &END  }
&LIB     data-list or empty  &END
```

In the description of the variables of all namelists, the abbreviations, described in the following, are used.

R4	means	REAL*4 quantities
R8	means	REAL*8 quantities
I4	means	INTEGER*4 quantities
L4	means	LOGICAL*4 quantities

In the "dimension column" either the maximal dimension for arrays or the word scalar for scalar quantities is given.
The values in the "default value column" are used, if the variable is not found in the namelist input. If no default value exists a "-"sign is made.

There are some syntax rules for the namelist input data, which can be found in FORTRAN manuals (see e.g. /9/). The important rules will be listed in the following.

(1) The first character in each record to be read must be a blank.

(2) The second character must be an &, immediately followed by the namelist name, which must not contain any embedded blanks.

(3) The namelist name must be followed by at least one blank.

(4) After this blank the data items separated by commas are listed. (A comma after the last item is optional.) There may be blanks between the comma and the next data item symbolic (resp. array) name.

(5) The end of a specified namelist input is signaled by &END.

(6) The input for one namelist can be given on more than one card. The data item list should end with a comma on one card and can be continued in each column of the next card.

(7) A data item is of the form

symbolic name = constant

or

array name = set of constants
(separated by commas)

The symbolic name can be an array element name (e.g. A(21)) or a variable name (e.g. V).
(Valid symbolic or array names are given in the following tables).
The constants may be of the type integer (e.g. 5), real (e.g. 3.0E+3), literal (e.g. 'ALPHA'), complex, or logical (e.g. .TRUE., or F).

(8) In the case of a set of constants with the same value, you can write k* constant, where k gives the number of times the value constant is to occur.

(9) If the input for one namelist is distributed over several cards, the cards must not contain a numeration in columns 73 to 80.

Examples for the namelist input are given in 2.2.10.

In the following tables a description of all variables of all namelists is given. Normally only input for the namelist INPUT is necessary, whereas the namelist input for SPACE and LIB can be of the form

```
 &SPACE &END
```

respectively

```
 &LIB &END
```

In the case of the namelist INPUT a survey of all necessary quantities is given at the end of the table.

namelist name: SPACE

name	type	dimension	default value	meaning
NBKX	I4	scalar	12 000	maximal dimension of the array BK (see 2.3.3)
N1X	I4	scalar	500	maximal dimensions of the arrays NBT and JNT (see 2.3.3)
N2X	I4	scalar	5 000	maximal dimensions of the arrays X and Y (see 2.3.3)
NBX	I4	scalar	3 000	maximal dimensions of the array B (see 2.3.3)
IADIM	I4	scalar	20 000	maximal dimensions of the array A (see 2.3.3)
ITDIM	I4	scalar	32 560	maximal dimension of the array TAB (see 2.3.3)

Caution: The default values of the quantities in the namelist SPACE should be only changed by the input, if the user has a sufficient knowledge of the organization of the data storage in the program as described in section 2.3.3. If there are difficulties with the maximal dimension of an array, we recommend to enlarge the total region and to give only "&SPACE &END" with an empty list.

namelist name: INPUT

name	type	dimension	default value	meaning
MODE	I4	scalar	3	input format of ENDF/B-data (see /3/ chapter R-2.2). (MODE=3 means card image tape with standard arrangement.)
LABEL	I4	scalar	-1	< 0 TPID-record (see /3/ chapter R-2.3.6) is not read. The tape identification number is not checked. = 0 TPID-record is read. The tape identification number is printed out but not checked. > 0 TPID-record is read. The tape identification number is printed out and must be equal LABEL. We recommend to give always the correct tape identification number and not to use LABEL-values $\leq$ 0.
MATP	I4	scalar	-	ENDF/B-material number
NAMKDK	R8	scalar	-	alphamerical KEDAK-name of the material MATP
MATKDK	I4	scalar	-	numerical KEDAK-name of the material MATP
EB	R4	scalar	-	binding energy of the last neutron (in eV)

namelist name: INPUT (continuation)

name	type	dimension	default value	meaning
NOTTRL	R8	(25)	'blank'	data types contained in this array (as KEDAK datatype names), are not translated from ENDF/B to KEDAK. These data types - all sections of ENDF/B MF=3 - can be: SGT, SGN, SGX, SGI, SG2N, SG3N, SGF, SGIA, SGI3A, SG2NA, SG3NA, SGA, SGIP, SGI2A, SGIZ, SGIZC, SGG, SGP, SGD, SGH3, SGHE3, SGALP, SG2HE, SGTR, ETA, ALPHA, MUEL.
KEDAVE	I4	scalar	1	0 an old not longer used KEDAK version will be created. (All SGIZ and SGIZG will have the same energy grid.) 1 the translated data will fulfil the valid KEDAK conventions.
NISO	I4	scalar	0	number of isotopes in the ENDF/B material, for which resonance (and some other) information is given.(NISO=0 and NISO=1 have the same meaning: resonance information is given only for one isotope.) If NISO > 1 the namelist ISO must be given in the input

namelist name: INPUT (continuation)

name	type	dimension	default value	meaning
LNER	I4	scalar	0	used in the subroutine MARY (see 2.3.5.12). Maximum number of points per resonance used in calculating the energy grid in the resolved resonance region with the subroutine EPT (see 2.3.5.17). 0 51 points per resonance are used ≥1 LNER points per resonance - with a maximum of 200 - are used
MLSLSW	I4	scalar	0	≥1 single level Breit Wigner formalism is used in the resolved resonance region to calculate pointwise cross sections independently from the ENDF/B recommended formalism. =0 the ENDF/B recommended formalism (single or multilevel Breit Wigner) is used
ELINEG	I4	scalar	0	Deletion of negative SGT and SGN cross section values. ≠1 negative SGT and SGN cross section values are printed out =1 the values are printed out and deleted

namelist name: INPUT (continuation)

name	type	dimension	default value	meaning
IF4	I4	scalar	0	control of calling the subroutine ISABEL (Processing of ENDF/B MF=4 data; angular distributions of secondary neutrons. See 2.3.5.42) = 1 only the subroutine ISABEL is called in BRIGITTE (only angular distributions of secondary neutrons are converted) =-1 the subroutine ISABEL is not called in BRIGITTE (angular distributions of secondary neutrons are not converted) ≠ 1 and ≠ -1 normal processing of BRIGITTE (ENDF/B MF=4 data are included in the conversion process)
IF5	I4	scalar	0	control of calling the subroutine FILE5 (Processing of ENDF/B MF=5 data: energy distributions of secondary neutrons. See 2.3.5.44). = 1 and IF4≠1 only the subroutine FILE5 is called (only energy distributions are converted) =-1 and IF4≠1 the subroutine FILE5 is not called in BRIGITTE (energy distributions of secondary neutrons are not converted) ≠ 1 and ≠ -1 and IF4≠1 normal processing of BRIGITTE (ENDF/B MF=5 data are included in the conversion process)

namelist name: INPUT (continuation)

name	type	dimension	default value	meaning
ICOMB	I4	scalar	0	used in the subroutine COMBP (see 2.3.5.67) ≠ 1 the optimal interpolation scheme is choosen by the routine = 1 only a linear-linear interpolation scheme is used Caution: This variable should have always the value 0 in production runs
EPSRES	R4	scalar	1.0E-2	accuracy used in the following cases: - energy grid used in the calculation of pointwise cross sections of resolved resonance parameters. It is guaranteed that a cross section value between two energy points can be predicted by linear interpolation with a relative error less or equal EPSRES - in the calculation of pointwise cross sections out of unresolved resonance parameters the energy grid created guarantees that each of the statistical resonance parameters can be linearly interpolated with an error less or equal EPSRES - in changing the interpolation methods used in ENDF/B to linear-linear. It is guaranteed that functional values between two argumental values can be predicted by linear interpolation with

namelist name: INPUT (continuation)

name	type	dimension	default value	meaning
				an error less or equal EPSRES. In some cases the program increases the given value of EPSRES temporarily up to $2.0 \cdot 10^{-2}$ - in the combination of two pointwise cross sections with non linear interpolation rule. It is guaranteed, that cross section values between two energy points can be predicted - using the associate interpolation rule - with an error less or equal EPSRES - in cropping unnecessary points, before inserting the data in KEDAK, functional values, which can be predicted by linear interpolation with an error less or equal ε, are cropped. Normally ε has the value SCREPS (see next variable). But if two adjacent points fulfil the condition $\left\|\frac{x_2-x_1}{x_1}\right\| < 5.0 \cdot 10^{-6}$ ε is increased to the value EPSRES. (EPSRES should always be greater than SCREPS). (see 2.3.5.75)

namelist name: INPUT (continuation)

name	type	dimension	default value	meaning
				- in the construction of scratch classes (see 2.3.5.75) in the subroutine DZMSCH (see 2.3.5.40). In this case 10 classes are built up with the boundaries 0, EPSRES, EPSRES + $\frac{1.-EPSRES}{9.0}$, EPSRES + 2 • $\frac{1.-EPSRES}{9.0}$, ... , EPSRES + 8 • $\frac{1.-EPSRES}{9.0}$, 1.0. Points, which are replaced by DZMSCH, are checked, with which accuracy they could have been interpolated. This accuracy defines the membership in a specified scratch class. The number of points in the scratch classes is printed out. EPSRES is also used in the construction of scratch classes in the subroutine CROP (see 2.3.5.75). In this subroutine it has the name EPSMAX.
SCREPS	R4	scalar	$5.0 \cdot 10^{-3}$	Before inserting the pointwise cross section into KEDAK and after combination of two cross sections it is checked, whether there are points, which can be predicted by interpolation between adjacent points to relative accuracy of ε.

namelist name: INPUT (continuation)

name	type	dimension	default value	meaning
				Points with an ε better than or equal to SCREPS (resp. EPSRES, see description of this quantity) are deleted. SCREPS and EPSRES (see variable before) are also used in the subroutine CROP to built up - for printing purposes - so called scratch classes (see 2.3.5.75 "problem solved").
FAICOD	R4	scalar	10.0	used in combining two pointwise cross sections in the subroutine COMBP (see 2.3.5.67 "problem solved" point (5)). The variable FAICOD should not be changed by the user. Only program users, who have a very well knowledge of the program BRIGITTE, can try to work with this variable.

namelist name: INPUT (continuation)

name	type	dimension	default value	meaning
LEDIT1	I4	scalar	1	control of print output = 1 descriptive data and dictionary of ENDF/B (MF=1, MT=451) are printed. ≠ 1 these data are not printed.
LEDIT2	I4	scalar	1	control of print output ≠ 1 the data, listed in the following, are not printed. = 1 the data, listed in the following, are printed. - resolved resonance parameters in ENDF and KEDAK representation - unresolved resonance parameters in ENDF representation - the KEDAK data types ST and STGF (for fissionable materials)
LEDITS	I4	scalar	0	Control of print output ≠ 1 the data, listed in the following, are not printed. = 1 the data, listed in the following, are printed. - pointwise cross sections calculated out of resolved and unresolved resonance parameters - pointwise total cross sections calculated out of resolved and unresolved resonance parameters - ENDF/B (pointwise) reaction types of MF=3

namelist name: INPUT (continuation)

name	type	dimension	default value	meaning
BIB12	I4	scalar	1	number of records of the length 400 4-byte words used for the table of contents and addresses of the direct access data set NF=12 (see 2.3.5.78). The maximal value for BIB12 is 10.
ITAPE	I4	scalar	0	control of writing the converted data of ENDF/B MF=3 on unit 15. The data written in the subroutine SUZY (see 2.3.5.37) are neutron cross sections of ENDF/B file 3 with added resolved and unresolved resonance contributions. The interpolation scheme is linear-linear and all unnecessary points have been deleted (all conversion work of this type has been done). But writing is done in ENDF mode=3 (card image). ≠ 1 no writing is done = 1 writing is done
SALINE	L4	scalar	.FALSE.	used in the subroutine SOPHIE (see 2.3.5.36). Normally the program assumes, if in ENDF/B MF=3 is no section MT=51, that there are also no sections 52,53,...,90. But there are some materials with special data types stored for example under MT=54 with no MT=51,52,53 existing. To be able to convert such cross section types, SALINE is used. = .FALSE. if MT=51 is not found, no search for MT=52,53,..,90 is made. = .TRUE. if MT=51 is not found, a search is started to find higher MT-values.

In the namelist INPUT there are quantities, which must be given, should be given, must be given only for special cases, and should not be given. These quantities are:

(1) quantities, which must be always given in the input:

MATP,NAMKDK,MATKDK,EB

(2) quantities, which should be given in the input:

LABEL

(3) quantities, which must be given in the input only for special cases:

MODE (ENDF/B-data not in card image)

NOTTRL (suppress translation of some datatypes)

KEDAVE (old KEDAK conventions wished)

NISO (information for more than one isotope given in the resonance region)

MLSLSW (always single level Breit Wigner desired)

ELINEG (suppress negative σ_t and σ_e cross section values)

IF4 (control of processing angular distributions of secondary neutrons)

IF5 (control of processing energy distributions of secondary neutrons)

EPSRES (change of the accuracy)

SCREPS (" " " ")

LEDIT1, LEDIT2, LEDITS } (more or less print output desired)

BIB12 (table of contents too small. In this case BRIGITTE produces an error message and a STOP)

SALINE (no MF=3 MT=51 but higher MT occurs)

(4) quantities which should not be given in the input:

LNER (if too small, not enough or too much points may be produced)

ICOMB, FAICOD } (should only be used after a very good knowledge of the program

ITAPE (there is normally no use for the produced dataset)

namelist name: ISO (Only given for NISO > 1. See namelist INPUT)

name	type	dimension	default value	meaning
NAMISO	R8	(10)	-	alphamerical KEDAK-names for the additional isotopes of a material (NISO-names) (see 2.2.10)
MATISO	I4	(10)	-	numerical KEDAK-names - associated to the alphamerical ones - for the additional isotopes of a material (NISO-names) (see 2.2.10)
EBISO	R4	(10)	-	binding energies of the last neutron for the NISO additional isotopes (NISO-values). (EB, in the namelist INPUT, means for NISO > 1 the binding energy of the last neutron of the natural element.)

namelist name: LIB

name	type	dimension	default value	meaning
IFIRST	I4	scalar	0 [1]	≤ 0 the KEDAK library [2] is existing. The converted material has to be added to this library. ≥ 1 it is assumed that the direct access data set contains no KEDAK library. The KEDAK library is initialized. Caution: Existing libraries are destroyed by IFIRST ≥ 1
IMOD	I4	scalar	0 [1]	$= 0$ the converted material is not yet present in the library. It is to be inserted. (If it is present no insertion is done.) $\neq 0$ material yet present in the library. It is to be replaced. (If it is not present, a warning message is printed and the converted material is inserted in the KEDAK-library.) The quantity IMOD is only meaningful for IFIRST ≤ 0
IDAT	I4	scalar	date [1] fetched by the routine DATUM (see 2.3.5.98)	creation (or change) date of the KEDAK-library in the form ddmmyy secifying day, month, year (see /4/ or II). In nearly all cases the date fetched by the subroutine DATUM should be used.

[1] Before each reading of the LIB namelist input these default values are set.

[2] This library must not be created by or processed with KEMA; it must have been created by BRIGITTE in a previous run (see 2.2.4).

2.2.7 Output description

The printed output, produced by the program, is almost self explaining. Therefore only a general survey together with some helps for some details, where questions may arise, will be given. The production of some parts of the printed output is controlled by the values of the variables LEDIT1, LEDIT2, and LEDITS. These variables have default values but may also be given in the input (see 2.2.6 namelist INPUT).

The printed output of the program BRIGITTE starts with the program name in "big" letters. After this a list of the control input is printed. This list is followed by the results of the dynamic working space allocation (see 2.3.3). On the next page a printout of the contents of the namelist INPUT is given. After this the program name together with the material name and some other informations are printed.

Under the heading "ENDF/B FILE 1" the results of the conversion of the ENDF/B MF=1 data are printed. In this section first the printout of the not translated section MT=451 is given. This may be followed by a listing of the $\bar{\mu}$-data.

The next heading is "ENDF/B File 2". If desired, resolved and unresolved parameters in ENDF/B- and KEDAK-representation are printed together with the calculated pointwise cross sections. All pointwise cross sections are listed in the ENDF/B MODE=4 (see /3/ chapter R 2.2.4). At the end of the printout of the pointwise total cross sections, calculated from resolved resonance parameters by adding the partial cross sections, the accuracy EPS used in combining the partial cross sections is printed out.

After this the text "FILE 3 - SMOOTH CROSS SECTIONS" is printed starting on a new page. Under this heading the cross sections read from ENDF/B MF=3 are listed. If there is a resonance part, which has been calculated from the data of MF=2, this part has been added before printing. The printout of all these cross sections is controlled by the quantity LEDITS (see 2.2.6 namelist INPUT). The printout also includes the calculated cross sections as e.g. σ_a and σ_{tr}. In the case of σ_a the partial cross section names, used in calculating σ_a, are listed.

After changing the interpolation rule to linear-linear, deletion of points, which can be predicted by linear interpolation, combination of points with nearly equal energy values, and suppressing of negative cross section values, all cross sections read from MF=3 (including the calculated ones) are printed once more. These data will be inserted into the KEDAK-library. Each listing of a cross section type is headed with some information. In the first line the ENDF/B MT-number is listed together with the used accuracies. The "PRECISION EPS" is used in changing the interpolation rule, "EPSMIN" in scratching points, which can be predicted by linear interpolation, and "EPSMAX" in scratching points, which can be predicted by linear interpolation and are having nearly equal energy values. This first line is followed by two lines, which will show the number of points, which have been scratched due to the possibility of linear interpolation. These points are classified in so called "scratchclasses". If e.g. 38 points are in the scratchclass 0.2 % to 0.3 % (1 % means 1 percent), this means, that these scratched points could have been predicted with an accuracy in the range 0.2 % to 0.3 %.

If there are some points with nearly equal energy values, which have endured the scratching process, these points will be combined. Some information about this combination will be printed. The accuracy classes have the same meaning as the described scratchclasses.

If there are negative σ_t or σ_e cross section values, the confounded energy rangres will be printed. If desired (see ELINEG in the namelist INPUT in 2.2.6), the values will be suppressed. All this information is followed by the KEDAK cross section type name printed in "big" letters and finally the data itself. In order to know where the computing time is spent, an asterisk surrounded text is printed. This text includes the date, the time of the day, and the consumed CPU-time (CPU abbreviates Central Processing Unit). This information may be found very often in the program printout.

The next heading is "FILE 4 - ANGULAR DISTRIBUTIONS OF SECONDARY NEUTRONS". This part ends with "REACTION TYPE TABLES". The first column in these tables means the energy. In the "polynomial" table, which is the first one, if a polynomial representation exists, the second and the third column means internal addresses. In the "pointwise" table, which is normally the second one, but may also be, if no polynomial representation exists, the first one, the second column means the number of cosine values. The third and fourth column are internal addresses.

The next heading is "FILE 5 - ENERGY DISTRIBUTIONS OF SECONDARY NEUTRONS". If a secondary energy distribution law (used in ENDF/B) can not be processed by the program, an one line text is printed, which may be overlooked, if the printoutput is inspected only very carelessly. We recommend, at least in the beginning, to read the program printoutput very carefully. All data type names, which will be included into KEDAK, are listed under the heading "CONTENTS OF FILE 11". In the first column the numerical KEDAK material name can be found. The second column means the numeration of the isotope (normally 1). In the third column the alphameric KEDAK datatype name is listed. The fourth column means the record number in file 11, where the data starts. The number of these data can be found in column five, which is the last one.

The section headed with "INSERTION OF TRANSLATED DATA INTO KEDAK" starts with a printout of the namelist LIB, followed by some messages, resp. warnings. After this the first words of record 1 of the KEDAK-library are listed, followed by the material conversion table, the type conversion table and the material address table (see II.1). After this the type address table for the inserted material is listed. The last line of this section is

NEXT AVAILABLE RECORD IN THE KEDAK LIBRARY

followed by a number. This number must not be grater than the extents (plus 1) of the library measured in 3520 byte records. The last section is a "PRINTOUT OF THE KEDAKTABLES". The general data of the KEDAK-library, the material conversion table, the type conversion table, the material address table, and the type address table are listed in an edited form. The last line of the printoutput of the program is

"***** NORMAL END OF TABPRT *****"

We recommend, to check whether all addresses in the KEDAK-tables are within the library extent.

2.2.8 Listing of possible errors

In this section a listing of all possible not self explaining error messages, which are printed by the program, is given.

Messages generated by the operating system or input/output routines will not be listed here, because these messages must be self explaining. Messages relating to auxiliary datasets see section 2.2.5.

The error messages produced by the program have all the same format. First the text

ERROR STOP lmn

with lmn to be found in the following tables, is printed. Then the values of MAT, MF, and MT in the Common RECS are printed out followed by the text

DUMP OF THE ARRAYS NBT,JNT,X,AND Y.

These arrays, which are a part of the data storage (see 2.3.3), are normally used to store a TAB1-record (see /3/ chapter R-2.3.3). After the printout of the contents of the arrays a STOP 1 is made.

In the list on the following pages first the error number lmn is tabulated. Then the names of the subroutines, where this error can happen, are specified. Afterwards the reason for the occurance of the error number lmn is given. In the last column proposals for possible corrections are made.

In nearly all cases the numbered error stops are caused by exceeding the maximal dimensions of an array. That means, increasing the total region or the maximal dimension of an array, will correct the error. But there are also other errors. Therefore it is better, before increasing the region, to look at the following table.

The table of the numbered error messages is followed by the table "Possible corrections for the error type program or ENDF/B-data error". In the last part of this section some hints for other error situations are given. The remarks (1) and (2) will be found at the end of the numbered error messages.

error number	happened in subroutine (s)	reason	possible correction(s)
99	RREC	ENDF/B-tape number NT less or equal zero	Program or ENDF/B-data error. See special table at the end of this section
100	RREC	JT out of range 1-6	
101	RREC	MODE out of range 1-3	
102	RREC	Temperature T not in range given in data	
103	RRECS FETCH	Maximal dimensions of the arrays NBT and JNT tto small.	Increase total region or N1X in the namelist SPACE (see 2.2.6). See also remarks (1) .
	RREC	Maximal dimensions of the arrays NBT and JNT too small or incorrect ENDF/B data (N1 less or equal zero).	In the first case, which is normally the reason for this error, increase the total region or N1X in the namelist SPACE (see 2.2.6). See also remarks (1) . In the second case, correct the ENDF/B-data, which are read by the subroutine.
104	RRECS FETCH	Maximal dimension of the array B too small	Increase the total region or NBX in the namelist SPACE (see 2.2.6)
	RREC	Maximal dimension of the array B too small or incorrect ENDF/B-data (N1 less or equal zero)	In the first case, which is normally the reason for this error, increase the total region or NBX in the namelist SPACE (see 2.2.6). See also remarks (2) . In the second case, correct the ENDF/B-data, which are read by the subroutine.
105	MELA ANNICK ADDPNT RRECS FETCH	Maximal dimension of arrays X and Y too small	Increase total region or N2X in the namelist SPACE (see 2.2.6). See also remarks (1) .

error number	happened in subroutine(s)	reason	possible correction(s)
105 (continuation)	RREC	Maximal dimension of the arrays X and Y too small or incorrect ENDF/B-data (N2 less or equal zero).	In the first case, which is normally the reason of the error, increase the total region or N2X in the namelist SPACE (see 2.2.6). See also remarks (1) . In the second case correct the ENDF/B-data read by the subroutine.
106	RREC	Improper temperature dependence	Program or ENDF/B-data error. See special table at the end of this section
107	RREC	Incorrect MAT,MF,MT on the ENDF/B-tape	Correct the ENDF/B data
108	WREC	JT out of range 1-6	Program or ENDF/B-data error. See special table at the end of this section
109	WREC	MODE out of range 1-4	
133	TERP1	Interpolation code is less than 1 or greater than 5	Program or ENDF/B-data error. See special table at the end of this section
134	DANI TERP1	-	
135	TERP1	Lower and upper end of the interpolation interval have the same value	
201	INGRID	Material MATP not found in specified ENDF/B library	Correct MATP in the namelist INPUT (see 2.2.6) or give correct ENDF/B library
202	INGRID	$\bar{\nu}$-data (number of neutrons per fission, MF=1,MT=452) not found for the specified material MATP although the material is flagged as fissionable and MT=452 is listed in the table of contents.	Correct the ENDF/B data

error number	happened in subroutine(s)	reason	possible correction(s)
203	LUCY	I out of range 2-3	Program or ENDF/B-data error. See special table at the end of this section
204	ANNICK ANNI01 REGINE	Maximal dimension of array BK too small	Increase total region or NBKX in the namelist SPACE (see 2.2.6)
205	INGRID	Maximal dimension of array A too small	Increase total region or IADIM in the namelist SPACE (see 2.2.6)
206	MARY MARY01	Maximal dimension of array BK too small	Increase total region or NBKX in the namelist SPACE (see 2.2.6)
207	MARY	Number of l-states, for which resolved resonance parameter sets are given, is greater than 5.	Can only be corrected by program changes if checking of ENDF/B-data leads to NLS > 5. (Due to ENDF conventions NLS must be less or equal 3).
208	MARY MILLI	Maximal dimension of array A too small	Increase total region or IADIM in the namelist SPACE (see 2.2.6)
209	MARCEL	Negative l-state number	Check ENDF/B resolved resonance data (quantity L). If correct, inform program author. See also special table "Possible corrections for the error type program or ENDF/B-data error".
210	REGINE	Number of isotopes given in ENDF/B MF=2 MT=151 data greater 10	Inform program author to increase the appropriate dimensions in the program
219	REGINE	Number of isotopes given in ENDF/B MF=2 M=151 not consistent with the number given in the control input	Correct NISO in the namelist INPUT and rerun the job
220	ANNICK	Maximal dimension of array A too small	Increase total region or IADIM in the namelist SPACE (see 2.2.6).

error number	happened in subroutine(s)	reason	possible correction(s)
221	ANNICK	Maximal dimension of arrays X and Y too small	Increase total region or N2X in the namelist SPACE (see 2.2.6). See also remarks (1) .
222	ANNICK	Maximal dimension of array A too small	Increase total region or IADIM in the namelist SPACE (see 2.2.6)
224	ANNIO1	-	Check unresolved resonance parameter set on consistence If correct, see special table "Possible corrections for the error type program or ENDF/B-data error".
225	ANNIO1	-	Check unresolved resonance parameter set on consistence If correct, see special table "Possible corrections for the error type program or ENDF/B-data error".
226	ANNIO1	Maximal dimension of array BK too small	Increase total region or NBKX in namelist INPUT (see 2.2.6).
300	STORE	JT out of range 1-6	Program or ENDF/B-data error. See special table at the end of this section
301	STORE	Identification number MA is equal zero	
310	COMB	Lower energy limit greater or equal upper energy limit	
311	COMB	Identification of the records to be combined is zero	
312	CROP	Incorrect interpolation tables NBT and JNT.	
313	CROP	Maximal dimensions of the arrays NBT and JNT too small	Increase total region or N1X in the namelist SPACE (see 2.2.6). See also remarks (1) .
314	IPDS	Incorrect interpolation tables NBT and JNT	Program or ENDF/B-data error. See special table at the end of this section

error number	happened in subroutine(s)	reason	possible correction(s)
317	SEARCH	MODE less 1 or greater 3	Correct MODE in the namelist INPUT (see 2.2.6). If correct, see special table "Possible corrections for the error type program or ENDF/B-data errors
400	INGRID	MODE out of range	MODE must be 1,2, or 3 (see 2.2.6 namelist INPUT variable MODE)
401	INGRID	ENDF/B type identification number does not match with variable LABEL	Correct variable LABEL in the namelist INPUT (see 2.2.6)
403	MYRIAM	Maximal dimension of the arrays X and Y too small	Increase total region or N2X in the namelist SPACE (see 2.2.6). See also remarks (1) .
404	MYRIAM	-	Program or ENDF/B-data error. See special table at the end of this section
407	MYRIAM	Maximal dimension of array A too small	Increase total region or IADIM in namelist SPACE (see 2.2.6)
408	MARION	Maximal dimension of array A too small	Increase total region or IADIM in the namelist SPACE (see 2.2.6)
409	MARION	Maximal dimensions of the arrays X and Y too small	Increase total region or N2X in the namelist SPACE (see 2.2.6). See also remarks (1) .
410	MARION	-	Program or ENDF/B-data error. See special table at the end of this section
500	SOPHIE	Maximal dimension of the array A too small	Increase total region or IADIM in the namelist SPACE (see 2.2.6)
501	SOPHIE	Maximal dimension of the arrays X and Y too small	Increase total region or N2X in the namelist SPACE (see 2.2.6). See also remarks (1) .

error number	happened in subroutine(s)	reason	possible correction(s)
502	SOPHIE	-	Program or ENDF/B-data error. See special table at the end of this section
505	SOPHIE	Maximal dimension of the arrays X and Y too small	Increase total region or N2X in the namelist SPACE (see 2.2.6). See also remarks (1) .
506	SOPHIE	Maximal dimension of the array BK too small	Increase total region or NBKX in the namelist INPUT (see 2.2.6).
510	SOPHIE	Maximal dimension of the array BK too small	Increase total region or NBKX in the namelist SPACE (see 2.2.6).
515	SUZY	Maximal dimension of the array BK too small	Increase total region or NBKX in the namelist SPACE (see 2.2.6).
516	WDA	Maximal dimension of array BK too small	Increase total region or NBKX in the namelist SPACE (see 2.2.6).
550	WDA	Too many entries in the table of contents and addresses of file 11	Probably program error. Rerun the job. See also special table "Possible corrections for the error type program or ENDF/B-data error. If the error persists, reduce the number of datatypes converted by using NOTTRL in the namelist INPUT (see 2.2.6). In any case inform the author of the program
600	ANNICK	Number of j-states for a particular l-state for statistical resonance parameter sets is greater than 6.	Check ENDF/B data (unresolved resonance parameter set, quantity NJS). If NJS is less than or equal to 6 inform program author, else program changes must be done due to changes in ENDF conventions.
601	ANNICK	Number of l-states in a statistical resonance parameter set > 3.	Check ENDF/B data (unresolved resonance parameter set, quantity NLS). If NLS is ≤ 3 inform program author, else program changes must be made due to changes in ENDF conventions.

error number	happened in subroutine(s)	reason	possible correction(s)
620	DANT	Changing interpolation to linear-linear, 300 points had not been enouth to represent a non linear interval	Check whether this is correct. If true increase the arrays in the common RE and the number 300 in the subroutine DANI. Inform the program author
621	CHGINT	Interpolation code less 1 or greater 5 found in the array JNT	Program or ENDF/B-data error. See special table at the end of this section
700	ISABEL	Maximal dimension of array BK too small	Increase total region or NBKX in the namelist SPACE (see 2.2.6).
701	ISABEL	Too many entries in the table of contents and addresses of file 11	See error number 550
703	ISABEL	Maximal dimension of array A too small	Increase total region or IADIM in the namelist SPACE (see 2.2.6).
750	FILE5	Maximal dimension of the arrays X and Y too small	Increase total region or N2X in the namelist SPACE (see 2.2.6). See also remarks (1) .
751	FILE5	Maximal dimension of array ENERG in FILE5 too small	Check ENDF/B data of MF=5, which are to be converted, on correctness. If no errors found, increase NCMAX and dimension of the array ENERG in the subroutine In any case inform the author of the program.
752	FILE5	Maximal dimension of the array BK too small	Increase total region or NBKX in the namelist SPACE (see 2.2.6).
777	MILLI MARION MYRIAM ELIMZ SUZY DZMSCH ADDPNT SUPNEG CHGINT CROP		Program or ENDF/B-data error. See special table at the end of this section

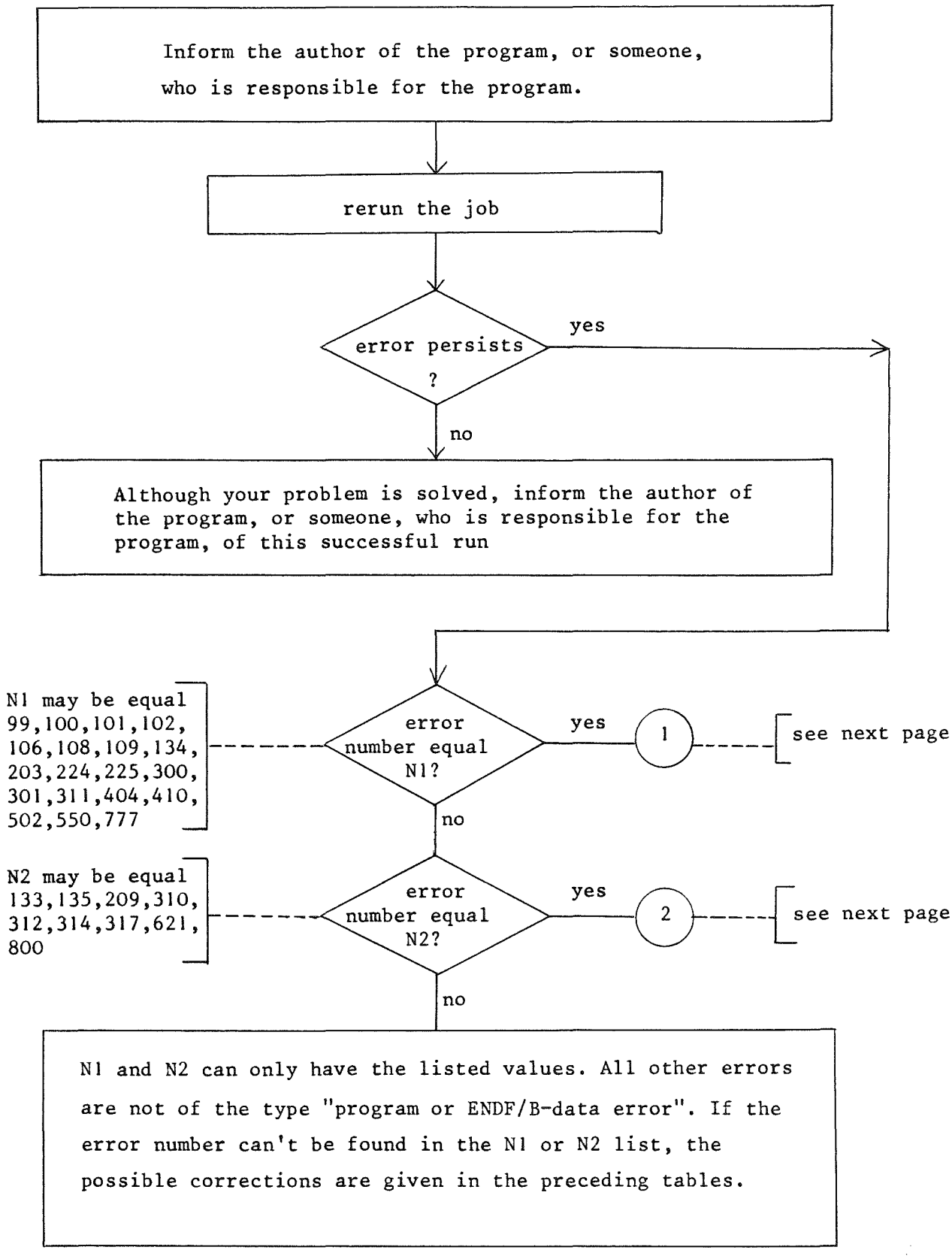
error number	happened in subroutine(s)	reason	possible correction(s)
799	ISABEL	Maximal dimension of array A too small	Increase total region or IADIM in the namelist SPACE (see 2.2.6)
800	ITAB1	Incorrect interpolation tables (arrays NBT and JNT).	Program or ENDF/B-data error. See special table at the end of this section
850	SUZY	Maximal dimension of the arrays X and Y too small	Increase total region or N2X in the namelist SPACE (see 2.2.6). See also remarks ① .
851	SUZY	Maximal dimension of the arrays X and Y too small	Increase total region or N2X in the namelist SPACE (see 2.2.6). See also remarks ① .

① Before increasing the total region or maximal dimensions of arrays, look at the printout of the error message. There is a part headed with "***DUMP OF THE ARRAYS NBT,JNT,X, and Y***". In this part the values of N1 and N2 can be found. Are these values correct? If not, correct the ENDF/B-data.

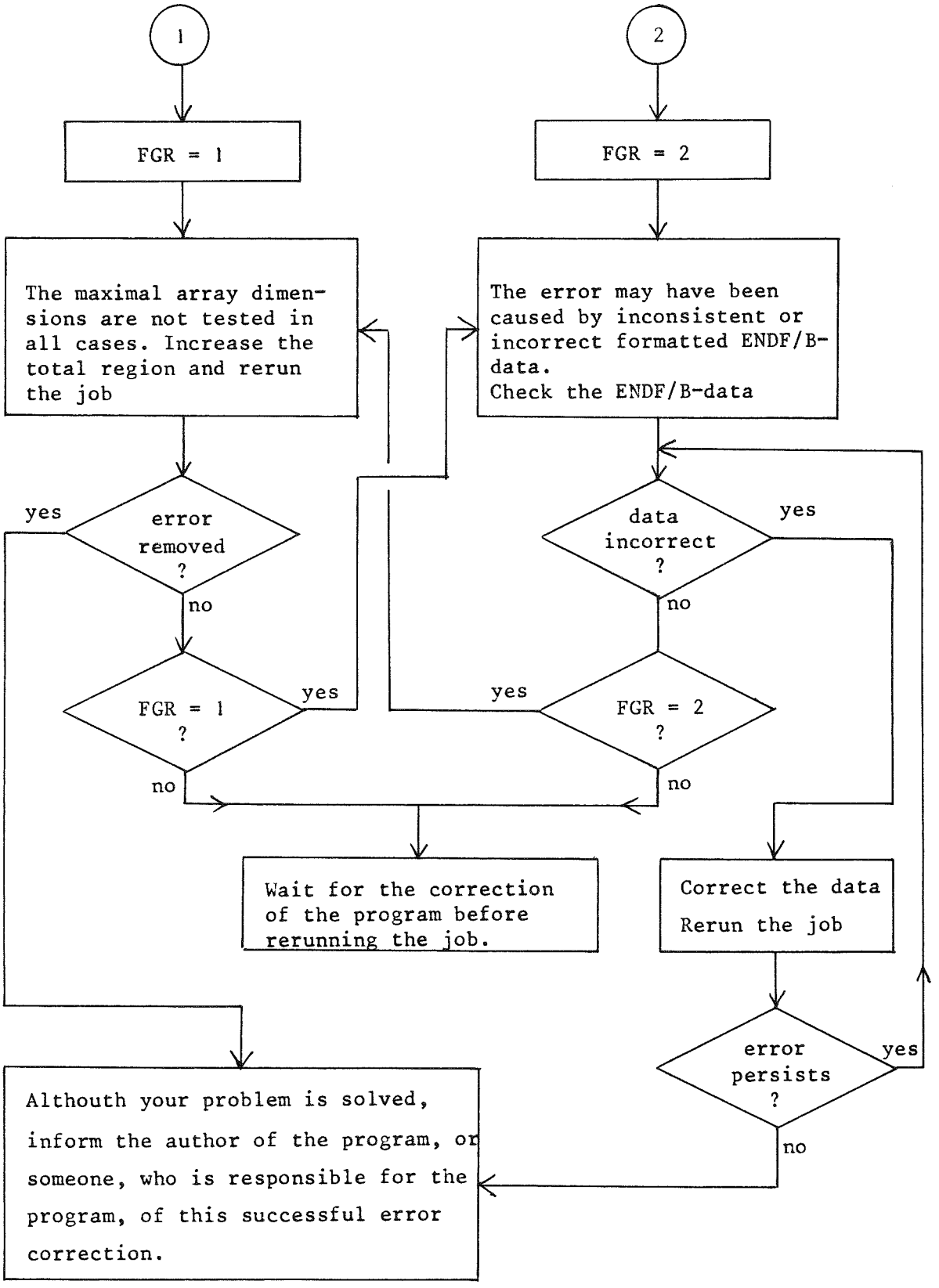
② Before increasing the total region or maximal dimensions of arrays, check - if possible -, whether the ENDF/B-LIST-record, which is processed, is correct. If not, correct the ENDF/B-data.

Possible corrections for the error type

"program or ENDF/B-data error".



Possible corrections for the error type "program or ENDF/B-data error" (continuation of the preceding page).



Unnumbered error messages
In the case of an error message produced by system input output routines, which can not be explained, check the following: - SPACE-parameter given for FT01F001 (see 2.2.5)? - SPACE-parameter for FT01F001 in units of 3520 bytes? - SPACE-parameter for FT11F001 and FT12F001 in units of 1600 bytes?
In case of the error message RECORD NUMBER N OUT OF RANGE ON UNIT M with M equal 11 or 12, increase the appropriate SPACE. If M is equal 1, give another or a new KEDAK-library.

2.2.9 JCL-cards for an IBM 05370

To run the program BRIGITTE on an IBM computer of the series 360 or 370, which are controlled by an operating system OS360 respectively OS370, the following JCL-cards are necessary:

```
(1)   //    (job card information),
(2)   //   REGION=nnnn K,TIME=(minutes)
(3)   //  EXEC PGM=progname
(4)   //STEPLIB DD DSN=dsname,UNIT=type,
(5)   //  VOL=SER=volname,DISP=SHR
(6)   //SYSABEND DD DUMMY
(7)   //FT05F001 DD DDNAME=SYSIN
(8)   //FT06F001 DD UNIT=(CTC,,DEFER),LABEL=(,NL),
(9)   //  DCB=(LRECL=133,BLKSIZE=1995,RECFM=FBA)
(10)  //FT01F001 DD UNIT=typek,VOL=SER=volnamek,
(11)  // DISP=(dispka,dispkn)DSN=dsnamek,
(12)  // DCB=(RECFM=F,BLKSIZE=3520,LRECL=3520,DSORG=DA),
(13)  // SPACE=(3520,(pq))
(14)  //FT09F001 DD UNIT=typee,VOL=SER=volnamee,
(15)  // DISP=(dispea,dispen),
(16) {// LABEL=(ln,la,,IN),}
(17) {// DCB=(RECFM=recfm,BLKSIZE=blksize,LRECL=lrecl,DEN=den),}
(18)  // DSN=dsnamee
(19)  //FT10F001 DD UNIT=types,
(20)  // SPACE=(blk10,(pq10,sq10)),
(21)  // DISP=(NEW,DELETE,DELETE),
(22)  // DCB=(RECFM=recfm10,BLKSIZE=blk10)
(23)  //FT11F001 DD UNIT=types,
(24)  // SPACE=(1600,(pq11),,CONTIG),
(25)  // DISP=(NEW,DELETE,DELETE),
```

```
(26)   // DCB=(DSORG=DA,RECFM=F,BLKSIZE=1600)
(27)   //FT12F001 DD UNIT=types,
(28)   // SPACE=(1600,(pq12),,CONTIG),
(29)   // DISP=(NEW,DELETE,DELETE),
(30)   // DCB=(DSORG=DA,RECFM=F,BLKSIZE=1600)
(31)  {//FT15F001 DD  parameter}
(32)   //G.SYSIN DD *
            control input(see 2.2.6)
(33)   //
```

Explanation of the parameters in the listing of the JCL-cards:

We have used the rule, that information, which must be given by the user, is written with underlined lowercase characters. The other characters must be given as typed.

card (1)

(job card information) all necessary information to start a job (see /10/).

card (2)

nnnn	region size. You should start with a value of 300.
minutes	estimated time value in minutes. You should start with values of 1 to 10 minutes.

card (3)

progname	name of the program in the partioned dataset dsname (see card (4)). For GFK-users progname = BRIGITTE.

card (4)

dsname	name of the partioned dataset containing the program named progname (see card (3)). For GFK-users dsname = STEINPDS

type — unit containing dsname. For GFK-users type=3330.

card (5)

volname — name of the volume containing dsname (see card (4)). For GFK-users volname=KAPROS.

card (10)

typek — unit containing the KEDAK-dataset+) dsnamek (see card (11)). Must be a direct access device.

volnamek — name of the volume containing the KEDAK-dataset+) dsnamek (see card (11)). Must be a direct access device.

card (11)

dispka — Must be NEW for new datasets, or OLD for old datasets.

dispkn — Must be KEEP if the KEDAK-library+) should be kept. If it should be deleted, it can be DELETE.

dsnamek — dataset name of the KEDAK-library+).

card (13)

pq — extension of the KEDAK-library measured in units of the BLKSIZE=3520. This card must also be given for old datasets.

card (14)

typee — unit containing the ENDF/B-data. (Normally a tape unit).

volnamee — name of the volume containing the ENDF/B-data (Normally a tape).

+) Never use a KEDAK-library created by or processed with KEMA to receive the translated data. This library must have been created by BRIGITTE itself. (see 2.2.4).

card (15)

dispea	In the case of ENDF/B-data stored on a tape dispea=OLD. In other cases dispea=SHR may be better.
dispen	In the case of ENDF/B-data stored on a tape dispen=PASS. In other cases the subparameter dispen can be omitted.

card (16)

This card is only given for tapes.

ln	number of the label.
la	SL for standard label tapes. NL for non label tapes.

card (17)

This card must be normally only given for non label tapes.

recfm	record format of the dataset
blksize	block size of the dataset
lrecl	logical record lenght of the dataset
den	density, with which the data have been written.

card (18)

dsname	dataset name of the ENDF/B-dataset. (Not used for non label tapes).

card (19)

types	unit, which will contain, the auxiliary dataset. For GFK-users types=SYSDA.

card (20)

blk10	block size for the auxiliary dataset. Recommended values see 2.2.5.
pq10	primary space quantity for the auxiliary dataset. Recommended values see 2.2.5.

sq10 secondary space quantity for the auxiliary dataset. Can be one tenth of pq10.

card (22)

recfm10 Must be VS or VBS.

blk10 see card (20)

card (23)

types see card (19).

card (24)

pq11 primary space quantity for the auxiliary dataset. Recommended values see 2.2.5.

card (27)

types see card (19).

card (28)

pq12 primary space quantity for the auxiliary dataset. Recommended values see 2.2.5.

card (31)

These cards must only be given, if ITAPE is equal 1 (see 2.2.6 namelist INPUT). If the data written on FT15F001 should be punched, the correct parameters for the puncher must be given. If the data should be written on a tape or a disk or something like that, also the correct parameters must be given. See 2.2.5 for DCB-parameters.

card (32)

control input see 2.2.6

GFK-users can use the local procedures and must use additionally ASP control cards. Complete jobs to run BRIGITTE at the GFK computers are listed in 2.2.10.

2.2.10 Examples (input and JCL) for the use of the program

In this chapter three examples will be given for the use of the program. All these examples contain the JCL-cards and ASP-cards, which are necessary to run the jobs on the GFK computer installation. Because the installation and the systems can change, the cards are only valid for the existing installation and the operating systems at the time of writing this report.

In all three examples, it is assumed, that the ENDF/B-data can be found on a magnetic tape and are written with MODE=3 (card image standard arrangement see /3/ chapter R-2.2.3). The specifications of the tape, where the ENDF/B-data to be converted can be found, must be asked at the author of this report or the local ENDF/B-manager.

Example 1

We assume, that Cm-244 is to be translated. The KEDAK-library is created in the job. A maximal output is wished, and negative σ_t and σ_e cross section values in the resolved resonance region will be deleted.

```
(1)  //      (job card information),
(2)  //    REGION=300K,TIME=3
(3)  /*MAIN  LINES=2
(4)  /*SETUP  DEVICE=TAPE9,ID=(ididid,NORING,,SL)
(5)  /*FORMAT  PR,DDNAME=FT06F001,OVFL=ON
(6)  //  EXEC  FHG,NAME=BRIGITTE
(7)  //G.STEPLIB  DD  UNIT=3330,VOL=SER=KAPROS,DSN=STEINPDS,DISP=SHR
(8)  //G.FT01F001  DD  UNIT=uuuu,VOL=SER=volvol,
(9)  //  SPACE=(3520,(200)),DISP=(NEW,KEEP),
(10) //  DSN=dsndsn,
(11) //  DCB=(RECFM=F,BLKSIZE=3520,DSORG=DA,LRECL=3520)
(12) //G.FT09F001  DD  UNIT=TAPE9,VOL=SER=ididid,
```

```
(13)   //  DSN=dsname,DISP=(OLD,PASS),
(14)   //  LABEL=(n,SL,,IN),
(15)   //  DCB=(RECFM=FB,LRECL=80,BLKSIZE=3520,DEN=2)
(16)   //G.FT10F001  DD  UNIT=SYSDA,SPACE=(3520,(30,5)),
(17)   //  DISP=(NEW,DELETE,DELETE),
(18)   //  DCB=(RECFM=VS,BLKSIZE=3520)
(19)   //G.FT11F001  DD  UNIT=SYSDA,SPACE=(1600,(400),,CONTIG),
(20)   //  DCB=(DSORG=DA,RECFM=F,BLKSIZE=1600)
(21)   //G.FT12F001  DD  UNIT=SYSDA,SPACE=(1600,(400),,CONTIG),
(22)   //  DCB=(DSORG=DA,RECFM=F,BLKSIZE=1600)
(23)   //G.SYSIN  DD  *
(24)    &SPACE  &END
(25)    &INPUT   MATP=mmmm,NAMKDK='CM244EN4',
(26)     MATKDK=o960244,EB=6.799E+6,
(27)     LEDITS=1,LABEL=111,ELINEG=1,
(28)    &END
(29)    &LIB  IFIRST=1,&END
(30)   /*
(31)   //
```

Remarks to the cards:

card (1)

(job card information) These data are jobname, JOB-card identification, accounting information, and name of the user.

card (4)

ididid the tape name must be asked at the local ENDF/B-manager.

card (5)

By this card the maximal printed lines for a page is limited to 66. The next line(s) is (are) printed on a new page.

card (8)

uuuu volvol	}	unit type and volume name of a disk storage.

card (10)

dsndsn	this name can be choosen by the user. The translated data are written on this data set. (Never use a KEDAK-library, which has been created by or processed with KEMA. The library must have been created by BRIGITTE itself. See 2.2.4)

card (12)

ididid	see card (4)

card (13)

dsname	this name must be asked at the local ENDF/B-manager.

card (14)

n	the label number must be asked at the local ENDF/B-manager.

card (25)

mmmm	the ENDF/B material number must be asked at the local ENDF/B-manager.

card (27)

lll	the label identification, where the material mmmm can be found, must be asked at the local ENDF/B-manager.

When writing the namelist input cards, remember the rules listed in 2.2.6.

Example 2

Np-237 is to be translated. The result is to be inserted in an existing KEDAK library, which must have been created by BRIGITTE in a preceding job (see 2.2.4). If this library contains Np-237-data, these data are to be deleted. A maximal output is wished; negative σ_t and σ_e cross section values in the resolved resonance region are to be deleted.

```
(1)   //       (job card information),
(2)   //     REGION=700K,TIME=17
(3)   /*MAIN  LINES=4
(4)   /*SETUP  DEVICE=TAPE9,ID=(ididid,NORING,,SL)
(5)   /*FORMAT  PR,DDNAME=FT06F001,OVFL=ON
(6)   //  EXEC  FHG,NAME=BRIGITTE
(7)   //G.STEPLIB  DD  UNIT=3330,VOL=SER=KAPROS,DSN=STEINPDS,DISP=SHR
(8)   //G.FT01F001  DD  UNIT=uuuu,VOL=SER=volvol,
(9)   //  SPACE=(3520,(200)),DISP=(OLD,KEEP),
(10)  //  DSN=dsndsn,
(11)  //  DCB=(RECFM=F,BLKSIZE=3520,DSORG=DA,LRECL=3520)
(12)  //G.FT09F001  DD  UNIT=TAPE9,VOL=SER=ididid,
(13)  //  DSN=dsname,DISP=(OLD,PASS),
(14)  //  LABEL=(n,SL,,IN),
(15)  //  DCB=(RECFM=FB,LRECL=80,BLKSIZE=3520,DEN=2)
(16)  //G.FT10F001  DD  UNIT=SYSDA,SPACE=(3520,(30,5)),
(17)  //  DISP=(NEW,DELETE,DELETE),
(18)  //  DCB=(RECFM=VS,BLKSIZE=3520)
(19)  //G.FT11F001  DD  UNIT=SYSDA,SPACE=(1600,(800),,CONTIG),
(20)  //  DCB=(DSORG=DA,RECFM=F,BLKSIZE=1600)
(21)  //G.FT12F001  DD  UNIT=SYSDA,SPACE=(1600,(800),,CONTIG),
(22)  //  DCB=(DSORG=DA,RECFM=F,BLKSIZE=1600)
```

```
(23)   //G.SYSIN  DD  *
(24)    &SPACE  &END
(25)    &INPUT  MATP=mmmm,NAMKDK='NP237EN4',
(26)     MATKDK=0920237,EB=6.619E+6,
(27)     LEDITS=1,LABEL=111,ELINEG=1,
(28)    &END
(29)    &LIB  IFIRST=0,IMOD=1,&END
(30)   /*
(31)   //
```

Remarks to the cards:

See remarks for example 1.

Example 3

Mo (Molybdenum) is to be translated. The result is to be inserted in an existing BRIGITTE created KEDAK-library (see 2.2.4). If this library contains Mo-data, the converted data must not be inserted in the library. A maximal output is wished; negative σ_t and σ_e cross section values in the resolved resonance region are to be deleted. The ENDF/B-material contains in MF=2 (resonance parameters) information for 7 different isotopes. The namelist &ISO must be present. The default value of NIX (500) is to be increased on 600.

```
(1)   //      (job card information),
(2)   //    REGION=300K,TIME=2
(3)   /*MAIN  LINES=2
(4)   /*SETUP  DEVICE=TAPE9,ID=(ididid,NORING,,SL)
(5)   /*FORMAT  PR,DDNAME=FT06F001,OVFL=ON
(6)   //  EXEC  FHG,NAME=BRIGITTE
(7)   //G.STEPLIB  DD  UNIT=3330,VOL=SER=KAPROS,DSN=STEINPDS,DISP=SHR
```

```
(8)   //G.FT01F001  DD  UNIT=uuuu,VOL=SER=volvol,
(9)   //  SPACE=(3520,(200)),DISP=(OLD,KEEP),
(10)  //  DSN=dsndsn,
(11)  //  DCB=(RECFM=F,BLKSIZE=3520,DSORG=DA,LRECL=3520)
(12)  //G.FT09F001  DD  UNIT=TAPE9,VOL=SER=ididid,
(13)  //  DSN=dsname,DISP=(OLD,PASS),
(14)  //  LABEL=(n,SL,,IN),
(15)  //  DCB=(RECFM=FB,LRECL=80,BLKSIZE=3520,DEN=2)
(16)  //G.FT10F001  DD  UNIT=SYSDA,SPACE=(3520,(30,5)),
(17)  //  DISP=(NEW,DELETE,DELETE),
(18)  //  DCB=(RECFM=VS,BLKSIZE=3520)
(19)  //G.FT11F001  DD  UNIT=SYSDA,SPACE=(1600,(400),,CONTIG),
(20)  //  DCB=(DSORG=DA,RECFM=F,BLKSIZE=1600)
(21)  //G.FT12F001  DD UNIT=SYSDA,SPACE=(1600,(400),,CONTIG),
(22)  //  DCB=(DSORG=DA,RECFM=F,BLKSIZE=1600)
(23)  //G.SYSIN  DD  *
(24)   &SPACE  N1X=600,&END
(25)   &INPUT  MATP=mmmm,NAMKDK='MO   EN4',
(26)    MATKDK=0420000,EB=8.6424E+6,NISO=7
(27)    LEDITS=1,LABEL=111,ELINEG=1,
(28)   &END
(29)   &ISO  NAMISO='MO 92EN4','MO 94EN4',
(30)                'MO 95EN4','MO 96EN4',
(31)                'MO 97EN4','MO 98EN4',
(32)                'MO100EN4',
(33)         MATISO=0420092,0420094,0420095
(34)                0420096,0420097,0420098,
(35)                0420100,
```

```
(36)     EBISO=12.692.E+6,9.6722E+6,7.3751E+6,
(37)           9.1542E+6,6.8161E+6,8.6424E+6,
(38)           8.301E+6,&END
(39)     &LIB  IFIRST=0,IMOD=0,&END
(40)    /*
(41)    //
```

Remarks to the cards:

See example 1.

2.3 Programmer's guide

The following sections have mainly been written for programmers. But the section 2.3.1 is very useful for all those users, desiring to get a general survey about the program. It is also possible, that problems, which may arise have with the program, may be solved by reading the corresponding section in this part of the description.

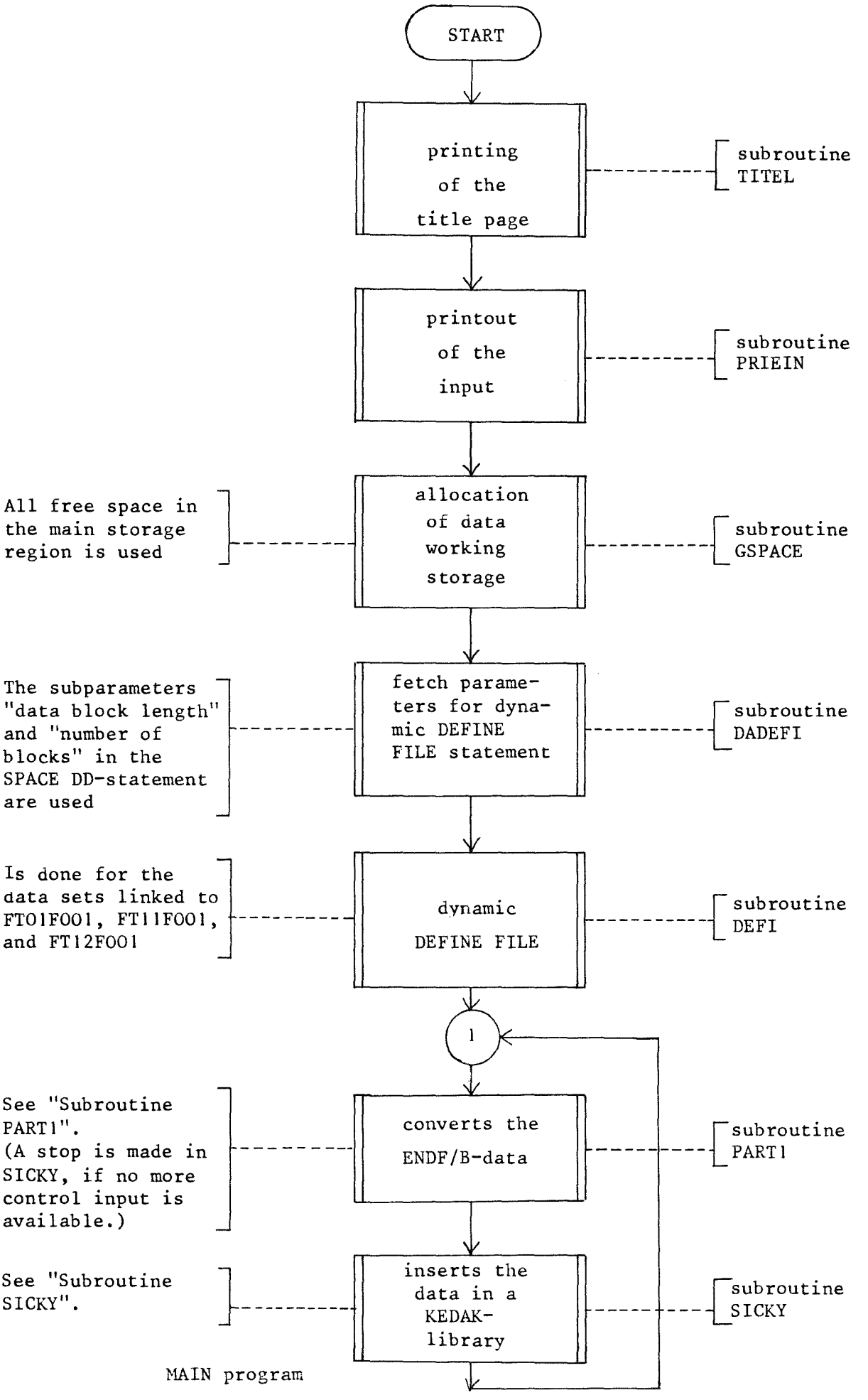
The contents of the following pages can best be seen in the table of contents. The sections containing general aspects of the program (2.3.1 until 2.3.4) are followed by a detailed description of each subroutine resp. function (2.3.5.1 until 2.3.5.99). The purpose of the subroutines and functions respectively can be found in headings preceding the groups of subroutines and functions respectively.

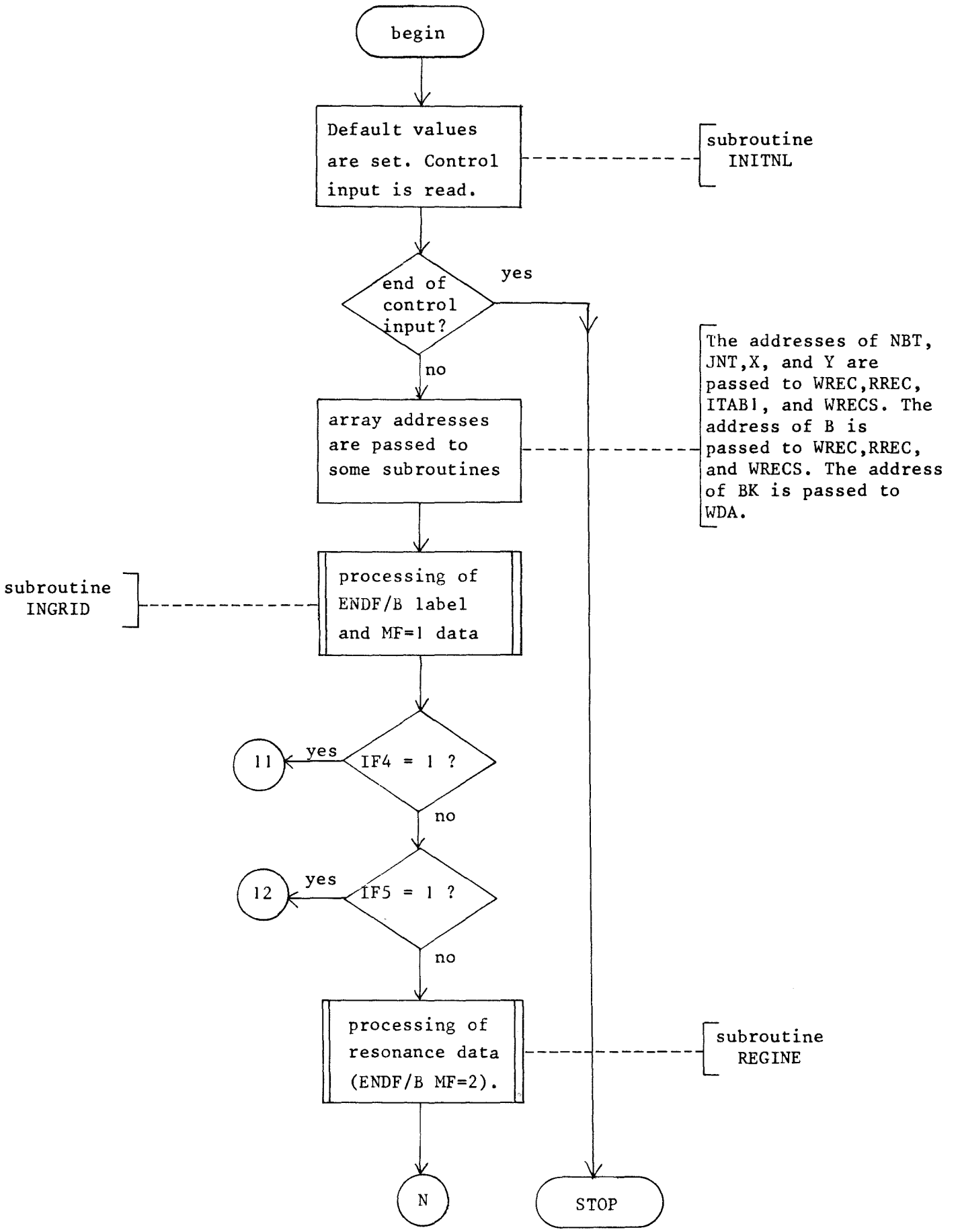
2.3.1 General program flow

To get a survey about the program, a block diagram of the program and data flow will be given.

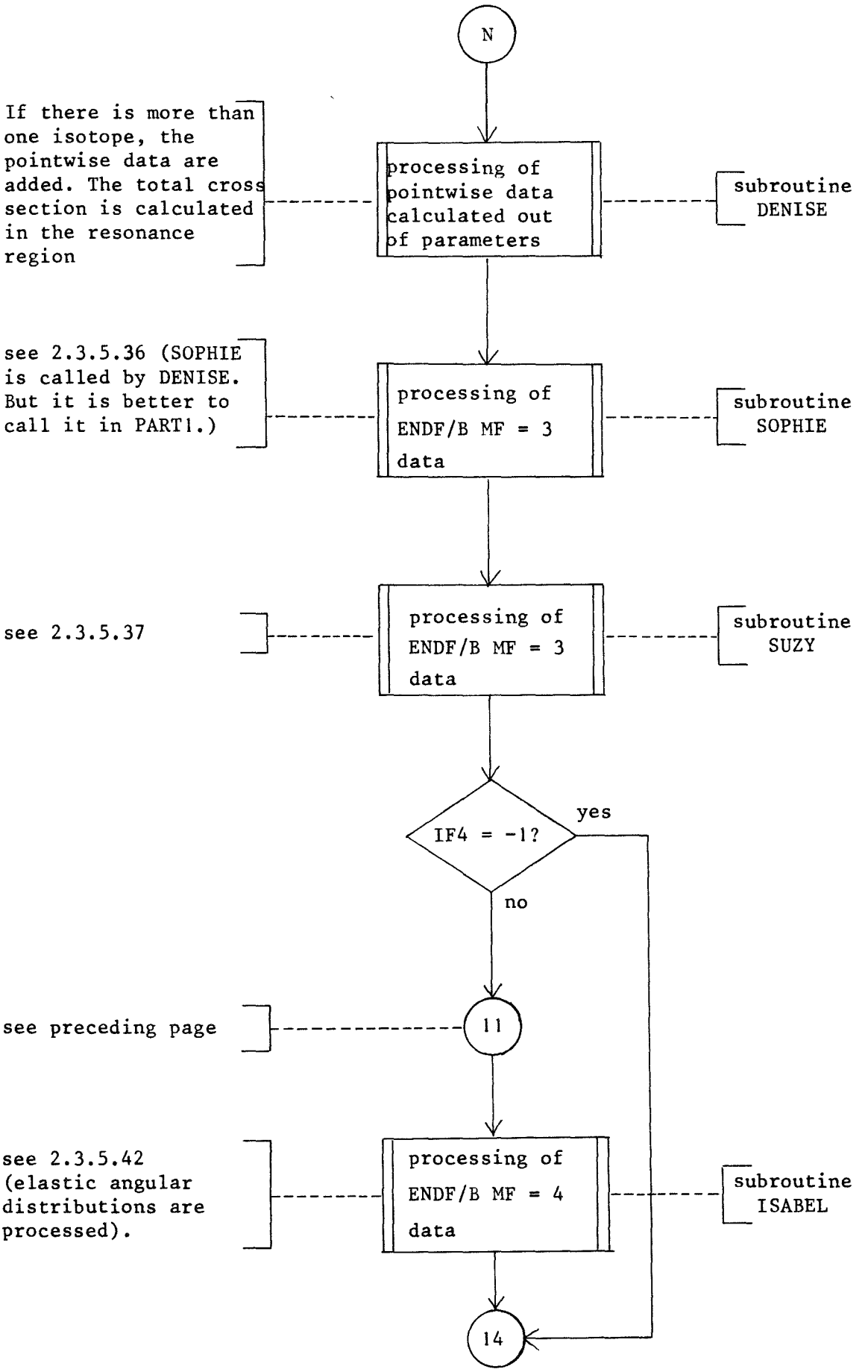
There are eight tables describing in a rough manner the program flow followed by two tables describing the data flow. The symbols used in the tables are those of DIN 66001 (see /12/). The first table "MAIN program" describes the MAIN-program. Two programs called by the MAIN-program will be described in the following tables. These are the subroutine PART1 and the subroutine SICKY. The subroutine PART1 resp. subroutines called by it does all the translation work, whereas the subroutine SICKY resp. subroutines called by it will insert the translated data into the KEDAK-dataset.

As one can see in the program flow tables, the ENDF/B-data are processed in the sequence of their file (MF)-numbers. This fact should be considered, when looking at the two data flow tables. Remember that the KEDAK-library on file 1 must have been initialized by BRIGITTE itself (see 2.2.4).

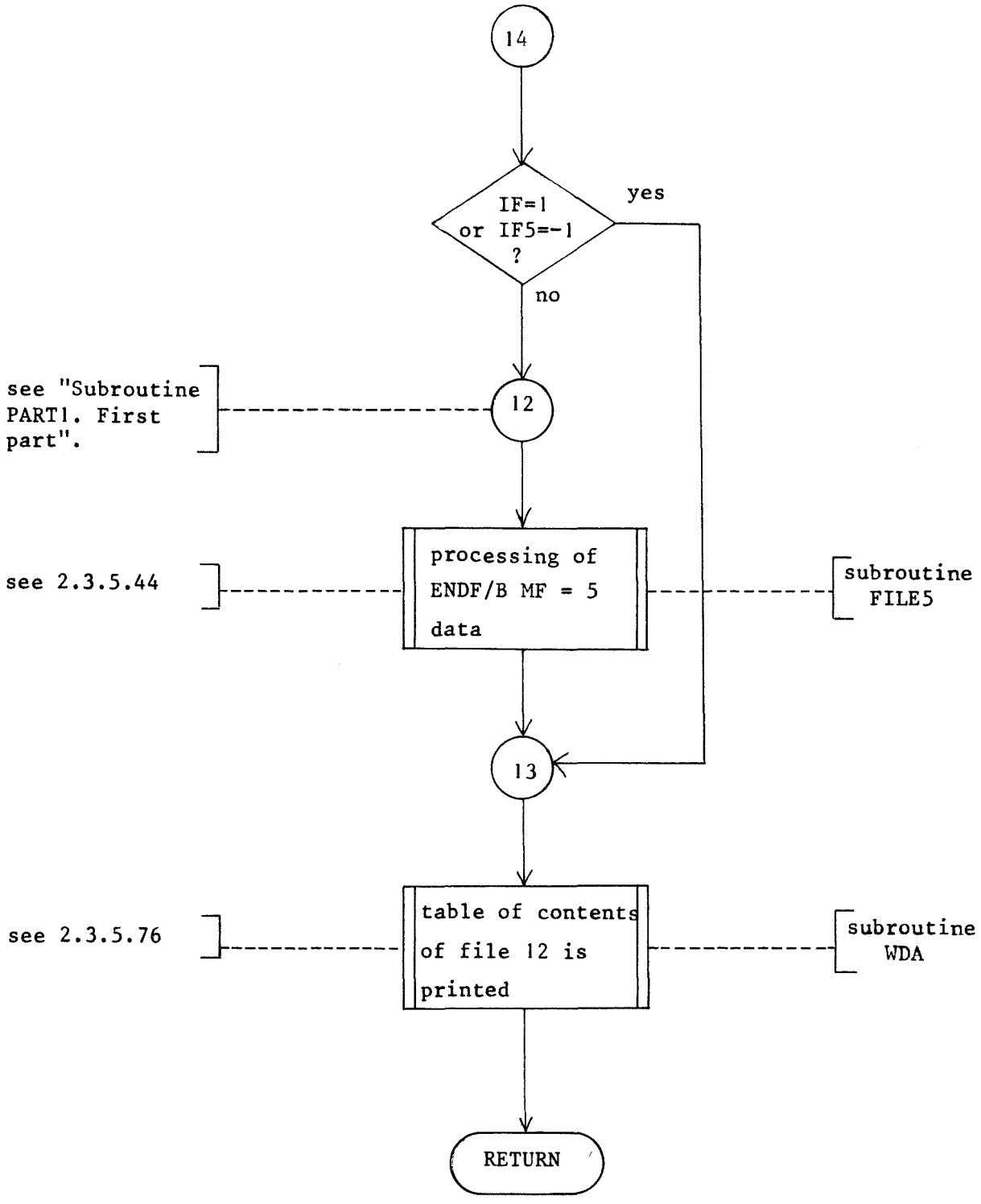

START
printing of the title page
subroutine TITEL
printout of the input
subroutine PRIEIN
All free space in the main storage region is used
allocation of data working storage
subroutine GSPACE
The subparameters "data block length" and "number of blocks" in the SPACE DD-statement are used
fetch parameters for dynamic DEFINE FILE statement
subroutine DADEFI
Is done for the data sets linked to FT01F001, FT11F001, and FT12F001
dynamic DEFINE FILE
subroutine DEFI
1
See "Subroutine PART1". (A stop is made in SICKY, if no more control input is available.)
converts the ENDF/B-data
subroutine PART1
See "Subroutine SICKY".
inserts the data in a KEDAK-library
subroutine SICKY
MAIN program



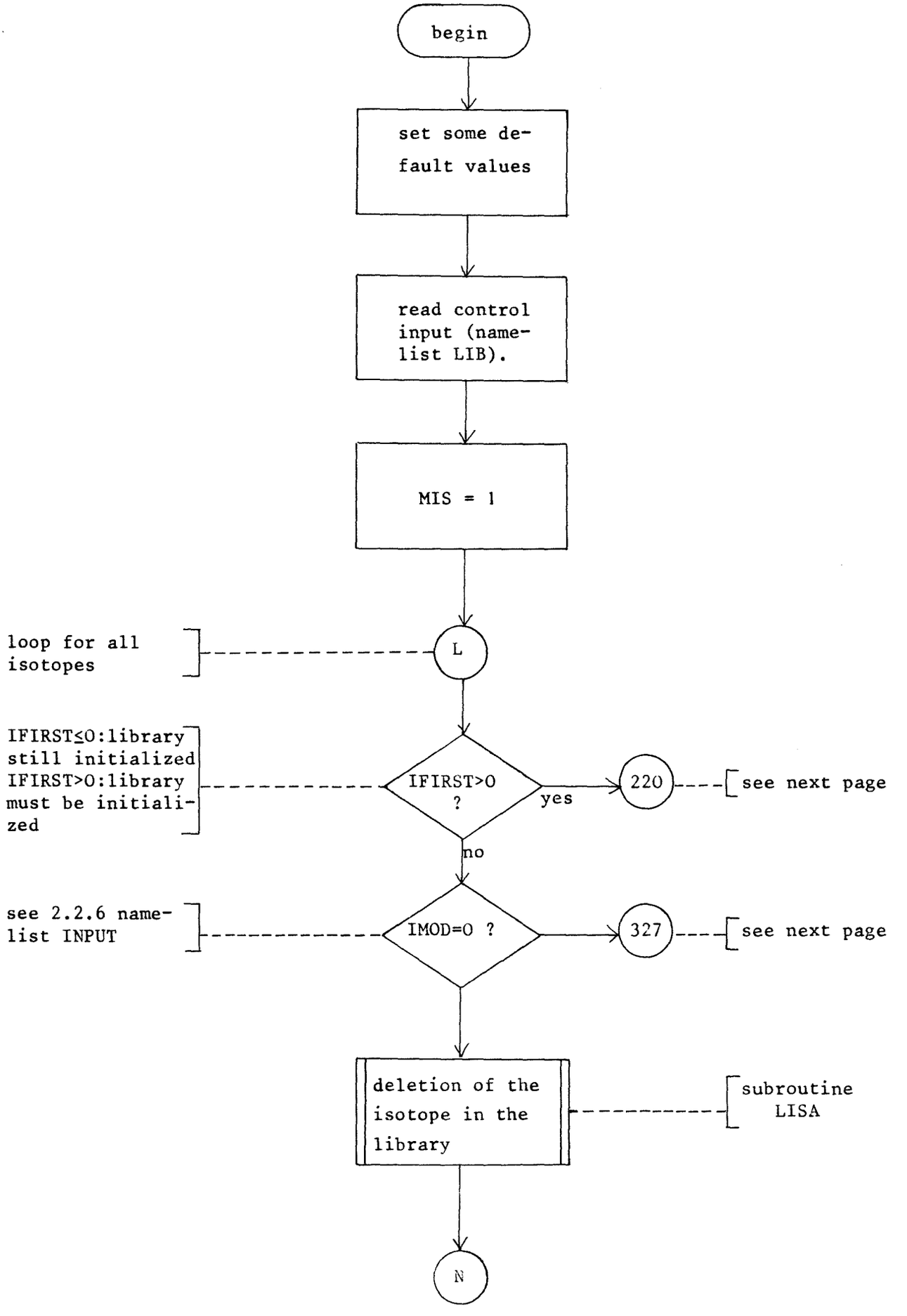
Subroutine PART1. Fist part (continued on next page).



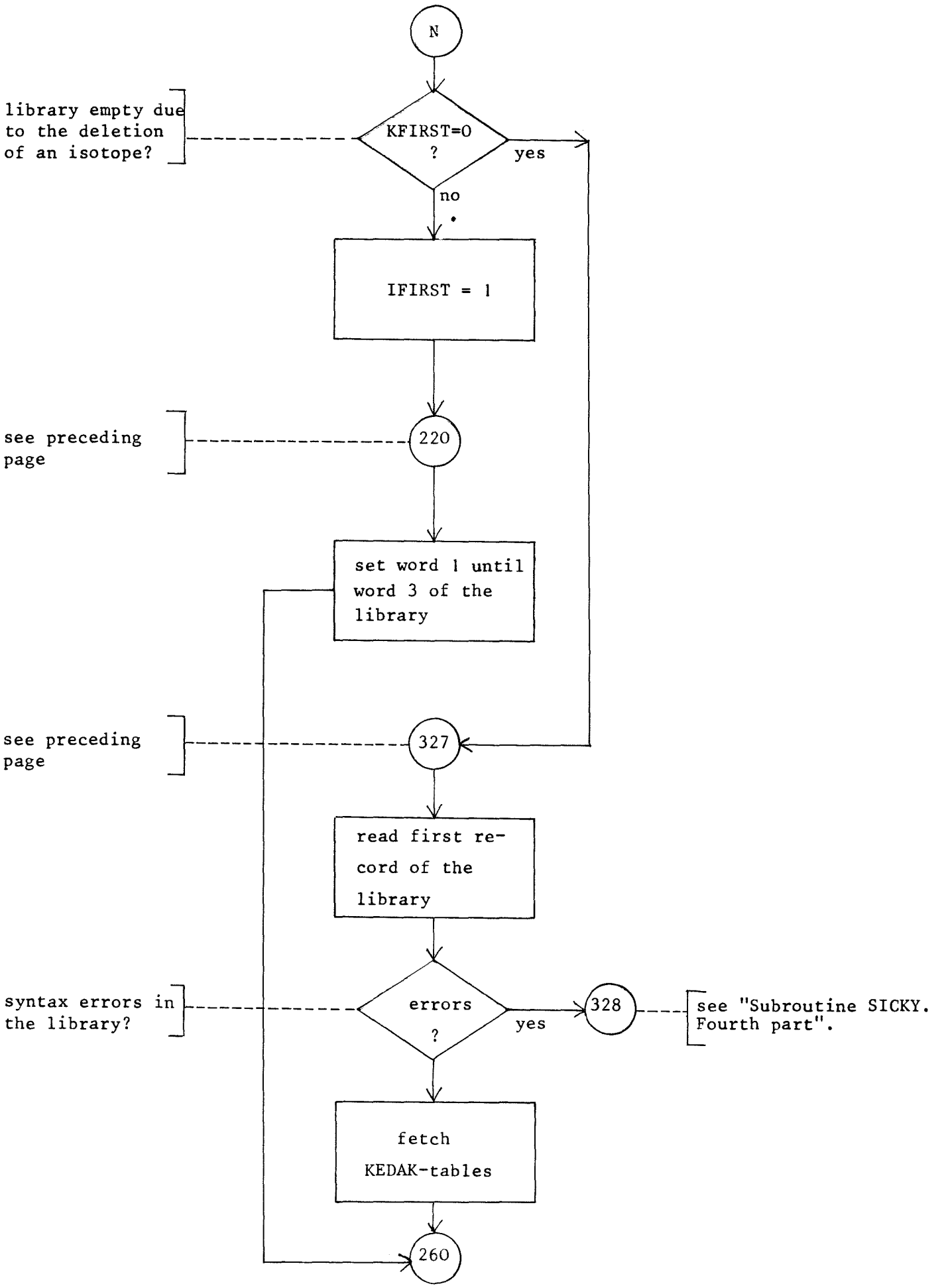
Subroutine PART1. Second part (continuation of preceding page; continued on next page).



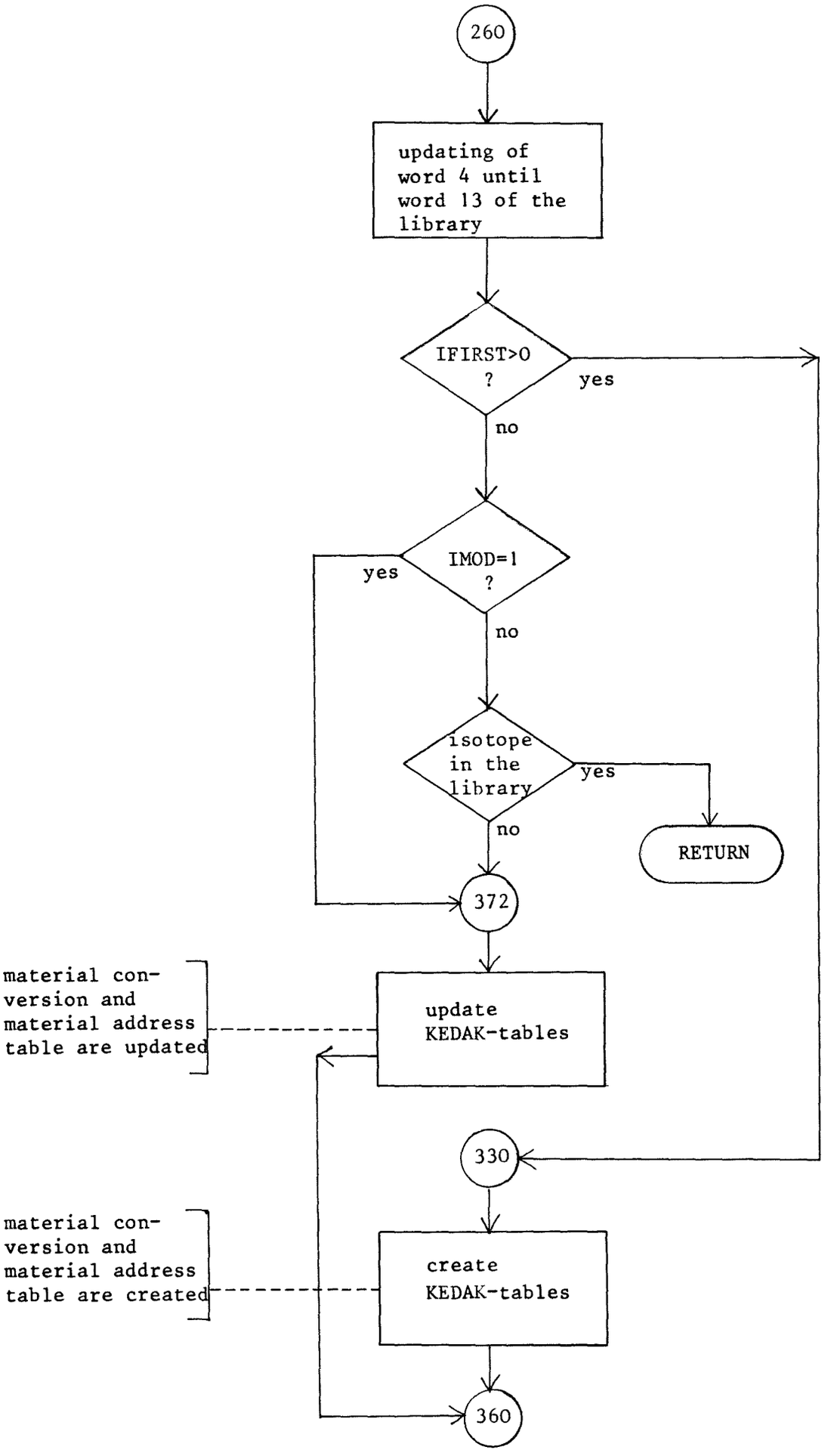
Subroutine PART1. Third part (continuation of preceding page).



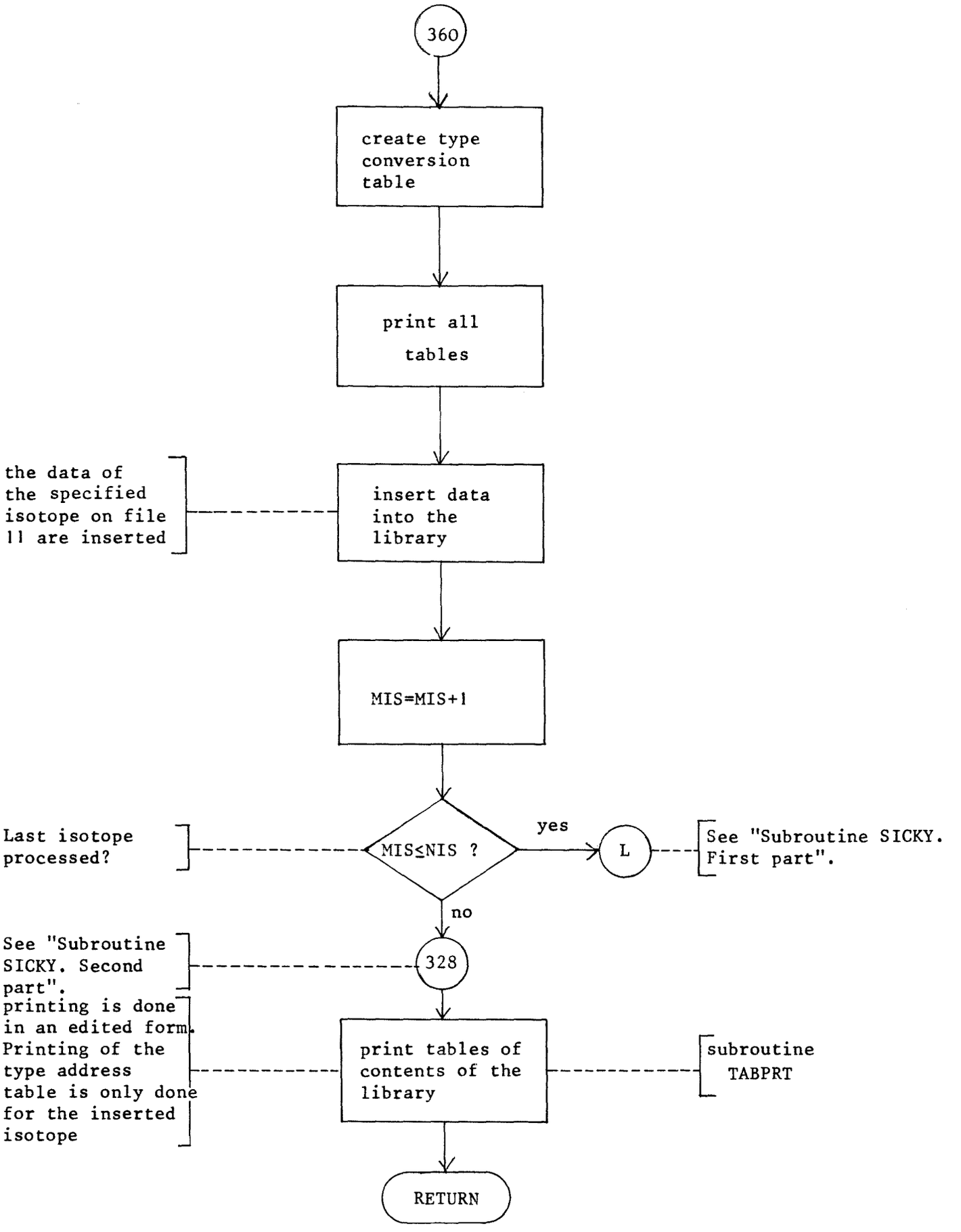
Subroutine SICKY. First part (continued on next page).



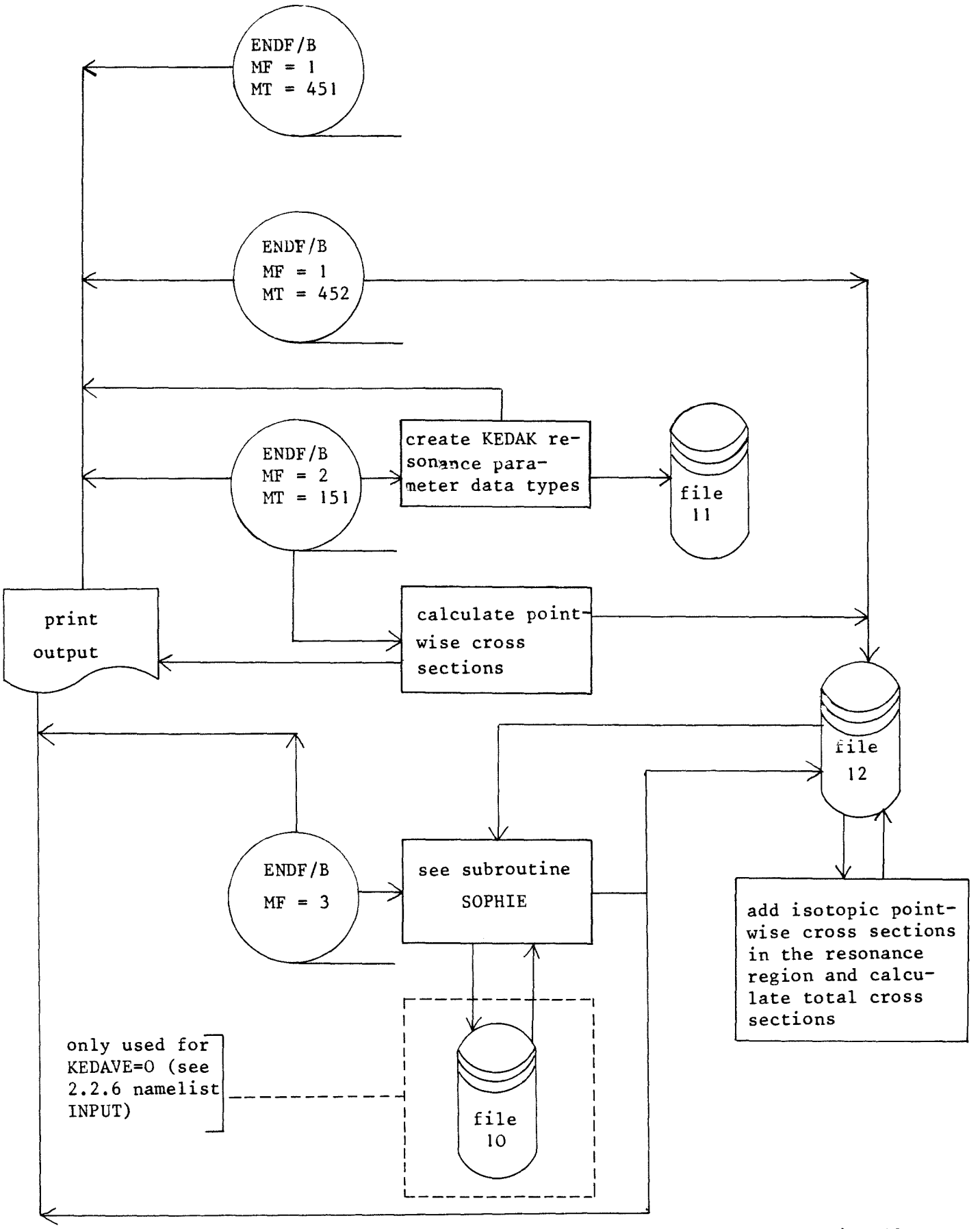
Subroutine SICKY.Second part (continuation from preceding page; continued on next page).



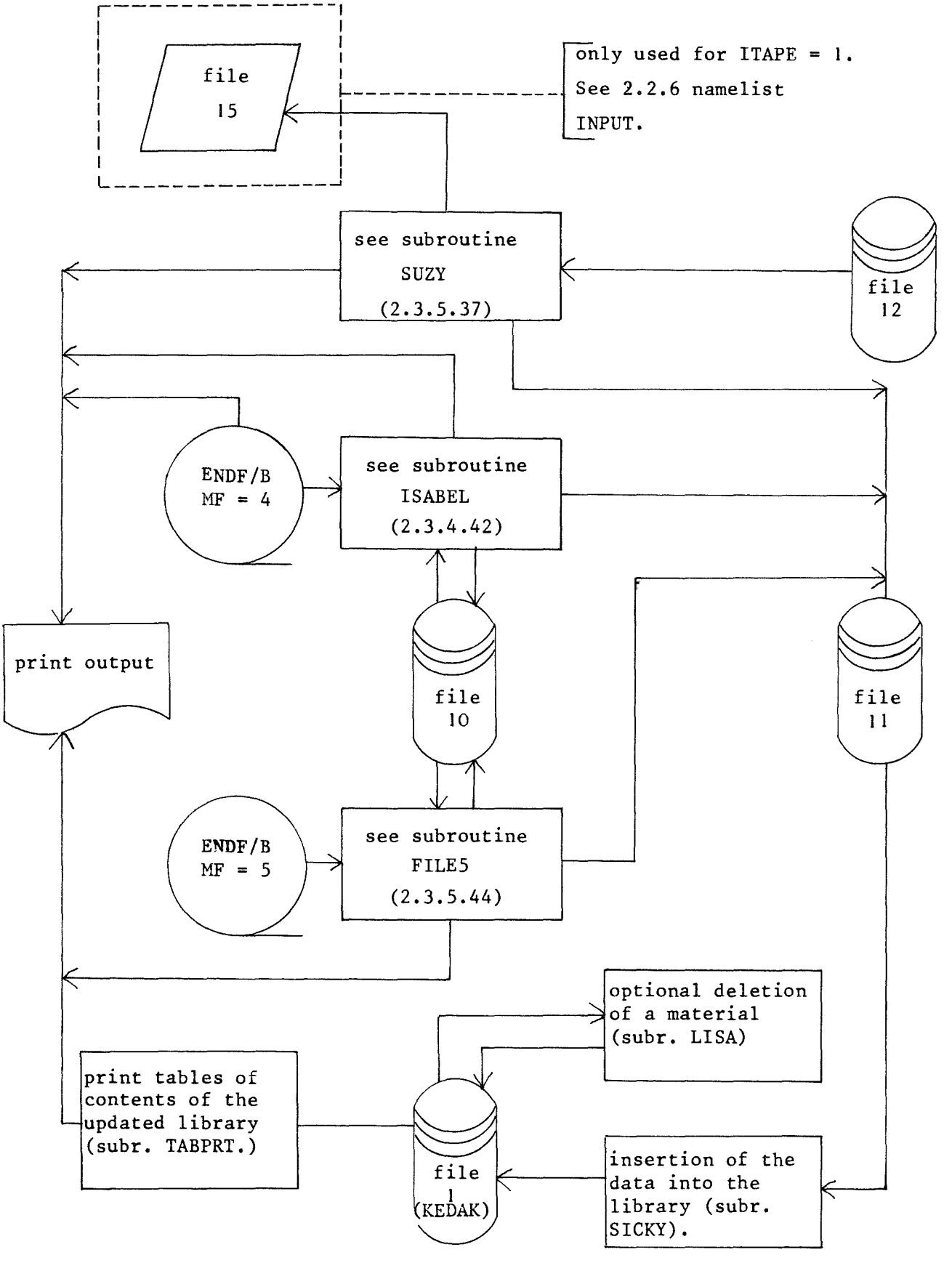
Subroutine SICKY. Third part (continuation from preceding page; continued on next page)



Subroutine SICKY. Fourth part (continuation of the preceding page).



Data flow in BRIGITTE. First part (continued on the next page). Note that file 10 is not obliged to reside on a disk storage.



Data flow in BRIGITTE. Second part (continuation of the preceding page). See note on the preceding page.

2.3.2 Overlay structure

BRIGITTE uses a two region overlay structure (see IBM System/370 literature). That means, the second region starts at the end of the largest segment of the first region. This two region structure enables all subroutines of the first region, to call any subroutine of the second region.

First a list of the used overlay control cards for the IBM linkage editor will be given. Second a graph will illustrate the overlay structure. To differ between subroutines and common blocks in this graph, capital letters have been used for subroutine names and small letters have been used for common block names.

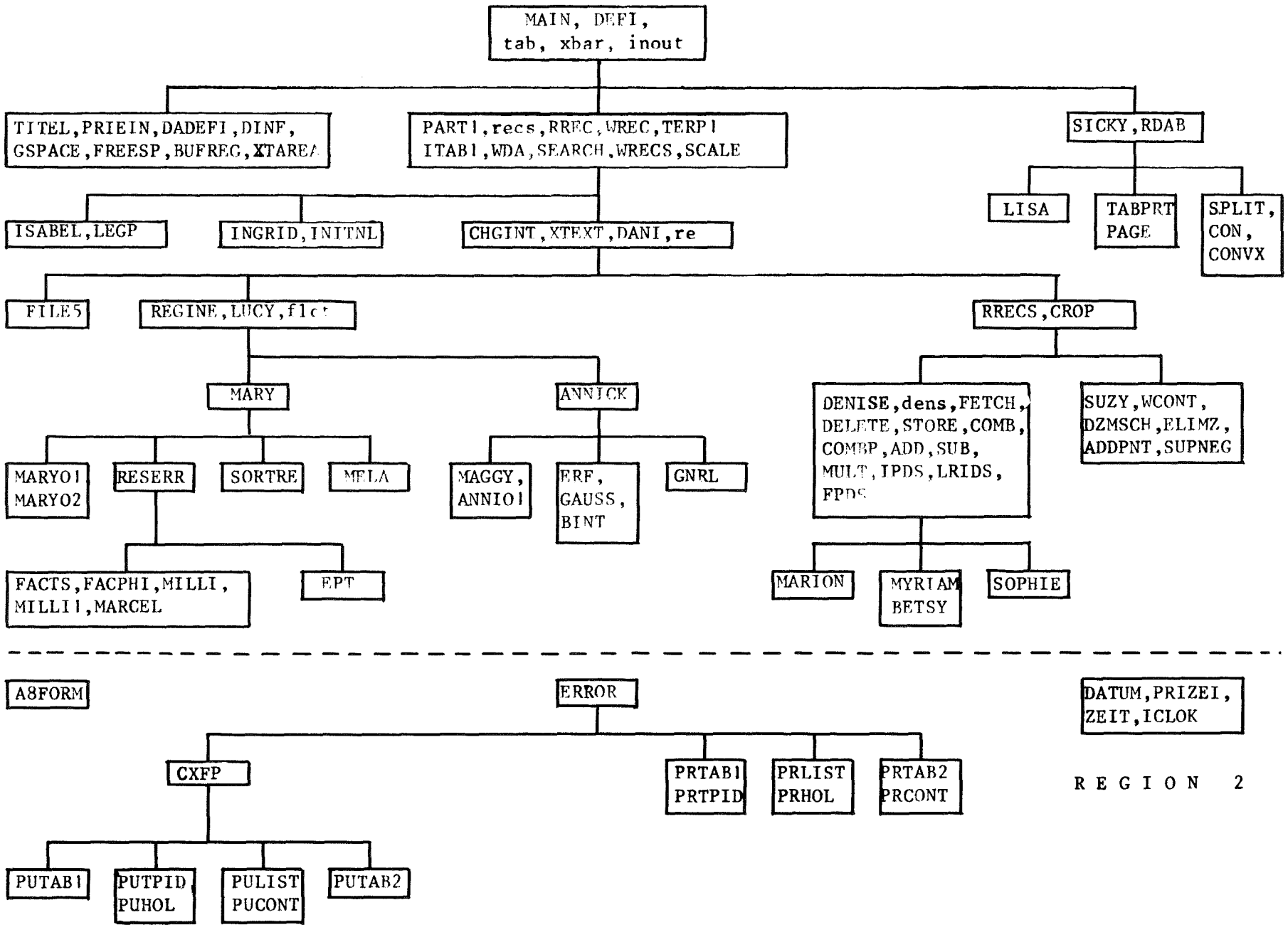
Overlay control cards:

```
OVERLAY  A
INSERT   PRIEIN,DADEFI,DINF,GSPACE,FREESP
INSERT   BUFREG,XTAREA,TITEL
OVERLAY  A
INSERT   SICKY,RDAB
OVERLAY  C
INSERT   SPLIT,CON,CONVX
OVERLAY  C
INSERT   TABPRT,PAGE
OVERLAY  C
INSERT   LISA
OVERLAY  A
INSERT   PART1,RECS,RREC,WREC,TERP1,ITAB1
INSERT   WDA,SEARCH,WRECS,SCALE
OVERLAY  D
INSERT   ISABEL,LEGP
OVERLAY  D
INSERT   INGRID,INITNL
OVERLAY  D
INSERT   CHGINT,XTEXT,DANI,RE
OVERLAY  F
INSERT   FILE5
```

```
OVERLAY  F
INSERT  REGINE,LUCY,FLCT
OVERLAY  G
INSERT  MARY
OVERLAY  G1
INSERT  MARYO1,MARYO2
OVERLAY  G1
INSERT  SORTRE
OVERLAY  G1
INSERT  RESERR
OVERLAY  G2
INSERT  EPT
OVERLAY  G2
INSERT  FACTS,FACPHI,MILLI,MILLI1,MARCEL
OVERLAY  G1
INSERT  MELA
OVERLAY  G
INSERT  ANNICK
OVERLAY  H
INSERT  MAGGY,ANNIO1
OVERLAY  H
INSERT  ERF,GAUSS,BINT
OVERLAY  H
INSERT  GNRL
OVERLAY  F
INSERT  RRECS,CROP
OVERLAY  I
INSERT  SUZY,WCONT,DZMSCH,ELIMZ,ADDPNT,SUPNEG
OVERLAY  I
INSERT  DENISE,DENS,FETCH,DELETE,STORE,COMB
INSERT  COMBP,ADD,SUB,MULT,IPDS,LRIDS,FPDS
OVERLAY  K
INSERT  MARION
OVERLAY  K
INSERT  MYRIAM,BETSY
OVERLAY  K
INSERT  SOPHIE
OVERLAY  R(REGION)
INSERT  ERROR
```

```
OVERLAY  R1
INSERT   CXFP
OVERLAY  R2
INSERT   PUTAB1
OVERLAY  R2
INSERT   PUTPID,PUHOL
OVERLAY  R2
INSERT   PULIST,PUCONT
OVERLAY  R2
INSERT   PUTAB2
OVERLAY  R1
INSERT   PRTAB1,PRTPID
OVERLAY  R1
INSERT   PRLIST,PRHOL
OVERLAY  R1
INSERT   PRTAB2,PRCONT
OVERLAY  R
INSERT   A8FORM
OVERLAY  R
INSERT   DATUM,PRIZEI,ZEIT,ICLOK
```

Overlay structure of BRIGITTE



2.3.3 Organisation of data storage

Nearly all arrays used in the program are variably dimensioned. Because there are no aids in FORTRAN to make a region dependant dimensioning of arrays, ASSEMBLER subroutines must be used.

The FORTRAN-subroutine GSPACE (see 2.3.5.5) called by the main program, first evaluates the free region by using the ASSEMBLER subroutine FREESP (see 2.3.5.95). The program only continues, if this region is greater or equal 10 k-byte. Now the number of bytes needed for buffers (see subroutine BUFREG 2.3.5.6) and 6 k-byte for system use are subtracted from the total free region, which is returned by the subroutine FREESP. The remaining region is used for data arrays. The subdivision of this region in the individual arrays is shown in the following graph. To get access to this region the ASSEMBLER subroutine XTAREA (see 2.3.5.93) is used. This routine returns the absolute address of the beginning of the data storage, after having assigned ownership of this storage to the active job, by using the ASSEMBLER makro GETMAIN. Because XTAREA also returns the absolute address of the first word of a given array, which address has been defined in the calling program and passed to the subroutine, the beginning of the data region can be expressed as an index value in this array. This is done by calculating a displacement IVER, so that F(1+IVER) is the first word of the data region, if F is the given array. This displacement is returned to the calling program, which passes to the working subroutines the address F(1+IVER) for the array BK, resp. F(1+IVER+IVAL) with corresponding IVAL for the other arrays.

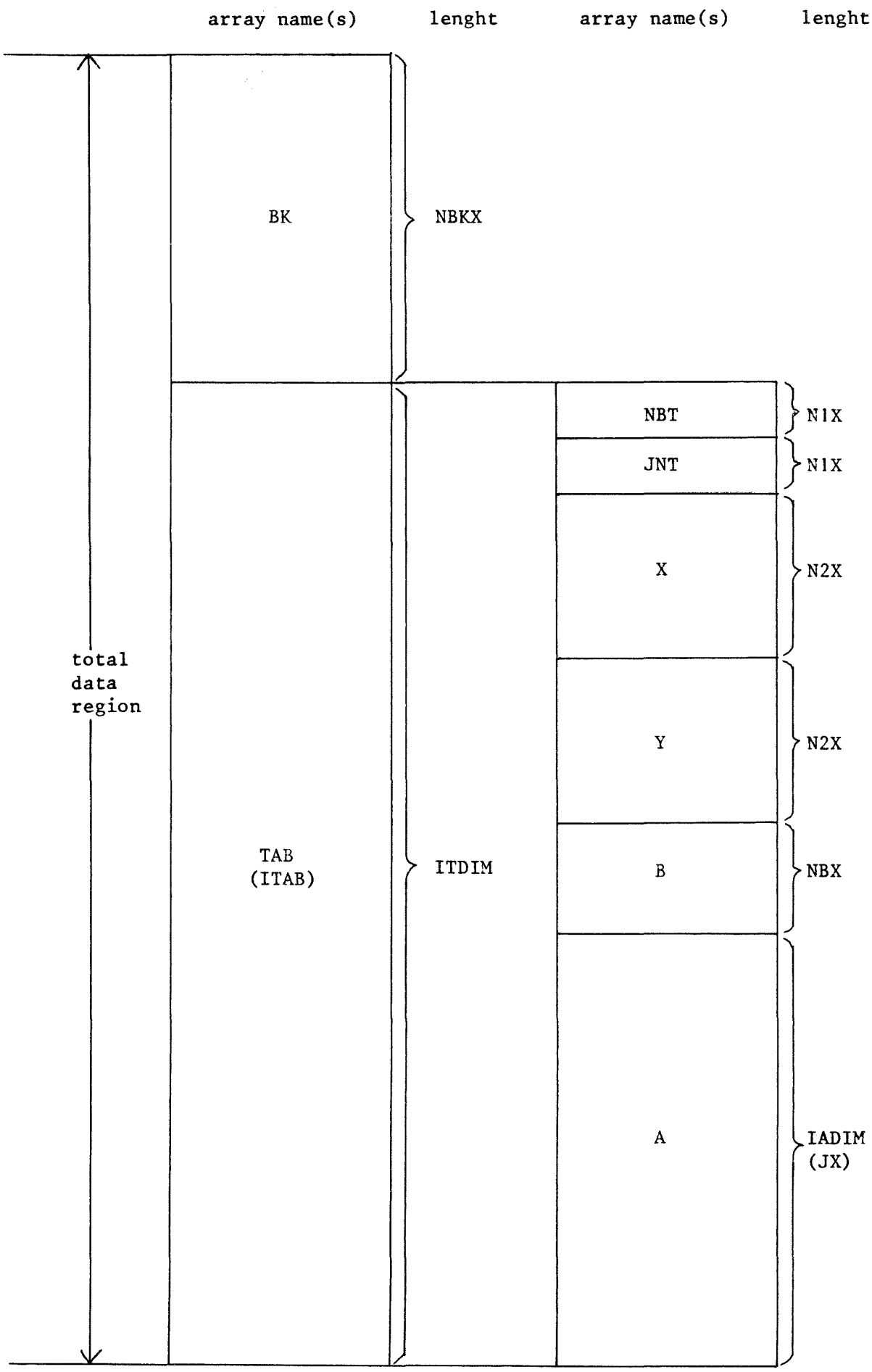
The given values for the array lenght - read or default values (see 2.2.6 namelist SPACE) - are checked by GSPACE. JX is set to IADIM, NBKX is made even, and ITDIM is set to maximum (ITDIM,2*N1X+2*N2X+NBX+IADIM).

Now NBX is set to ITDIM-IADIM-2*(N1X+N2X), and afterwards ITDIM is rounded to a multiple of 880. By using the default values listed in 2.2.6, we get the following dimensions:

NBKX = 12 000 words

ITDIM = 33 440 words

N1X = 500 words
N2X = 5000 words
NBX = 3000 words
IADIM = JX = 20000 words



Overlay of the data region.

Summing up the lenght of all arrays, a total of 46 000 words is the result.

If the extent of the data region is not large enough to contain these 46 000 words (≈ 180 k-byte), all dimensions will be shortened by multiplication with a factor KW/ISUM. KW means the extent of the existing data region, ISUM the result of summing up the lenghts of all arrays, both measured in 4-byte words. The new dimensions are once more checked and eventually changed (e.g. NBKX is made even). After the calculation of the beginning of all arrays relative to BK(1), RETURN is made.

If 1.1*ISUM is less or equal KW all dimensions will be increased by multiplication with the factor KW/ISUM. Afterwards the same is done as in the case of shortening the dimensions. All quantities calculated by GSPACE are listed in the program printout.

In the table"overlay of the data region", one can see that parts of the data region are used twice. This is naturally not done simultaneously in one subroutine but in different ones. Also some arrays are subdivided in some subroutines.

To get a survey about the usage of the arrays in the program, a table is made for each array. This tables contain in the heading the names and the lenght of the arrays. If the name of an array in a subroutine is the same, as in the heading, only the names of the subroutines, in which this array is used, are listed. Otherwise the names of the subroutines together with the names of the arrays, or the names of the subarrays and the maximal dimensions, are tabulated. If a subroutine may use - dependant on the argument list in the call statement - different arrays, it is not contained in the following tabulations.

Some of the arrays are only used for defined data. NBT,JNT,X, and Y contain - with a few exceptions - only TAB1-records (see /3/ chapter R-2.3.3). NBT and JNT contain only interpolation tables. The array B is used to store LIST-records (see /3/ chapter R-2.3.2) read from ENDF/B-tapes. In some cases it is also used as an auxiliary array.

name of the array: BK	lenght: NBKX
subroutine name	name of the array in the subroutine, or name and dimensions of subarrays
PART1	BK
REGINE	BK
MARY	BK
MARY01	BK
LUCY	BK
MILLI	NBTN(N1X),JNTN(N1X),NBTG(N1X),JNTG(N1X),NBTF(N1X), JNTF(N1X)
MILLI1	one of the arrays NBTN,JNTN, or NBTG,JNTG, or NBTF, JNTF
MELA	IBK
ANNICK	BK
ANNI01	BK
DENISE	BK
SOPHIE	XIN(NBKX/2),YIN(NBKX/2)
SUZY	BK,A,TAB,ITAB
ISABEL	ITAB,TAB,CLEG
FILE5	BK,IBK
WDA	BK
RDAB	BK
SICKY	BK

name of the array: NBT	lenght: N1X
used in the subroutines (with the name NBT): PART1,REGINE,MARY,MELA,ANNICK,ANNIO1,DENISE,SOPHIE,SUZY,ELIMZ,ADDPNT, DZMSCH,SUPNEG,ISABEL,FILE5,RREC,WREC,PRTAB1,PRTAB2,PUTAB1,PUTAB2,COMB, COMBP,ITAB1,CHGINT,CROP,WRECS,RRECS,STORE,FETCH	
name of the array: JNT	lenght: N1X
used in the subroutines (with the name JNT): PART1,REGINE,MARY,MELY,ANNICK,ANNIO1,DENISE,SOPHIE,SUZY,ELIMZ,ADDPNT, DZMSCH,SUPNEG,ISABEL,FILE5,RREC,WREC,PRTAB1,PRTAB2,PUTAB1,PUTAB2,COMB, COMBP,ITAB1,CHGINT,CROP,WRECS,RRECS,STORE,FETCH	
name of the array: X	lenght: N2X
used in the subroutines (with the name X): PART1,REGINE,MARY[1],EPT[1],MARYO2[1],MILLI[1],ANNICK[2],ANNIO1[2],MAGGY, DENISE,SOPHIE,SUZY,ELIMZ,ADDPNT,DZMSCH,SUPNEG,ISABEL,FILE5,RREC,WREC, PRTAB1,PUTAB1,COMB,COMBP,ITAB1,CHGINT,CROP,WRECS,RRECS,STORE,FETCH 1) name of the array: ENS 2) name of the array: BK2	
name of the array: Y	lenght: N2X
used in the subroutines (with the name Y): PART1,REGINE,MARY[1],MELA[1],ANNICK,ANNIO1,MAGGY,DENISE,MYRIAM,BETSY, SOPHIE,SUZY,ELIMZ,ADDPNT,DZMSCH,SUPNEG,ISABEL,FILE5,RREC,WREC,PRTAB1, PUTAB1,COMB,COMBP,ITAB1,CHGINT,CROP,WRECS,RRECS,STORE,FETCH 1) name of the array: SIGS	
name of the array: B	lenght: NBX
used in the subroutines (with the name B): PART1,REGINE,MARY,ANNICK,DENISE,SOPHIE[1],SUZY,ISABEL,FILE5[1],RREC,WREC, PRLIST,PRHOL,PRTPID,PULIST,PUHOL,PUTPID,WRECS,RRECS,STORE,FETCH 1) name of the array: B and TEMP	

name of the array: A	lenght: IADIM
subroutine name	name of the array in the subroutine, or name and dimensions of subarrays
PART1	A
REGINE	AH
MARY	ER(NDIM),AJ(NDIM),GT(NDIM),GN(NDIM),GG(NDIM),GF(NDIM), AL(NDIM),G(NDIM),SER(NDIM),PER(NDIM),AH(JXRED),NDIM = JXRES = IADIM/200; JXRED = $\frac{19}{20}$ * IADIM
SORTE	ER(NDIM),AJ(NDIM),GT(NDIM),GN(NDIM),GG(NDIM),GF(NDIM), AL(NDIM),G(NDIM),SER(NDIM),PER(NDIM) See MARY
MARY01	ER(NDIM),AJ(NDIM),GT(NDIM),GN(NDIM),GG(NDIM),GF(NDIM), AL(NDIM),G(NDIM) See MARY
EPT	ER(NDIM),GTR(NDIM) See MARY(GTR is GT)
MARCEL	ER(NDIM),GN(NDIM),GG(NDIM),GF(NDIM),AL(NDIM),G(NDIM), SER(NDIM),PER(NDIM) See MARY
MILLI	AH
MELA	AH
ANNICK	SIGS(NPEMAX),E(NPEMAX),SIGG(NPEMAX),SIGF(NPEMAX), SIGT(NPEMAX),ESC(NPEMAX),XT(NPEMAX),YT(NPEMAX), TEMP(NPEMAX),D(6,NPEMAX),GX(6,NPEMAX),GNO(6,NPEMAX), GG(6,NPEMAX),GF(6,NPEMAX),AJ(6,NPEMAX),NPEMAX=IADIM/45
ANNIO1	SIGS(NPEMAX),E(NPEMAX),SIGG(NPEMAX),SIGF(NPEMAX), SIGT(NPEMAX),ESC(NPEMAX) See ANNICK
MAGGY	XD(NPEMAX),YD(NPEMAX) See ANNIO1. XD is E, YD is SIGS,SIGG, or SIGF
DENISE	A
ISABEL	A,ITAB2,TAB2

name of the array: A (continuation)	lenght: IADIM
subroutine name	name of the array in the subroutine, or name and dimensions of subarrays
STORE	A,LA
FETCH	A,LA
DELETE	A
FPDS	A,LA
IPDS	A,LA

name of the array: TAB	lenght: ITDIM
used in the subroutine (with the name TAB and ITAB): SICKY	

2.3.4 COMMONS used in the program

There are seven named COMMONS used by the program. The COMMON names are listed in the following in alphabetic order:

DENS
FLCT
INOUT

RE
RECS
TAB

XBAR

For each of this COMMON blocks a separate table (which can be distributed on several pages) is given. In this table you will find for each element of the COMMON the (mainly used) name of this element, the address (measured in 4*byte words, starting with 1), the dimension, the type, the number of the program(s), where the contents of this element are used, and the meaning of the element.

The abbreviations used,are:

in the column headed "dimension":

scalar	for scalar quantities
(n)	for an array with n-elements
(n,m)	for an array with n*m elements

in the column headed "type":

I4	for INTEGER*4 quantities
R4	for REAL*4 quantities
R8	for REAL*8 quantities
L4	for LOGICAL*4 quantities

in the column headed "used in program number":

n	fetched in program 2.3.5.n
n	stored (and fetched) in program 2.3.5.n

(n) in program 2.3.5.n the address of the quantity is passed to a subprogram. In this subprogram, or programs called by this subprogram, the value of the quantity is fetched.

(n) in program 2.3.5.n the address of the quantity is passed to a subprogram. In this subprogram, or programs called by this subprogram, the value of the quantity is stored (and fetched).

Only one of these four cases is listed.

If n and (n and/or (n) and/or (n)) happens, n is listed.

If (n) and (n and/or (n)) happens, (n) is listed.

If n and (n) happens, n is listed.

E.g. 15 and (15) and 15 will force the list entry 15.

Name of the COMMON block: DENS					
word number	name	dimension	type	used in programs number	meaning
1	JMT	(100)	I4	80, 81, 82, 85.	see 2.3.5.80 "problem solved"
101	JAT	(100)	I4	80, 81, 82, 85.	" "
201	JTT	(100)	I4	80, 81, 82.	" "
301	JLT	(100)	I4	80, 82.	" "
401	JNS	scalar	I4	80, 82.	" "
402	MNS	scalar	I4	80, 81, 82, 85.	" "
403	MX	scalar	I4	32, 80, 82.	lenght of the arrays JMT,JAT,JTT, and JLT (see 2.3.5.80).
Name of the COMMON block: FLCT					
1	ZYN	scalar	R4	25, 27, 29.	number of degrees of freedom in the neutron width distribution (see 2.3.5.29)
2	ZYF	scalar	R4	25, 29.	number of degrees of freedom in the fission widht distribution (see 2.3.5.29).
3	U	scalar	R4	25, 29.	used in the calculation of the Dresner factors in BINT (2.3.5.29).
4	V	scalar	R4	25, 29	used in the calculation of the Dresner factors in BINT (2.3.5.29).

word number	name	dimension	type	used in programs number	meaning
Name of the COMMON block: FLCT (continuation)					
5	RHON	scalar	R4	25, 27, 29.	used in the calculation of the Dresner factors in BINT (2.3.5.29)
6	RHOF	scalar	R4	25, 29.	used in the calculation of the Dresner factors in BINT (2.3.5.29).
7	NDIM6	scalar	I4	11, 25.	used for dimensioning. Is set to 6.
8	NPEMAX	scalar	I4	11, 25.	maximal number of energy values in the unresolved resonance part.
Name of the COMMON block: INOUT					
1	NO	scalar	I4	2, 4, 5, 6, 9, 10, 11, 12, 15, 18, 22, 24, 25, 30, 32, 35, 36, 37, 39, 41, 42, 44, 45, 46, 76, 78.	print output unit number
2	NI	scalar	I4	4, 5, 9.	control input unit number
3	NT	scalar	I4	8, 9, 10, 11, (12), 25, 36, 42, 44	tape unit containing ENDF/B-tape
4	MODE	scalar	I4	9, 10, 11, (12), (25), (36), (37), (42), (44),	arrangement of the ENDF/B-data (see /3/ chapter R-2.2).
5	LEDIT1	scalar	I4	9, 10	control of printing of - descriptive data and dictionary (MF = 1, MT = 451)

Name of the COMMON block: INOUT (continuation)					
word number	name	dimension	type	used in programs number	meaning
6	LEDIT2	scalar	I4	9, 12, 15, 25, 30	control of printing of -resolved resonance parameters (in ENDF representation) -resolved resonance data in KEDAK representation -unresolved resonance parameters (in ENDF representation) -KEDAK datatypes ST and STGF
7	LEDITS	scalar	I4	9, 24, 30, 32, 36.	Control of printing of -pointwise cross sections calculated from resolved resonance parameters -pointwise cross sections calculated from unresolved resonance parameters -pointwise total cross sections calculated from resolved resonance parameter -pointwise total cross sections calculated from statistical resonance parameter -ENDF/B (pointwise) reaction types of MF=3
8	KEDAVE	scalar	I4	9, 10, 36, 37	type of KEDAK-conventions used (see 2.2.6 namelist INPUT)
9	ABIS	(20)	R4	11, 34	abundance (weight fraction) of the isotopes
29	NOTTRL	(25)	R8	9, 36	names of datatypes not to be translated (see 2.2.6 namelist INPUT).
79	BIB12	scalar	I4	9, 78, 79	size of table of contents of dataset on FT12F001 (see 2.2.6 namelist INPUT).

Name of the COMMON block: INOUT (continuation)

word number	name	dimension	type	used in programs number	meaning
80	SALINE	scalar	L4	9, 36, 37	control of searching in MF=3 greater than MT=51 "starting" numbers (see 2.2.6 namelist INPUT).
81	NAMKDK	scalar	R8	9, 10, 11, 86	alphamerical KEDAK material name (see 2.2.6 namelist INPUT).
83	MATKDK	scalar	I4	9, 10, 11, 12, 30, (37), 42, (44), 86	numerical KEDAK material name (see 2.2.6 namelist INPUT).
84	NTYP	scalar	I4	77, 86	number of datatypes for a specified material
85	NAMISO	(10)	R8	9, 12, 15, 25, 30, 86	alphamerical KEDAK isotope names (see 2.2.6 namelist ISO).
105	MATISO	(10)	I4	9, 11, 15, 30, 86	numerical KEDAK isotope names (see 2.2.6 namelist ISO).
115	EBISO	(10)	R4	9, 11	binding energies of the last neutron of the isotopes (see 2.2.6 namelist ISO).
125	ICOMB	scalar	I4	9, 67	automatic choice of interpolation rule or linear-linear interpolation in the subroutine COMBP (see 2.2.6 namelist INPUT)
126	NBX	scalar	I4	4, 5, 49, 79, 81	maximal dimension of array B (see 2.3.3)
127	NIX	scalar	I4	4, 5, 12, 23, 24, (25), (36), (37), (44), 49, 67, 75, 79, 81	maximal dimension of the arrays NBT and JNT (see 2.3.3)

Name of the COMMON block: INOUT (continuation)

word number	name	dimension	type	used in programs number	meaning
128	N2X	scalar	I4	4, 5, (12), 22, 24, 25, (36), (37), 39, (44), 49, 67, 75, 79, 81	maximal dimension of the arrays X and Y (see 2.3.3)
129	N	scalar	I4	11, 15, 25, 30, 37, 42, 44, 76, 77, 86	number of data to be written in subroutine WDA (see 2.3.5.76) or read in subroutine RDAB (see 2.3.5.77)
130	NBKX	scalar	I4	4, 5, 11, 12, 15, 25, 30, 32, 37, 42, 44, 76	maximal dimension of the array BK (see 2.3.3)
131	JX	scalar	I4	4, 5, 11, 80, 82	maximal dimension of the array A (see 2.3.3) JX = IADIM
132	IADIM	scalar	I4	4, 5, 42	maximal dimension of the array A (see 2.3.3)
133	ITDIM	scalar	I4	4, 5, 86	maximal dimension of the arrays TAB resp. ITAB (see 2.3.3)
134	IVER	scalar	I4	1, 5	address of begin of the data region relative to F(1) (measured in 4-byte words) (see 2.3.3) (begin of array BK)
135	IVGENR	scalar	I4	1, 5	address of the begin of the arrays TAB resp. ITAB and NBT relative to BK(1) (measured in 4-byte words) (see 2.3.3)
136	IVJNT	scalar	I4	1, 5	address of the begin of the array JNT relative to BK(1) (measured in 4-byte words) (see 2.3.3)
137	IVX	scalar	I4	1, 5	address of the begin of the array X relative to BK(1) (measured in 4-byte words) (see 2.3.3)

Name of the COMMON block: INOUT (continuation)

word number	name	dimension	type	used in programs number	meaning
138	IVY	scalar	I4	1, 5	address of the begin of the array Y relative to BK(1) (measured in 4*byte words) (see 2.3.3)
139	IVB	scalar	I4	1, 5	address of the begin of the array B relative to BK(1) (measured in 4*byte words) (see 2.3.3)
140	IVDENS	scalar	I4	1, 5	address of the begin of the array A relative to BK(1) (measured in 4*byte words) (see 2.3.3)
141	EPSRES	scalar	R4	9, 22, (25), 32, 37, (44)	accuracy used in several cases (see 2.2.6 namelist INPUT)
142	SCREPS	scalar	R4	9, 37, (66)	accuracy used in deleting points, which can be predicted by linear interpolation (see 2.2.6 namelist INPUT)
143	LABEL	scalar	I4	(8), 9	control of reading the TPID-record and checking the tape identification number (see 2.2.6 namelist INPUT)
144	LNER	scalar	I4	(8), 9	number of points per resonance in EPT (see 2.2.6 namelist INPUT)
145	ITAPE	scalar	I4	(8), 9	control of writing the converted data of ENDF/B MF=3 on unit 15 (see 2.2.6 namelist INPUT)
146	MATP	scalar	I4	(8), 9	ENDF/B-material number (see 2.2.6 namelist INPUT)
147	EB	scalar	R4	(8), 9	binding energy of the last neutron (in eV) (see 2.2.6 namelist INPUT)
148	NISO	scalar	I4	(8), 9	number of isotopes for which data are given in the control input (see 2.2.6 namelists INPUT and ISO)

Name of the COMMON block: INOUT (continuation)					
word number	name	dimension	type	used in programs number	meaning
149	IF4	scalar	I4	(8), 9	control of calling the subroutine ISABEL (see 2.2.6 namelist INPUT)
150	IF5	scalar	I4	(8), 9	control of calling the subroutine FILE5 (see 2.2.6 namelist INPUT)
151	FAICOD	scalar	R4	9, 67	used in combining two pointwise cross sections in the subroutine COMBP (see 2.3.5.67 and 2.2.6 namelist INPUT)
152	MLSLSW	scalar	I4	9, 12	switch to enforce single level Breit Wigner instead of multi level Breit Wigner (see 2.2.6 namelist INPUT)
153	ANFZEI	scalar	R4	1, 45	value produced at the first ZEIT (2.3.5.97) call
154	ELINEG	scalar	I4	9, 10, 41	deletion of negative SGT and SGN cross section values (see 2.2.6 namelist INPUT)
Name of the COMMON block: RE					
1	XR	(300)	R4	73, 74	interchange of data between DANI (2.3.5.74) and CHGINT (2.3.5.73)
301	YR	(300)	R4	73, 74	see array XR

Name of the COMMON block: RECS

word number	name	dimension	type	used in programs number	meaning
1	MAT	scalar	I4	10, (37), 46, 48, 49, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 8o, 81	ENDF material number (see /3/ chapter R-2)
2	MF	scalar	I4	10, 36, 37, 46, 48, 49, 51, 52, 53, 54, 55, 56, 58, 59, 60, 61, 62, 80, 81	ENDF file number (see /3/ chapter R-2)
3	MT	scalar	I4	10, 36, 37, 46, 48, 49, 51, 52, 53, 54, 55, 56, 58, 59, 60, 61, 62, 80, 81	ENDF section (reaction type) number (see /3/ chapter R-2)
4	C1	scalar	R4	10, 11, 12, 24, 25, 37, 44, 49, 51, 52, 53, 54, 55, 56, 58, 59, 60, 61, 62, 80, 81	ENDF constant. (In most cases the temperature (see /3/ chapter R-2)
5	C2	scalar	R4	10, 11, 12, 25, 36, 37, 42, 44, 49, 51, 52, 53, 54, 55, 56, 58, 59, 60, 61, 62, 80, 81	ENDF constant (see /3/ chapter R-2)
6	L1	scalar	I4	10, 11, 12, 25, 37, 42, 49, 51, 52, 53, 54, 55, 56, 58, 59, 60, 61, 62, 80, 81	ENDF integer constant. (In most cases used to indicate temperature dependence) (see /3/ chapter R-2)
7	L2	scalar	I4	10, 11, 25, 37, 42, 44, 49, 51, 52, 53, 54, 55, 56, 58, 59, 60, 61, 62, 80, 81	ENDF integer constant (see /3/ chapter R-2)
8	N1	scalar	I4	10, 11, 12, 24, 25, 30, 36, (37), 39, 40, 42, (44), 49, 51, 52, 53, 54, 55, 56, 58, 59, 60, 61, 62, 66, 67, 71, 75, 78, 79, 80, 81	ENDF count of items in a list to follow (see /3/ chapter R-2)

Name of the COMMON block: RECS (continuation)					
word number	name	dimension	type	used in programs number	meaning
9	N2	scalar	I4	10, 12, 24, 25, 30, 34, 36, (37), 39, 40, 42, (44), 49, 51, 52, 53, 54, 55, 56, 58, 59, 60, 61, 62, 66, 67, 71, 75, 78, 79, 80, 81	ENDF count of items in a second list to follow (see /3/ chapter R-2)
10	NS	scalar	I4	10, 17, 37, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63	ENDF sequence number used in MODE=3 writing (see /3/ chapter R-2)
Name of the COMMON block: TAB					
1	ENDTAB	(25)	I4	4, 36, 37	ENDF MF=3 section numbers (MT) of data types to be converted
26	KEDTAB	(50)	I4	4, 36, 37	alphamerical KEDAK material names associated to the ENDF-section numbers stored in the array ENDTAB
76	KEDR	(240)	I4	4, 42, 44, 77, 86	list of alphamerical and associated numerical KEDAK data type names
316	NCNAME	(39)	I4	9, 42, 44, 86	number of incident energy points at which angular distributions and energy distributions of secondary neutrons are given
355	NRIF3	scalar	I4	9, 36, 37, 86	number of levels of KEDAK datatype SGIZ
356	NRIF4	scalar	I4	9, 42, 44, 86	number of datatypes from ENDF MF =4 and MF=5

	Name of the COMMON block: XBAR				
word number	name	dimension	type	used in programs number	meaning
1	XX	(4,10)	R4	4, 26	used in the calculation of the fluctuation integrals for capture, fission and elastic scattering

2.3.5 Description of the Main Program and the Subroutines

In the following the main program and all subroutines of BRIGITTE will be described. All the descriptions will have the following scheme:

- problem solved by the routine
- (- method of solution)
- definition with argument list (and entry names if existant)
- explanation of the quantities in the list(s)
- used common blocks and used quantities of these common blocks
- called subprograms (except those, which are FORTRAN-supplied procedures or subprograms like IBCOM)

In sections 2.3.5.92 until 2.3.5.99 a description of all auxiliary programs[+)], which are used by the program, will be given.

In the explanation lists of the arguments and in the used common blocks lists the following abbreviations will be used:

in the mode column:

- I — is used for variables which values must be defined before calling the subroutine or must have been set in the common by another program.
- O — is used for variables which are set in the subroutine (or in programs called by the subroutine) and are used outside (and within) the subroutine.
- L — is used for arrays (or variables) which are only used for placing some storage for the disposal of the subroutine and have only the meaning of local quantities.

[+)] all these programs are members of local libraries.

P means the same as L with the difference that the arrays or variables are not used in the subroutine itself but passed to other subroutines or functions. The values produced in this subroutines or functions are not used outside this subroutines or functions.

(L and P are only used in the list "Explanation of the argument", but not in the list "Used common blocks".)

IO is used for a combination of I and O.

PL is used for a combination of P and L. That means e.g. the array is used in some parts of the subroutine as local quantity and in others the address is passed to another subroutine.

\- is used for quantities, which are not referenced in the subprogram, or for quantities, for which none of the given classifications is true.

in the type column:

R4 for REAL*4 quantities

R8 for REAL*8 quantities

I4 for INTEGER*4 quantities

L4 for LOGICAL*4 quantities

B4 for 4 byte quantities of any type. The actual type will be determined by additional information.

F4 for functions supplying REAL*4 values

In the dimension column the maximal dimension of arrays will be given. For scalar quantities the word "scalar" is used.

In the "used common blocks" list a column headed with "word" is used. The number in this column means the starting address (measured in 4-byte words and beginning with word 1) of the associate variable or array.

In the "Explanation of the argument" lists often the word energy is used for the argument of a TAB1-record (see /3/ chapter R-2.3.3). This is done, because in nearly all cases this is an energy. But also other quantities are valid as argument, as for example cosine values.

2.3.5.1 Main_program

Problem solved:
Control of program flow. DEFINE FILE for direct access data sets.

Used common block:

name of the common block: INOUT									
name	dimension	type	mode	word	name	dimension	type	mode	word
NR1	(133)	I4	-	1	IVY	scalar	I4	I	138
IVER	scalar	I4	I	134	IVB	scalar	I4	I	139
IVGENR	scalar	I4	I	135	IVDENS	scalar	I4	I	140
IVJNT	scalar	I4	I	136	NR2	(12)	I4	-	141
IVX	scalar	I4	I	137	ANFZEI	scalar	I4	0	153

Called subprograms:

ZEIT
TITEL
PRIEIN
GSPACE
DADEFI
DEFI
PART1
SICKY

2.3.5.2 Subroutine TITEL

Problem solved:
Printing of title page.

Definition
Subroutine TITEL

Used common block:

name of the common block: INOUT				
name	dimension	type	mode	word
NO	scalar	I4	I	1

Called subprograms:

A8FORM
RAMANF } (Entries of A8FORM)
RAMEND }

2.3.5.3 Subroutine PRIEIN

Problem solved:
Printout of the input.

Definition:
SUBROUTINE PRIEIN(IEIN,IOUT,ZEILMA)

Explanation of the arguments:

name	mode	type	dimension	remarks
IEIN	I	I4	scalar	input unit number
IOUT	I	I4	scalar	print output unit number
ZEILMA	I	I4	scalar	maximal number of lines per printoutput page

Used common blocks:
None

Called subprograms:
None

2.3.5.4 BLOCK DATA

Problem solved:

Initialisation of some quantities in the common blocks.

Definition:

BLOCK DATA

Used common blocks:

name of the common block: TAB									
name	dimension	type	mode	word	name	dimension	type	mode	word
ENDTAB	(25)	I4	O	1	NCNAME	(39)	I4	-	316
KEDTAB	(50)	I4	O	26	NRIF3	scalar	I4	-	355
KEDR	(240)	I4	O	76	NRIF4	scalar	I4	-	356
name of the common block: XBAR									
XX	(4,10)	R4	O	1					
name of the common block: INOUT									
NO	scalar	I4	O	1	N	scalar	I4	-	129
NI	scalar	I4	O	2	NBKX	scalar	I4	O	130
NR1	(123)	I4	-	3	JX	scalar	I4	O	131
NBX	scalar	I4	O	126	IADIM	scalar	I4	O	132
N1X	scalar	I4	O	127	ITDIM	scalar	I4	O	133
N2X	scalar	I4	O	128	NR2	(21)	I4	-	134

Called subprograms:

None

2.3.5.5 Subroutine GSPACE

Problem solved:
Allocation of all free space in the main storage region as data working space.

Method of solution:
First the size of the region, which is not used by the program or for buffers, is determined. From this value all the region, which may be used for buffers later on, and the region for other system use is subtracted. The remaining value is used as data storage. The parcellation of this storage for the different arrays is also controlled by this routine.

Definition:
SUBROUTINE GSPACE(F)

Explanation of the arguments:

name	mode	type	dimension	remarks
F	L	R4	(1)	the subroutine uses this address as reference point

Used common block:

name of the common block: INOUT									
name	dimension	type	mode	word	name	dimension	type	mode	word
NO	scalar	I4	I	1	IADIM	scalar	I4	IO	132
NI	scalar	I4	I	2	ITDIM	scalar	I4	IO	133
NR1	(123)	I4	-	3	IVER	scalar	I4	O	134
NBX	scalar	I4	IO	126	IVGENR	scalar	I4	O	135
N1X	scalar	I4	IO	127	IVJNT	scalar	I4	O	136
N2X	scalar	I4	IO	128	IVX	scalar	I4	O	137
N	scalar	I4	-	129	IVY	scalar	I4	O	138
NBKX	scalar	I4	IO	130	IVB	scalar	I4	O	139
JX	scalar	I4	IO	131	IVDENS	scalar	I4	O	140

Called subprograms: BUFREG
FREESP
XTAREA

2.3.5.6 Subroutine BUFREG

Problem solved:
Determination of the region needed for buffers.

Method of solution:
All DD(data definition)-names in the TIOT (task input output table) of the form FTXXF001 with the attached BLKSIZE (block size) are fetched (for explanation of the terms see /10/ and other IBM system literature), the number of buffers is set to two, and with this values the place for buffers is calculated. If no values are found in the TIOT, they are fetched from the label.

Exceptions:
The file for print output (the file number is expected as first Integer*4 byte word in the named common block INOUT) is not considered.
If there are the DD-names SYSUDUMP or SYSABEND, two k-byte will be added to the buffer region for each one.

Definition:
BUFREG START

The program is written in IBM/370 Assembler. It is called from FORTRAN-programs by the statement CALL BUFREG (INTREG,&LABEL)

Explanation of the arguments:

name	mode	type	dimension	remarks
INTREG	O	I4	scalar	region-measured in k-byte-needed for buffers. If INTREG is less than zero, the program has found an error. In this case RETURN 1 is made
&LABEL	-	-	-	label for RETURN 1

Used common block:

name of the common block: INOUT				
name	dimension	type	mode	word
NO	scalar	I4	I	1

Called subprograms:

None

2.3.5.7 Subroutine DADEFI

Problem solved:

The subparameters dblkl (data block lenght) and nblk (number of blocks) out of SPACE = (dblkl,(nblk)) are fetched from the DD-cards FTOlF001, FT11F001, and FT12F001. They will be used as input in the dynamic DEFINE FILE statement program DEFI, which is called in the MAIN program.

Definition:

SUBROUTINE DADEFI(IF)

Explanation of the argument:

name	mode	type	dimension	remarks

IF	0	I4	(6)	dblkl and nblk (see problem solved) are stored in IF
				IF(1) and IF(4): dblkl and nblk of FTOlF001.
				IF(2) and IF(5): dblkl and nblk of FT11F001.
				IF(3) and IF(6): dblkl and nblk of FT12F001

Used common blocks:

None.

Called subprograms:

DINF

2.3.5.8 Subroutine PART1

Problem solved:

Control of program flow.

Definition:

SUBROUTINE PART1(NIS,MISO,KIF,KIF5,BK,NBT,JNT,X,Y,B,A)

Explanation of the arguments:

name	mode	type	dimension	remarks
NIS	O	I4	scalar	number of isotopes for the specified ENDF/B-material
MISO	O	I4	scalar	number of isotopes for which data are specified in the control input
KIF	O	I4	scalar	value of IF4 (default value or given in the control input) (see input description)
KIF5	O	I4	scalar	value of IF5 (default value or given in the control input) see input description 2.2.6)
BK	P	R4	(NBKX)	array and dimension see 2.3.3
NBT	P	I4	(N1X)	" " " "
JNT	P	I4	(N1X)	" " " "
X	P	R4	(N2X)	" " " "
Y	P	R4	(N2X)	" " " "
B	P	R4	(NBX)	" " " "
A	P	R4	(IADIM)	" " " "

Used common block:

name of the common block: INOUT									
name	dimension	type	mode	word	name	dimension	type	mode	word
NR1	(2)	I4	-	1	MATP	scalar	I4	O	146
NT	scalar	I4	I	3	EB	scalar	R4	O	147
NR2	(139)	I4	-	4	NISO	scalar	I4	O	148
LABEL	scalar	I4	O	143	IF4	scalar	I4	O	149
LNER	scalar	I4	O	144	IF5	scalar	I4	O	150
ITAPE	scalar	I4	O	145					

Called subprograms:

DENISE	
EITAB1	(entry of ITAB1)
FILE5	
INGRID	
INITNL	
IRREC	(entry of RREC)
ISABEL	
IWREC	(entry of WREC)
REGINE	
SUZY	
WDA1, WDA3	(entries of WDA)
WRECS1	(entry of WRECS)

2.3.5.9 Subroutine INITNL

Problem solved:
Default values are set for most of the variables in the namelist input. The control input (namelist(s) INPUT and eventually ISO) is read.

Definition:
SUBROUTINE INITNL(MISO,*)

Explanation of the arguments:

name	mode	type	dimension	remarks
MISO	O	I4	scalar	number of isotopes for which data are given in the control input (MISO<1 has the same meaning as MISO=1)
*	-	-	-	RETURN1 is made, if no (or nomore) namelist INPUT can be found in the control input

Used common blocks:

name of the common block: INOUT									
name	dimension	type	mode	word	name	dimension	type	mode	word
NO	scalar	I4	0	1	EBISO	(10)	R4	0	115
NI	scalar	I4	0	2	ICOMB	scalar	I4	0	125
NT	scalar	I4	0	3	NR1	(15)	I4	-	126
MODE	scalar	I4	0	4	EPSRES	scalar	R4	0	141
LEDIT1	scalar	I4	0	5	SCREPS	scalar	R4	0	142
LEDIT2	scalar	I4	0	6	LABEL	scalar	I4	0	143
LEDITS	scalar	I4	0	7	LNER	scalar	I4	0	144
KEDAVE	scalar	I4	0	8	ITAPE	scalar	I4	0	145
ABIS	(20)	R4	-	9	MATP	scalar	I4	0	146
NOTTRL	(25)	R8	0	29	EB	scalar	R4	0	147
BIB12	scalar	I4	0	79	NISO	scalar	I4	0	148
SALINE	scalar	L4	0	80	IF4	scalar	I4	0	149
NAMKDK	scalar	R8	0	81	IF5	scalar	I4	0	150
MATKDK	scalar	I4	0	83	FAICOD	scalar	R4	0	151
NTYP	scalar	I4	-	84	MLSLSW	scalar	I4	0	152
NAMISO	(10)	R8	0	85	NR3	(1)	I4	-	153
MATISO	(10)	I4	0	105	ELINEG	scalar	I4	0	154

name of the common block: TAB									
NR2	(315)	I4	-	1	NRIF3	scalar	I4	0	355
NCNAME	(39)	I4	0	316	NRIF4	scalar	I4	0	356

Called subprograms:

None.

2.3.5.10 Subroutine INGRID

Problem solved:

Reads and checks the ENDF-label. Prints some text. Reads MF=1 MT=451 (Descriptive Data and Dictionary) and prints this section if desired. If existant MF=1 MT=452 (number of neutrons per fission, $\bar{\nu}$) is read, printed out and stored for later processing.

Definition:

SUBROUTINE INGRID(MATP,LABEL,LRP,LFI,LNU,IF4,IF5)

Explanation of the arguments:

name	mode	type	dimension	remarks
MATP	I	I4	scalar	number of the ENDF/B-material to be converted
LABEL	I	I4	scalar	see 2.2.6 (input description)
LRP	O	I4	scalar	flag to indicate the existance of resolved and/or unresolved resonance data (see /2/)
LFI	O	I4	scalar	LFI = 0 material is not fissionable LFI = 1 material is fissionable
LNU	O	I4	scalar	LNU = 1 polynomial representation of $\bar{\nu}$. LNU = 2 tabulated representation of of $\bar{\nu}$.
IF4	I	I4	scalar	see 2.2.6 (input description)
IF5	I	I4	scalar	" " "

Used common blocks:

name of the common block: INOUT									
name	dimension	type	mode	word	name	dimension	type	mode	word
NO	scalar	I4	I	1	KEDAVE	scalar	I4	I	8
NI	scalar	I4	-	2	NR2	(72)	I4	-	9
NT	scalar	I4	I	3	NAMKDK	scalar	R8	I	81
MODE	scalar	I4	I	4	MATKDK	scalar	I4	I	83
LEDIT1	scalar	I4	I	5	NR3	(70)	I4	-	84
NR1	(2)	I4	-	6	ELINEG	scalar	I4	I	154

name of the common block: RECS									
MAT	scalar	I4	I	1	L1	scalar	I4	I	6
MF	scalar	I4	I	2	L2	scalar	I4	I	7
MT	scalar	I4	-	3	N1	scalar	I4	I	8
C1	scalar	R4	-	4	N2	scalar	I4	I	9
C2	scalar	R4	-	5	NS	scalar	I4	0	10

(The first nine words in the Common RECS are set to zero at the beginning of the subroutine.)

Called subprograms:

A8FORM

ERROR

PRIZEI

RAMANF } (entries of A8FORM)

RAMEND }

RREC

SEARCH

WREC

WRECS2 } (entries of WRECS)

WRECS4 }

2.3.5.11 Subroutine REGINE

Problem solved:

Processing of ENDF MF=2 (resonance data). Generation of ISOT1 and ISOT2 (and eventually ISOT3) for KEDAK.

Definition:

SUBROUTINE REGINE(MATP,LNER,LISRES,LISUNR,NIS,LFWRES,LFW,EB,LRP,NISO,BK, NBT,JNT,X,Y,B,AH)

Explanation of the arguments:

name	mode	type	dimension	remarks
MATP	I	I4	scalar	number of the ENDF/B-material, which is to be converted
LNER	I	I4	scalar	see 2.2.6 (input description)
LISRES	O	I4	scalar	= 0 cross sections in the resolved resonance region are to be calculated from resonance parameters plus smooth cross sections from file 3. = 1 only smooth cross sections in file 3 are used.
LISUNR	O	I4	scalar	= 0 cross sections in the unresolved resonance region are to be calculated from resonance parameters plus smooth cross sections from file 3. = 1 only smooth cross sections in file 3 are used.
NIS	O	I4	scalar	number of isotopes in ENDF MF=2 (resonance data)
LFWRES	O	I4	scalar	= 0 zero fission widths = 1 non zero fission widths in the resolved resonance region
LFW	O	I4	scalar	= 1 fission widths are given = 0 fission widths are not given } in the unresolved resonance region
EB	I	R4	scalar	see input description 2.2.6. In the case of more than one isotope EB is set to EBISO(MIS) (see input description 2.2.6) after storing the original value of EB in the ISOT2 data-block of the entire material

name	mode	type	dimension	remarks
LRP	I	I4	scalar	see 2.3.5.10 (explanation of the arguments)
NISO	I	I4	scalar	number of isotopes for which data are given in the input (see 2.2.6)
BK	PL	R4	(NBKX)	see 2.3.3
NBT	P	I4	(N1X)	"
JNT	P	I4	(N1X)	"
X	P	R4	(N2X)	"
Y	P	R4	(N2X)	"
B	P	R4	(NBX)	"
AH	P	R4	(JX)	"

Used common blocks:

name of the common block: INOUT									
name	dimension	type	mode	word	name	dimension	type	mode	word
NO	scalar	I4	I	1	MATKDK	scalar	I4	I	83
NI	scalar	I4	-	2	NR3	(21)	I4	-	84
NT	scalar	I4	I	3	MATISO	(10)	I4	I	105
MODE	scalar	I4	I	4	EBISO	(10)	R4	I	115
NR1	(4)	I4	-	5	NR4	(4)	I4	-	125
ABIS	(20)	R4	O	9	N	scalar	I4	O	129
NR2	(52)	I4	-	29	NBKX	scalar	I4	I	130
NAMKDK	scalar	R8	I	81	JX	scalar	I4	I	131

name of the common block: RECS									
name	dimension	type	mode	word	name	dimension	type	mode	word
MAT	scalar	I4	-	1	C2	scalar	R4	I	5
MF	scalar	I4	-	2	L1	scalar	I4	I	6
MT	scalar	I4	-	3	L2	scalar	I4	I	7
C1	scalar	R4	I	4	N1	scalar	I4	I	8

name of the common block: FLCT									
NR5	(6)	I4	-	1	NPEMAX	scalar	I4	0	8
NDIM6	scalar	I4	0	7					

Called subprograms:

ANNICK
A8FORM
ERROR
MARY
PRIZEI
RREC
SEARCH
WDA2 (entry of WDA)
XTEXT

2.3.5.12 Subroutine MARY

Problem solved:
Control of calculation of pointwise cross sections out of resolved resonance parameters.

Definition:

```
SUBROUTINE MARY(EL,EH,LRF,ZAI,ABN,LFWRES,NIS,LNER,MIS,LIS,SPI,AWRI,AP,BK,
                NBT,JNT,ENS,SIGS,B,AH,JXRED,ER,AJ,GT,GN,GG,GF,AL,G,SER,
                PER,NDIM)
```

Explanation of the arguments:

name	mode	type	dimension	remarks
EL	I	R4	scalar	lower energy (eV) limit for the resolved resonance region
EH	I	R4	scalar	upper energy limit (eV) for the resolved resonance region
LRF	I	I4	scalar	= 1 single level Breit-Wigner = 2 multi level Breit-Wigner
ZAI	I	R4	scalar	(Z,A) designation (see /2/)
ABN	I	R4	scalar	fractional abundance (see /2/)
LFWRES	O	I4	scalar	= 1 non zero fission widhts given (as resolved resonance parameters)
NIS	I	I4	scalar	number of isotopes in ENDF MF=2 (resonance data)
LNER	I	I4	scalar	see input description (2.2.6)
MIS	I	I4	scalar	numeration of the isotope worked with
LIS	O	I4	scalar	see LISRES in 2.3.5.11 (explanation of the arguments)
SPI	O	R4	scalar	nuclear spin of the target nucleus (see /2/)
AWRI	O	R4	scalar	ratio of the mass of a particular isotope to that of a neutron (see /2/)
AP	O	R4	scalar	effective scattering radius (see /2/)
BK	P	R4	(NBKX)	see 2.3.3
NBT	P	I4	(N1X)	"
JNT	P	I4	(N1X)	"
ENS	L	R4	(N2X)	energy grid (see 2.3.3 array X)
SIGS	P	R4	(N2X)	see 2.3.3 array Y

name	mode	type	dimension	remarks
B	L	R4	(NBX)	resonance parameters (see 2.3.3)
AH	L	R4	(JXRED)	used for storing SGG,SGN and eventually SGF pointwise cross sections
JXRED	I	I4	scalar	JXRED = JX - JX/20
ER	L	R4	(NDIM)	resonance energy (see /2/)
AJ	L	R4	(NDIM)	spin of the resonance (see /2/)
GT	L	R4	(NDIM)	resonance total width Γ evaluated at the resonance energy ER (see /2/)
GN	L	R4	(NDIM)	neutron width Γ_n (see /2/)
GG	L	R4	(NDIM)	radiation width Γ_γ (see /2/)
GF	L	R4	(NDIM)	fission width Γ_f (see /2/)
AL	L	R4	(NDIM)	neutron angular momentum quantum number l (see /2/)
G	L	R4	(NDIM)	(2*AJ(k)+1)/(2*(2*SPI+1)) (see /2/)
SER	L	R4	(NDIM)	shift factors (see /2/)
PER	L	R4	(NDIM)	penetration factors (see /2/)
NDIM	I	I4	scalar	NDIM = JX/200 (JX see 2.3.3)

Used common blocks:

name of the common block: INOUT									
name	dimension	type	mode	word	name	dimension	type	mode	word
NO	scalar	I4	I	1	NREF2	(22)	I4	-	105
NI	scalar	I4	-	2	N1X	scalar	I4	I	127
NT	scalar	I4	I	3	N2X	scalar	I4	I	128
MODE	scalar	I4	I	4	N	scalar	I4	-	129
LEDIT1	scalar	I4	-	5	NBKX	scalar	I4	I	130
LEDIT2	scalar	I4	I	6	NREF4	(21)	I4	-	131
NREF1	(78)	I4	-	7	MLSLSW	scalar	I4	I	152
NAMISO	(10)	R8	I	85					

name of the common block: RECS									
name	dimension	type	mode	word	name	dimension	type	mode	word
NREF3	(3)	I4	-	1	L2	scalar	I4	-	7
C1	scalar	R4	I	4	N1	scalar	I4	I	8
C2	scalar	R4	I	5	N2	scalar	I4	I	9
L1	scalar	I4	I	6					

Called subprograms:

EPT
ERROR
FACTS
MARCEL
MARGOT (entry of MARCEL)
MARYO1
MARYO2
MELA
MILLI
PRIZEI
RREC
SORTRE

2.3.5.13 Subroutine FACTS

Problem solved:
Calculates penetration and shift factors liberated from Gregson et al.

Definition:
SUBROUTINE FACTS(L,RHO,SE,PE)

Explanation of the arguments:

name	mode	type	dimension	remarks
L	I	I4	scalar	l-value (see AL in 2.3.5.12)
RHO	I	R4	scalar	see ρ in /2/ page D-3
SE	O	R4	scalar	shift-factor (see /2/ page D-3)
PE	O	R4	scalar	penetration-factor (see /2/ page D-4)

Used common blocks:
None

Called subprograms:
None

2.3.5.14 Subroutine SORTRE

Problem solved:
Sorts (resolved) resonance data on increasing energy.

Definition:
SUBROUTINE SORTRE(NRES,ER,AJ,GT,GN,GG,GF,AL,G,SER,PER)

Explanation of the arguments:

name	mode	type	dimension	remarks
NRES	I	I4	scalar	number of (resolved) resonances
ER	IO	R4	(NDIM)	see 2.3.5.12 (argument list)
AJ	IO	R4	(NDIM)	" "
GT	IO	R4	(NDIM)	" "
GN	IO	R4	(NDIM)	" "
GG	IO	R4	(NDIM)	" "
GF	IO	R4	(NDIM)	" "

name	mode	type	dimension	remarks
AL	IO	R4	(NDIM)	see 2.3.5.12 (argument list)
G	IO	R4	(NDIM)	" "
SER	IO	R4	(NDIM)	" "
PER	IO	R4	(NDIM)	" "

Used common blocks:
None

Called subprograms:
None

2.3.5.15 Subroutine MARYO1

Problem solved:
Stores (resolved) resonance data in the array BK. Changes Γ_n for negative resonance energies. Optional printout of resolved resonance data in KEDAK representation. Printout of KEDAK type RANGRES. Stores resolved resonance data on file 11 with the name RES. Determines if fission widths are zero.

Definition:
SUBROUTINE MARYO1(EL,EH,NRES,BK,ER,AL,AJ,G,GT,GN,GG,GF,NIS,MIS,LIS,LFWRES,*)

Explanation of the arguments:

name	mode	type	dimension	remarks
EL	I	R4	scalar	see 2.3.5.12 argument list
EH	I	R4	scalar	" "
NRES	I	I4	scalar	number of (resolved) resonances
BK	L	R4	(NBKX)	local storage for (resolved) resonance data and other quantities
ER	I	R4	(NDIM)	see 2.3.5.12 (argument list)

name	mode	type	dimension	remarks
AL	I	R4	(NDIM)	see 2.3.5.12 (argument list)
AJ	I	R4	(NDIM)	" "
G	I	R4	(NDIM)	" "
GT	I	R4	(NDIM)	" "
GN	I	R4	(NDIM)	" "
GG	I	R4	(NDIM)	" "
GF	I	R4	(NDIM)	" "
NIS	I	I4	scalar	number of isotopes in ENDF MF=2 (resonance data)
MIS	I	I4	scalar	numeration of the isotope actually worked with
LIS	I	I4	scalar	see LISRES in 2.3.5.11 (argument list)
LFWRES	O	I4	scalar	= 1 nonzero fission widths given (as resolved resonance parameters) = 0 all fission widths are zero
*	-	-	-	RETURN 1 is made for LIS not equal zero. But LIS is always zero after the program MARYO1 has been called. Therefore RETURN 1 is never made.

Used common block:

name of the common block: INOUT									
name	dimension	type	mode	word	name	dimension	type	mode	word
NO	scalar	I4	I	1	NAMISO	(10)	R8	I	85
NR1	(4)	I4	-	2	MATISO	(10)	I4	I	105
LEDIT2	scalar	I4	I	6	NR4	(14)	I4	-	115
NR2	(76)	I4	-	7	N	scalar	I4	O	129
MATKDK	scalar	I4	I	83	NBKX	scalar	I4	I	130
NR3	(1)	I4	-	84					

Called subprograms:

A8FORM

ERROR

LUCY

WDA2 (entry of WDA)

XTEXT

2.3.5.16 Subroutine LUCY

Problem solved:

Prints resolved or statistical resonance data, which are stored in the array BK, with a defined format.

Definition:

SUBROUTINE LUCY(I,BK)

Explanation of the arguments:

name	mode	type	dimension	remarks
I	I	I4	scalar	= 2 resolved resonance data format = 3 statistical data format
BK	I	R4	(NBKX)	data to be printed

Used common block:

name of the common block: INOUT									
name	dimension	type	mode	word	name	dimension	type	mode	word
NO	scalar	I4	I	1	N	scalar	I4	I	129
NR1	(127)	I4	-	2					

Called subprograms:

ERROR

2.3.5.17 Subroutine EPT

Problem solved:

Calculates energy mesh based on NPR points per resonance.

Method of solution:

See comment statements in the source program listing within the subroutine EPT.

Definition:

SUBROUTINE EPT(EMIN,EMAX,NPR,NTP,NR,ER,GTR,ENS,N2X)

Explanation of the arguments:

name	mode	type	dimension	remarks
EMIN	I	R4	scalar	lower limit for the energy mesh
EMAX	I	R4	scalar	upper limit for the energy mesh
NPR	I	I4	scalar	(maximum) number of points used per resonance
NTP	O	I4	scalar	number of points to describe NR resonances from EMIN to EMAX
NR	I	I4	scalar	number of resonances
ER	I	R4	(NR)	resonance energies
GTR	I	R4	(NR)	total widths of resonances
ENS	O	R4	(N2X)	calculated energy mesh
N2X	I	I4	scalar	maximum dimension of array ENS

Used common block:

name of the common block: RECS									
name	dimension	type	mode	word	name	dimension	type	mode	word
NR1	(9)	I4	-	1	NS	scalar	I4	L	10

Called subprograms:

RESERR

2.3.5.18 Subroutine RESERR

Problem solved:

Prints error messages for the resolved resonance calculating program section.

Definition:

SUBROUTINE RESERR(N,K,NSE,IERR)

Explanation of the arguments:

name	mode	type	dimension	remarks
N	I	I4	scalar	selection of different error messages
K	I	I4	scalar	number to be printed in the subroutine
NSE	I	I4	scalar	number to be printed in the subroutine
IERR	I	I4	scalar	selection of different error messages

Used common block:

name of the common block: INOUT				
name	dimension	type	mode	word
IO	scalar	I4	I	1

Called subprograms:

None

2.3.5.19 Subroutine MARY02

Problem solved:

Deletes all but one energy point with the same value.

Definition:

SUBROUTINE MARY02(ENS,NSE)

Explanation of the arguments:

name	mode	type	dimension	remarks
ENS	IO	R4	(N2X)	energy points
NSE	IO	I4	scalar	number of energy points

Used common blocks:
None

Called subprograms:
None

2.3.5.20 Subroutine MARCEL

Problem solved:
Calculates for one energy point E with a single level (or multi level) Breit-Wigner formulae elastic, radiative capture and depending on LFWRES fission cross sections. For LRF = 1 the single level and for LRF = 2 the multi level formalism is used.

Definition:
For initialisation:

```
SUBROUTINE MARCEL(LFWRES,LRF,C,CON3,CON4,PIM4,AP,NRES,NLS,LLSTR,ER,GN,GG,
                  GF,G,AL,SER,PER)
```

For calculation:

```
ENTRY MARGOT(E,SGN,SGG,SGF)
```

Explanation of the arguments:

name	mode	type	dimension	remarks
LFWRES	I	I4	scalar	only for LFWRES=1 fission cross section are calculated
LRF	I	I4	scalar	for LRF=2 a multi level Breit Wigner formulae is used. Else a single level Breit Wigner formulae is used
C	I	R4	scalar	= $2.196771E-3 \cdot AP \cdot \frac{AWRI}{AWRI+1.0}$ (AP and AWRI see 2.3.5.12 argument list)
CON3	I	R4	scalar	= $2.196771E-3 \cdot \frac{AWRI}{AWRI+1.0} \cdot 0.1 \cdot (0.8+1.23 \cdot AWRI^{1/3})$ (AWRI see 2.3.5.12 argument list)
CON4	I	R4	scalar	= $4\pi/(2.196771E-3)^2 \cdot (\frac{AWRI+1.0}{AWRI})^2$ (AWRI see 2.3.5.12 argument list)
PIM4	I	R4	scalar	= 4π
AP	I	R4	scalar	see 2.3.5.12 argument list
NRES	I	I4	scalar	number of (resolved) resonances
NLS	I	I4	scalar	number of l-states (see /2/ page 7.5)
LLSTR	I	I4	(5)	l-values for the different l-states (see /2/ page 7.5)
ER	I	R4	(NDIM)	resonance energies (see /2/ page 7.5) (NDIM see 2.3.5.12 argument list)
GN	I	R4	(NDIM)	neutron width Γ_n
GG	I	R4	(NDIM)	radiation width Γ_γ
GF	I	R4	(NDIM)	fission width Γ_f
G	I	R4	(NDIM)	see 2.3.5.12 argument list
AL	I	R4	(NDIM)	" "
SER	I	R4	(NDIM)	shift factors (see /2/ page D-3)
PER	I	R4	(NDIM)	penetration factors (see /2/ page D-3)
E	I	R4	scalar	energy for which the pointwise cross sections are to be calculated
SGN	O	R4	scalar	$\sigma_e(E)$
SGG	O	R4	scalar	$\sigma_\gamma(E)$
SGF	O	R4	scalar	$\sigma_f(E)$ (only for LFWRES=1)

Used common blocks:

None

Called subprograms:

ERROR

FACPHI

FACTS

2.3.5.21 Subroutine FACPHI

Problem solved:

Calculates phase shift for the Breit Wigner formulae (see /2/ page D-4).

Definition:

SUBROUTINE FACPHI(L,RHO,PHI)

Explanation of the arguments:

name	mode	type	dimension	remarks
L	I	I4	scalar	l-value (see /2/ page 7.5)
RHO	I	R4	scalar	= k•AP (see /2/ page D-3 and D-4)
PHI	O	R4	scalar	calculated phase shift

Used common blocks:

None

Called subprograms:

None

2.3.5.22 Subroutine MILLI

Problem solved:

Checks if there are enough points describing the point resonance cross sections. If not the number of points will be enlarged. Also the best interpolation scheme is determined.

Definition:

SUBROUTINE MILLI(ENS,NSE,AH,NBTN,JNTN,NBTG,JNTG,NBTF,JNTF,LFWRES,NISGN,
NISGG,NISGF,JXRED)

Explanation of the arguments:

name	mode	type	dimension	remarks
ENS	IO	R4	(N2X)	energy grid
NSE	IO	I4	scalar	number of energy points in ENS
AH	IO	R4	(JXRED)	1 ... NSE : σ_e NSE+1 ... 2*NSE: σ_γ For LFWRES = 1 2*NSE+1 ... 3*NSE: σ_f
NBTN	O	I4	(NIX)	interpolation ranges for σ_e
JNTN	O	I4	(NIX)	interpolation scheme identification number for σ_e (in the resolved resonance region)
NBTG	O	I4	(NIX)	interpolation ranges for σ_γ
JNTG	O	I4	(NIX)	interpolation scheme identification number for σ_γ
NBTF	O	I4	(NIX)	interpolation ranges for σ_f
JNTF	O	I4	(NIX)	interpolation scheme identification number for σ_f
LFWRES	I	I4	scalar	Only for LFWRES=1 fission cross section are processed
NISGN	O	I4	scalar	number of interpolation ranges for σ_e
NISGG	O	I4	scalar	number of interpolation ranges for σ_γ
NISGF	O	I4	scalar	number of interpolation ranges for σ_f
JXRED	I	I4	scalar	maximal usable extent of array AH (measured in 4 byte words)

Used common block:

name of the common block: INOUT									
name	dimension	type	mode	word	name	dimension	type	mode	word
IOUT	scalar	I4	I	1	NR2	(12)	I4	-	129
NR1	(126)	I4	-	2	EPS	scalar	R4	I	141
N2X	scalar	I4	I	128					

Called subprograms:

ERROR

MARGOT

MILLI1

RESERR

TERP1

2.3.5.23 Subroutine MILLI1

Problem solved:

Inserts interpolation code IC and interpolation range K in JNT resp. NBT.

Definition:

SUBROUTINE MILLI1(NBT,JNT,N1,IC,K,NSE)

Explanation of the arguments:

name	mode	type	dimension	remarks
NBT	O	I4	(N1X)	interpolation ranges
JNT	IO	I4	(N1X)	interpolation codes
N1	IO	I4	scalar	number of interpolation ranges resp. interpolation codes
IC	I	I4	scalar	interpolation code to be inserted in JNT
K	I	I4	scalar	interpolation range to be inserted in NBT
NSE	I	I4	scalar	number of energy points processed until now

Used common block:

name of the common block: INOUT									
name	dimension	type	mode	word	name	dimension	type	mode	word
NR1	(126)	I4	-	1	N1X	scalar	I4	I	127

Called subprograms:

RESERR

2.3.5.24 Subroutine MELA

Problem solved:

Prints and stores (on unit 12) the calculated pointwise cross sections σ_e, σ_γ and σ_f (for LFWRES=1) in the resolved resonance region.

Definition:

SUBROUTINE MELA(LFWRES,IBK,AH,NBT,JNT,SIGS,N1SGN,N1SGG,N1SGF,NSE,MIS,EL,EH)

Explanation of the arguments:

name	mode	type	dimension	remarks
LFWRES	I	I4	scalar	fission cross sections are only processed for LFWRES=1
IBK	I	I4	(NBKX)	1 ... N1X: NBT for σ_e N1X+1 ... 2*N1X: JNT for σ_e 2*N1X+1 ... 3*N1X: NBT for σ_γ 3*N1X+1 ... 4*N1X: JNT for σ_γ only for LFWRES=1: 4*N1X+1 ... 5*N1X: NBT for σ_f 5*N1X+1 ... 6*N1X: JNT for σ_f (6*N1X≤NBKX)
AH	I	R4	(JXRED)	σ_e,σ_γ and for LFWRES=1 σ_f (see 2.3.5.22 argument list)
NBT	O	I4	(N1X)	interpolation ranges
JNT	O	I4	(N1X)	interpolation codes

name	mode	type	dimension	remarks
SIGS	O	R4	(N2X)	pointwise cross sections
N1SGN	I	I4	scalar	see 2.3.5.22 argument list
N1SGG	I	I4	scalar	" " " "
N1SGF	I	I4	scalar	" " " "
NSE	I	I4	scalar	number of energy values in the array X
MIS	I	I4	scalar	actual isotope numeration
EL	I	R4	scalar	lower limit for the resolved resonance region
EH	I	R4	scalar	upper limit for the resolved resonance region

Used common blocks:

name of the common block: INOUT									
name	dimension	type	mode	word	name	dimension	type	mode	word
NO	scalar	I4	I	1	NR2	(119)	I4	-	8
NR1	(5)	I4	-	2	N1X	scalar	I4	I	127
LEDITS	scalar	I4	I	7	N2X	scalar	I4	I	128

name of the common block: RECS									
NR3	(3)	I4	-	1	N1	scalar	I4	O	8
C1	scalar	R4	O	4	N2	scalar	I4	O	9
NR4	(3)	I4	-	5					

Called subprograms:

ERROR

WREC

WRECS2 (entry of WRECS)

2.3.5.25 Subroutine ANNICK

Problem solved:

Calculation of pointwise cross sections in the unresolved resonance region out of average resonance parameters (see /2/ pages D-9 to D-17).

Definition:

```
SUBROUTINE ANNICK(EL,EH,LRF,ZAI,ABN,LFW,NIS,MIS,LIS,SPI,AWRI,AP,BK,NBT,JNT,
                  BK2,Y,B,SIGS,E,SIGG,SIGF,SIGT,ESC,XT,YT,TEMP,D,GX,GNO,GG,
                  GF,AJ)
```

Explanation of the arguments:

name	mode	type	dimension	remarks
EL	I	R4	scalar	lower limit for the unresolved resonance range
EH	I	R4	scalar	upper limit for the unresolved resonance range
LRF	I	I4	scalar	= 1 only average fission widths are energy-dependant = 2 average level spacing, competitive reaction widths, reduced neutron widths, radiation widths, and fission widths are energy dependant
ZAI	-	R4	scalar	not used in ANNICK
ABN	-	R4	scalar	not used in ANNICK
LFW	I	I4	scalar	= 0 average fission widths are not given = 1 average fission widths are given in the unresolved resonance region
NIS	I	I4	scalar	number of isotopes
MIS	I	I4	scalar	pointer to the actual isotope, if NIS is greater than 1
LIS	O	I4	scalar	is set to zero
SPI	O	R4	scalar	nuclear spin of target nucleus
AWRI	O	R4	scalar	ratio of the mass of the particular isotope to that of the neutron
AP	-	R4	scalar	not used in ANNICK
BK	L	R4	(NBKX)	unresolved resonance parameters

name	mode	type	dimension	remarks
NBT	L	I4	(N1X)	interpolation ranges
JNT	L	I4	(N1X)	interpolation codes
BK2	L	R4	(N2X)	(see array X in 2.3.3) data for KEDAK-type STGF (see /4/ p. 44)
Y	P	R4	(N2X)	-
B	L	R4	(NBX)	is filled in RREC with the unresolved resonance data
SIGS	L	R4	(NPEMAX)	(part of the array AH from word 1 until word NPEMAX) elastic (pointwise) scattering cross sections. (NPEMAX see common FLCT)
E	L	R4	(NPEMAX)	(AH(NPEMAX+1)...AH(2*NPEMAX)) energy grid in the unresolved resonance region with any interpolation code
SIGG	L	R4	(NPEMAX)	(AH(2*NPEMAX+1)...AH(3*NPEMAX)) radiative capture cross sections
SIGF	L	R4	(NPEMAX)	(AH(3*NPEMAX+1)...AH(4*NPEMAX)) fission cross sections
SIGT	L	R4	(NPEMAX)	(AH(4*NPEMAX+1)...AH(5*NPEMAX)) total cross sections
ESC	L	R4	(NPEMAX)	(AH(5*NPEMAX+1)...AH(6*NPEMAX)) energy grid in the unresolved resonance region with linear interpolation
XT	L	R4	(NPEMAX)	(AH(6*NPEMAX+1)...AH(7*NPEMAX)) local storage of energy grids
YT	L	R4	(NPEMAX)	(AH(7*NPEMAX+1)...AH(8*NPEMAX)) local storage of $\overline{\Gamma_n^o}$ or $\overline{\Gamma_f}$
TEMP	P	R4	(NPEMAX)	(AH(8*NPEMAX+1)...AH(9*NPEMAX)) auxiliary array
D	L	R4	(NDIM6,NPEMAX)	(AH(9*NPEMAX+1)...AH(15*NPEMAX)) (NDIM6 has the value 6) mean level spacing
GX	L	R4	(NDIM6,NPEMAX)	(AH(15*NPEMAX+1)...AH(21*NPEMAX)) average competitive reaction width
GNO	L	R4	(NDIM6,NPEMAX)	(AH(21*NPEMAX+1)...AH(27*NPEMAX)) average reduced neutron width

name	mode	type	dimension	remarks
GG	L	R4	(NDIM6,NPEMAX)	(AH(27*NPEMAX+1)...AH(33*NPEMAX)) average radiation width
GF	L	R4	(NDIM6,NPEMAX)	(AH(33*NPEMAX+1)...AH(39*NPEMAX)) average fission width
AJ	L	R4	(NDIM6,NPEMAX)	(AH(39*NPEMAX+1)...AH(45*NPEMAX)) floating point value of the j-state

Used common blocks:

name of the common block: INOUT									
name	dimension	type	mode	word	name	dimension	type	mode	word
NO	scalar	I4	I	1	MATISO	(10)	I4	-	105
NI	scalar	I4	-	2	NR2	(11)	I4	-	115
NT	scalar	I4	I	3	NBX	scalar	I4	-	126
MODE	scalar	I4	I	4	N1X	scalar	I4	I	127
LEDIT1	scalar	I4	-	5	N2X	scalar	I4	I	128
LEDIT2	scalar	I4	I	6	N	scalar	I4	O	129
NR1	(76)	I4	-	7	NBKX	scalar	I4	I	130
MATKDK	scalar	I4	-	83	NR3	(10)	I4	-	131
NTYP	scalar	I4	-	84	EPS	scalar	R4	I	141
NAMISO	(10)	R8	I	85					

name of the common block: FLCT									
ZYN	scalar	R4	O	1	RHON	scalar	R4	O	5
ZYF	scalar	R4	O	2	RHOV	scalar	R4	O	6
U	scalar	R4	O	3	NDIM6	scalar	I4	I	7
V	scalar	R4	O	4	NPEMAX	scalar	I4	I	8

name of the common block: RECS									
name	dimension	type	mode	word	name	dimension	type	mode	word
MAT	scalar	I4	-	1	L1	scalar	I4	I	6
MF	scalar	I4	-	2	L2	scalar	I4	I	7
MT	scalar	I4	-	3	N1	scalar	I4	IO	8
C1	scalar	R4	I	4	N2	scalar	I4	IO	9
C2	scalar	R4	I	5					

Called subprograms:

ANNIO1
CHGINT
ERF
ERROR
GENER (entry of SCALE)
GNRL
INITSC (entry of SCALE)
PRIZEI
RREC

2.3.5.26 Subroutine GNRL

Problem solved:
Calculation of the fluctuation integrals for capture, fission, and elastic scattering. The subroutine is called by ANNICK.

Definition:
SUBROUTINE GNRL(GALPHA,GBETA,GAMMA,MU,NU,LAMBDA,S,DF,ID)

Explanation of the arguments:

name	mode	type	dimension	remarks
GALPHA	I	R4	scalar	$\bar{\Gamma}_n$
GBETA	I	R4	scalar	$\bar{\Gamma}_f$
GAMMA	I	R4	scalar	$\bar{\Gamma}_\gamma$
MU	I	I4	scalar	number of degrees of freedom used in the neutron width distribution

name	mode	type	dimension	remarks
NU	I	I4	scalar	number of degrees of freedom used in the fission width distribution
LAMBDA	I	I4	scalar	number of degrees of freedom used in the competitive width distribution
S	O	R4	scalar	fluctuation integral value for elastic scattering (ID=1) capture (ID=2) fission (ID=3)
DF	I	R4	scalar	$\bar{\Gamma}_x$
ID	I	I4	scalar	see description of S

Used common block:

name of the common block: XBAR				
name	dimension	type	mode	word
XX	(4,10)	R4	I	1

Called subprograms:

None

2.3.5.27 Subroutine ERF

Problem solved:

Part of the calculation of the Dresner factors S_f, S_γ, R_f, R_γ (see /4/, /8/ and part II.). Subdivides the integration range in parts of 0.05 and stops the integration process if a integral over such an interval is less than 10^{-6}.

Definition:

SUBROUTINE ERF(EER)

Explanation of the argument:

name	mode	type	dimension	remarks
ERR	O	R4	scalar	result of the integration

Used common block:

name of the common block: FLCT									
name	dimension	type	mode	word	name	dimension	type	mode	word
ZYN	scalar	R4	I	1	V	scalar	R4	-	4
ZYF	scalar	R4	-	2	RHON	scalar	R4	I	5
U	scalar	R4	-	3	RHOF	scalar	R4	-	6

Called subprogram:

GAUSS

2.3.5.28 Subroutine GAUSS

Problem solved:

Numerical integration with a Gauss method with five points (see /7/).

Definition:

SUBROUTINE GAUSS(HU,HO,B)

Explanation of the arguments:

name	mode	type	dimension	remarks
HU	I	R4	scalar	lower integration limit
HO	I	R4	scalar	upper integration limit
B	O	R4	scalar	result of the integration

Used common **blocks**:

None

Called subprogram:

BINT

2.3.5.29 Function BINT

Problem solved:
Used for the calculation of the Dresner factors S_f, S_γ, R_f, R_γ (see /4/, /8/ and II.). Calculation of the function to be integrated.

Definition
FUNCTION BINT(X)

Explanation of the argument:

name	mode	type	dimension	remarks
X	I	R4	scalar	value for which the functional value is to be calculated

Used common block:

name of the common block: FLCT									
name	dimension	type	mode	word	name	dimension	type	mode	word
ZYN	scalar	R4	I	1	V	scalar	R4	I	4
ZYF	scalar	R4	I	2	RHON	scalar	R4	I	5
U	scalar	R4	I	3	RHOF	scalar	R4	I	6

Called subprograms:
None

2.3.5.30 Subroutine ANNIO1

Problem solved:
Optionally printing of the unresolved resonance parameters in KEDAK representation and of the pointwise cross sections. Storing of the unresolved resonance parameter blocks on logical unit 11 and of the pointwise cross sections on logical unit 12.

Definition:

```
SUBROUTINE ANNIO1(NE,SIGT,SIGG,SIGS,SIGF,NIS,MIS,E,NB,BK,ISTGF,NB2,
                  BK2,LFCHG,NCU,ESC,IDIMBK,INTS,NBT,JNT,Y,EL,EH,LFW)
```

Explanation of the arguments:

name	mode	type	dimension	remarks
NE	I	I4	scalar	number of the energy points of the pointwise cross sections in the unresolved resonance region
SIGT	L	R4	(NPEMAX)	see 2.3.5.25 argument list
SIGG	I	R4	(NPEMAX)	" "
SIGS	I	R4	(NPEMAX)	" "
SIGF	I	R4	(NPEMAX)	" "
NIS	I	I4	scalar	number of isotopes
MIS	I	I4	scalar	pointer to the actual isotope, if NIS is greater than 1
E	I	R4	(NPEMAX)	see 2.3.5.25 argument list
NB	I	I4	scalar	number of values in the array BK
BK	I	R4	(NBKX)	unresolved resonance parameters
ISTGF	I	I4	scalar	= 0 no fission widths given = 1 fission widths given
NB2	I	I4	scalar	number of values in the array BK2
BK2	I	R4	(N2X)	data for KEDAK-type STGF(I) and pointwise cross sections (0). (BK2 and X have the same starting address.)
LFCHG	I	I4	scalar	= 0 all parameters energy independant = 1 only fission-widths are energy dependant = 2 all parameters are energy dependant
NCU	I	I4	scalar	number of values in the array ESC
ESC	I	R4	(NPEMAX)	see 2.3.5.25 argument list
IDIMBK	I	I4	scalar	IDIMBK = NBKX
INTS	I	I4	(18)	interpolation codes for the unresolved parameters

name	mode	type	dimension	remarks
NBT	L	I4	(N1X)	interpolation ranges
JNT	L	I4	(N1X)	interpolation codes
Y	L	R4	(N2X)	pointwise cross section values
EL	I	R4	scalar	see 2.3.5.25 argument list
EH	I	R4	scalar	" "
LFW	I	I4	scalar	" "

Used common blocks:

name of the common block: INOUT									
name	dimension	type	mode	word	name	dimension	type	mode	word
NO	scalar	I4	I	1	NTYP	scalar	I4	-	84
NI	scalar	I4	-	2	NAMISO	(10)	R8	I	85
NT	scalar	I4	-	3	MATISO	(10)	I4	I	105
MODE	scalar	I4	-	4	NR2	(11)	I4	-	115
LEDIT1	scalar	I4	-	5	NBX	scalar	I4	-	126
LEDIT2	scalar	I4	I	6	N1X	scalar	I4	-	127
LEDITS	scalar	I4	I	7	N2X	scalar	I4	-	128
NR1	(75)	I4	-	8	N	scalar	I4	O	129
MATKDK	scalar	I4	I	83	NBKX	scalar	I4	I	130

name of the common block: RECS									
MAT	scalar	I4	-	1	L1	scalar	I4	-	6
MF	scalar	I4	-	2	L2	scalar	I4	-	7
MT	scalar	I4	-	3	N1	scalar	I4	O	8
C1	scalar	R4	-	4	N2	scalar	I4	O	9
C2	scalar	R4	-	5					

Called subprograms:

A8FORM
ERROR
LUCY
MAGGY
TERP1
WDA2 (entry of WDA)
WRECS2 (entry of WRECS)
XTEXT

2.3.5.31 Subroutine MAGGY

Problem solved:
The contents of the arrays XD and YD are stored in the arrays X and Y.

Definition:
SUBROUTINE MAGGY(XD,YD,N,X,Y)

Explanation of the arguments:

name	mode	type	dimension	remarks
XD	I	R4	(N)	input array
YD	I	R4	(N)	" "
N	I	I4	scalar	number of data in the arrays
X	O	R4	(N)	output array
Y	O	R4	(N)	" "

Used common blocks:
None

Called subprograms:
None

2.3.5.32 Subroutine DENISE

Problem solved:
The pointwise cross section data calculated out of resolved and unresolved resonance parameters are fetched from logical unit 12. The data for different isotopes - if existant - are added and stored on logical unit 12. The total cross section in the resolved and unresolved resonance region is calculated and stored on logical unit 12. Optional printout of the pointwise cross section is done.

Definition:
SUBROUTINE DENISE(MATP,LRP,LISRES,LISUNR,NIS,LFW,LNU,BK,NBT,JNT,X,Y,B,A)

Explanation of the arguments:

name	mode	type	dimension	remarks
MATP	I	I4	scalar	ENDF material number
LRP	-	I4	scalar	not used in DENISE
LISRES	I	I4	scalar	= 1 no resolved pointwise cross sections - calculated out of parameters - available ≠ 1 resolved pointwise cross sections - calculated out of parameters - are available
LISUNR	I	I4	scalar	= 1 no unresolved pointwise cross sections - calculated out of parameters - available ≠ 1 unresolved pointwise cross sections - calculated out of parameters - are available
NIS	I	I4	scalar	number of isotopes
LFW	I	I4	scalar	= 0 no pointwise fission cross sections - calculated out of parameters - available = 1 pointwise fission cross sections - calculated out of parameters - are available in the unresolved resonance region
LNU	I	I4	scalar	see 2.3.5.10 argument list
BK	P	R4	(NBKX)	see 2.3.5.36 argument list

name	mode	type	dimension	remarks
NBT	L	I4	(N1X)	interpolation ranges
JNT	L	I4	(N1X)	interpolation codes
X	L	R4	(N2X)	(mainly energy values for cross sections)
Y	L	R4	(N2X)	(mainly cross section values)
B	L	R4	(NBX)	see subroutines called by DENISE, in which B is used
A	L	R4	(IADIM)	see subroutines called by DENISE, in which A is used

Used common blocks:

name of the common block: DENS									
name	dimension	type	mode	word	name	dimension	type	mode	word
NR3	(402)	I4	-	1	MX	scalar	I4	0	403

name of the common block: INOUT									
NO	scalar	I4	I	1	NBKX	scalar	I4	I	130
NR1	(5)	I4	-	2	NR4	(10)	I4	-	131
LEDITS	scalar	I4	I	7	EPS	scalar	R4	I	141
NR2	(122)	I4	-	8					

Called subprograms:

DELETE
IFETCH (entry of FETCH)
IFPDS (entry of FPDS)
IICOMB (entry of COMB)
IIPDS (entry of IPDS)
IRRECS (entry of RRECS)
ISTORE (entry of STORE)
MARION
MYRIAM
SOPHIE
WREC
WRECS2 (entry of WRECS)
XTEXT

2.3.5.33 Subroutine MYRIAM

Problem solved:

The pointwise cross sections calculated out of resolved resp. unresolved resonance parameters are added up for the different isotopes with respect to the abundance. The result is stored in the arrays X and Y.

Definition:

SUBROUTINE MYRIAM(NIS,LISRES,LISUNR,NAME,EL1,EH1,EPS,ITEST,Y)

Explanation of the arguments:

name	mode	type	dimension	remarks
NIS	I	I4	scalar	number of isotopes
LISRES	I	I4	scalar	only for LISRES=0 adding up is done in the resolved resonance region
LISUNR	I	I4	scalar	only for LISUNR=0 adding up is done in the unresolved resonance region
NAME	I	I4	scalar	type of the cross section (SCAT,CAPT,FISS)
EL1	O	R4	scalar	lower energy limit for the resolved resp. the unresolved resonance region
EH1	O	R4	scalar	upper energy limit for the resolved resp. the unresolved resonance region
EPS	I	R4	scalar	see 2.3.5.66 argument list
ITEST	I	I4	scalar	≠ 1 resolved cross sections are processed = 1 unresolved cross sections are processed
Y	L	R4	(N2X)	cross section values for the different isotopes

Used common blocks:

None

Called subprograms:

ADD
BETSY
COMB
ERROR
FETCH
RRECS
STORE

2.3.5.34 Subroutine BETSY

Problem solved:
The pointwise cross sections values are multiplicated with the abundance associated to the processed isotope.

Definition:
SUBROUTINE BETSY(MIS,Y)

Explanation of the arguments:

name	mode	type	dimension	remarks
MIS	I	I4	scalar	pointer to the isotope, which is to be processed
Y	IO	R4	(N2X)	cross section values

Used common blocks:

name of the common block: RECS									
name	dimension	type	mode	word	name	dimension	type	mode	word
NR2	(8)	I4	-	1	N2	scalar	I4	I	9
name of the common block: INOUT									
NR1	(8)	I4	-	1	ABIS	(20)	R4	I	9

Called subprograms:

None

2.3.5.35 Subroutine MARION

Problem solved:

The total cross section in the resolved (NAME2='RES') or the unresolved (NAME2='STAT') resonance region is calculated by adding σ_e, σ_γ and σ_f if existant.

Definition:

SUBROUTINE MARION(LFW,EL,EH,EPS,NAME2)

Explanation of the arguments:

name	mode	type	dimension	remarks
LFW	I	I4	scalar	σ_f is only added if LFW=1
EL	I	R4	scalar	lower energy limit for the resolved resp. the unresolved resonance region
EH	I	R4	scalar	upper energy limit for the resolved resp. the unresolved resonance region
EPS	IO	R4	scalar	see 2.3.5.66 argument list Accuracy used for calculating σ_t
NAME2	I	I4	scalar	= 'RES' cross sections of the resolved resonance region are processed = 'STAT' cross sections of the unresolved resonance region are processed

Used common blocks:

name of the common block: INOUT				
name	dimension	type	mode	word
NO	scalar	I4	I	1

Called subprograms:

ADD

COMB

ERROR

FETCH

RRECS

STORE

2.3.5.36 Subroutine SOPHIE

Problem solved:

Processing of ENDF/B MF=3 data. First following ENDF/B data types from MF=3 with the MT-numbers listed below are searched, and read if found (the names in brackets are the corresponding KEDAK names):

1(SGT),2(SGN),3(SGX),4(SGI),16(SG2N),17(SG3N),18(SGF),22(SGIA),23(SGI3A), 24(SG2NA),25(SG3NA),27(SGA),28(SGIP),29(SGI2A),51...90(SGIZ),91(SGIZC), 102(SGG),103(SGP),104(SGD),105(SGH3),106(SGHE3),107(SGALP),108(SG2HE), 201(SGTR),206(ETA),207(ALPHA),251(MUEL).

In the case of SGT,SGN,SGF and SGG the resolved and unresolved resonance part - if existant - is added to the pointwise cross sections. The pointwise cross sections are stored on logical unit number 12.

Optionally (only used for an old KEDAK version) for KEDAVE=0 a common energy grid for SGI, all SGIZ, and SGIZC is generated after changing the interpolation code of all this cross sections to linear-linear. All cross section values for the types SGI, all SGIZ, and SGIZC are calculated and stored at this energy values.

If no absorption cross section has been found on ENDF/B it is generated by adding all the following existing cross section types:

SGG,SGF,SGP,SGD,SGH3,SGHE3,SGALP,SG2HE.

If possible SGTR is calculated using the formulae SGTR=SGT-MUEL*SGN. If the $\bar{\nu}$-data, read in the subroutine INGRID, had been given in polynomial representation, they will be calculated pointwise and stored on logical unit number 12.

Definition:

```
SUBROUTINE SOPHIE(MATP,LISRES,LISUNR,LFW,LNU,EPS,BK,XIN,YIN,NCMAX,
                  NBT,JNT,X,Y,B,TEMP)
```

Explanation of the arguments:

name	mode	type	dimension	remarks
MATP	I	I4	scalar	ENDF/B material number
LISRES	I	I4	scalar	flag for the existance of pointwise cross sections calculated out of resolved resonance parameters = 1 no data available
LISUNR	I	I4	scalar	flag for the existance of pointwise cross sections calculated out of unresolved resonance parameters = 1 no data available
LFW	I	I4	scalar	flag for the existance of pointwise fission cross sections calculated out of resonance parameters = 1 data available
LNU	I	I4	scalar	for LNU=1 pointwise $\bar{\nu}$-data are calculated out of the polynomial representation
EPS	I	R4	scalar	accuracy used in the subroutines COMB (see 2.3.5.66) and CHGINT (see 2.3.5.73)
BK	-	R4	(NBKX)	not used in SOPHIE
XIN	L	R4	(NCMAX)	used for the common energy grid for the total inelastic and the inelastic excitation cross sections (BK(1)...BK(NBKX/2)
YIN	L	R4	(NCMAX)	used for the cross section values of SGI,SGIZ and SGIZC calculated at the energy points, which are stored in XIN (BK(NBKX/2+1)...BK(NBKX)
NCMAX	I	I4	scalar	NCMAX = NBKX/2
NBT	L	I4	(N1X)	interpolation ranges
JNT	L	I4	(N1X)	interpolation codes
X	L	R4	(N2X)	energy values
Y	L	R4	(N2X)	cross section values
B	L	R4	(NBX)	polynomial coefficients for $\bar{\nu}$
TEMP	L	R4	(NBX)	(array B) auxiliary array used in GENER (see 2.3.5.65)

Used common blocks:

name of the common block: RECS									
name	dimension	type	mode	word	name	dimension	type	mode	word
MAT	scalar	I4	-	1	L1	scalar	I4	-	6
MF	scalar	I4	0	2	L2	scalar	I4	-	7
MT	scalar	I4	0	3	N1	scalar	I4	0	8
C1	scalar	R4	-	4	N2	scalar	I4	IO	9
C2	scalar	R4	I	5					

name of the common block: INOUT									
NO	scalar	I4	I	1	NOTTRL	(50)	I4	I	29
NI	scalar	I4	-	2	BIB12	scalar	I4	-	79
NT	scalar	I4	I	3	SALINE	scalar	L4	I	80
MODE	scalar	I4	I	4	NR2	(2)	I4	-	81
LEDIT1	scalar	I4	-	5	MATKDK	scalar	I4	-	83
LEDIT2	scalar	I4	-	6	NR3	(42)	I4	-	84
LEDITS	scalar	I4	I	7	NBX	scalar	I4	-	126
KEDAVE	scalar	I4	I	8	N1X	scalar	I4	I	127
NR1	(20)	I4	-	9	N2X	scalar	I4	I	128

name of the common block: TAB									
ENDTAB	(25)	I4	I	1	NCNAME	(39)	I4	-	316
KEDTAB	(50)	I4	I	26	NRIF3	scalar	I4	IO	355
KEDR	(240)	I4	-	76	NRIF4	scalar	I4	-	356

Called subprograms:

ADD
CHGINT
COMB
ERROR
GENER } (entries of SCALE)
INITSC }
ITAB1
MULT
PRIZEI
RREC
RRECS
RRECS4 (entry of RRECS)
SEARCH
STORE
SUB
WREC
WRECS2 (entry of WRECS)
XTEXT

2.3.5.37 Subroutine SUZY

Problem solved:

The cross sections, which have been written by the subroutine SOPHIE (see 2.3.5.36) on logical unit number 12, are read. The interpolation is changed to linear-linear, points with zero cross section values are suppressed, except those points, which are preceding the threshold of a cross section. At the threshold also zeros may be generated. Points, which can be predicted by linear interpolation, are deleted. Cross section values with energy values, which fulfill the condition $(E_2-E_1)/E_2<5.0*10^{-6}$, are replaced by mean values. Points with negative cross section values for the types σ_t or σ_e are deleted. The result is printed out and written together with KEDAK type names on logical unit number 11. Optionally it can also be written in ENDF/B-format on logical unit number 15.

Definition:

SUBROUTINE SUZY(LNU,ITAPE,BK,A,TAB,ITAB,NBT,JNT,X,Y,B)

Explanation of the arguments:

name	mode	type	dimension	remarks
LNU	-	I4	scalar	not used in SUZY
ITAPE	I	I4	scalar	control of writing on unit 10 = 1 the data produced in the subroutine are written on the external unit 10 ≠ 1 no writing on unit 10 is done
BK	L	R4	(NBKX)	energy grid and cross sections which will be written in WDA2 on an external unit
A	L	R4	(NBKX)	(same starting address as BK) same contents as BK
TAB	L	R4	(NBKX)	(same starting address as BK) used for building up the KEDAK further name address table for the excitated inelastic cross section
ITAB	L	I4	(NBKX)	(same starting address as BK) same contents as TAB
NBT	L	I4	(N1X)	interpolation ranges
JNT	L	I4	(N1X)	interpolation codes
X	L	R4	(N2X)	energy grids
Y	L	R4	(N2X)	cross section values
B	P	R4	(NBX)	address is passed to the subroutine RREC

Used common blocks:

name of the common block: INOUT									
name	dimension	type	mode	word	name	dimension	type	mode	word
NO	scalar	I4	I	1	NR4	(42)	I4	-	84
NI	scalar	I4	-	2	NBX	scalar	I4	-	126
NT	scalar	I4	-	3	N1X	scalar	I4	I	127
MODE	scalar	I4	I	4	N2X	scalar	I4	I	128
NR1	(3)	I4	-	5	N	scalar	I4	O	129
KEDAVE	scalar	I4	I	8	NBKX	scalar	I4	I	130
NR2	(71)	I4	-	9	NR5	(10)	I4	-	131
SALINE	scalar	L4	I	80	EPS	scalar	R4	I	141
NR3	(2)	I4	-	81	SCREPS	scalar	R4	I	142
MATKDK	scalar	I4	I	83					

name of the common block: RECS									
MAT	scalar	I4	I	1	L1	scalar	I4	O	6
MF	scalar	I4	O	2	L2	scalar	I4	O	7
MT	scalar	I4	O	3	N1	scalar	I4	O	8
C1	scalar	R4	O	4	N2	scalar	I4	O	9
C2	scalar	R4	O	5	NS	scalar	I4	O	10

name of the common block: TAB									
ENDTAB	(25)	I4	I	1	NCNAME	(39)	I4	-	316
KEDTAB	(50)	I4	I	26	NRIF3	scalar	I4	I	355
KEDR	(240)	I4	-	76	NRIF4	scalar	I4	-	356

Called subprograms:

ADDPNT
A8FORM
CHGINT
CROP
DZMSCH
ELIMZ
ERROR
IRRECS (entry of RRECS)
PRIZEI
RRECS
SUPNEG
WCONT
WDA2 (entry of WDA)
WREC
XTEXT

2.3.5.38 Subroutine ELIMZ

Problem solved:
Points with y = 0.0 are suppressed, except the point preceding the threshold of the cross section. (The interpolation scheme must be linear-linear.)

Definition:
SUBROUTINE ELIMZ(N1,N2,NBT,JNT,X,Y)

Explanation of the arguments:

name	mode	type	dimension	remarks
N1	I	I4	scalar	must have the value 1
N2	IO	I4	scalar	number of points in the arrays X and Y
NBT	IO	I4	(N1X)	interpolation ranges. NBT(1) must be equal N2
JNT	I	I4	(N1X)	interpolation codes. JNT(1) must be equal 2 or equal 1
X	IO	R4	(N2X)	energy grid
Y	IO	R4	(N2X)	cross section values

Used common blocks:

None

Called subprograms:

ERROR

2.3.5.39 Subroutine ADDPNT

Problem solved:

For threshold reactions two zero cross section values at the beginning of the energy range must exist. A threshold reaction is assumed, if the first energy point is greater than $1.0 \cdot 10^{-5}$ eV and if the first cross section value is less or equal $1.0 \cdot 10^{-20}$. The subroutine generates such cross section values, if they don't exist. If the first cross section value is less or equal $1.0 \cdot 10^{-20}$ it is set to zero. (Interpolation must be linear-linear.)

Definition:

SUBROUTINE ADDPNT(X,Y,NBT,JNT,MT)

Explanation of the arguments:

name	mode	type	dimension	remarks
X	IO	R4	(N2X)	energy grid
Y	IO	R4	(N2X)	cross section values
NBT	IO	I4	(N1X)	interpolation ranges. NBT(1) must be equal N2
JNT	I	I4	(N1X)	interpolation codes. JNT(1) must be less or equal 2
MT	I	I4	scalar	ENDF/B section number

Used common blocks:

name of the common block: RECS									
name	dimension	type	mode	word	name	dimension	type	mode	word
NR1	(7)	I4	-	1	N2	scalar	I4	IO	9
N1	scalar	I4	I	8					

name of the common block: INOUT									
NO	scalar	I4	I	1	N2X	scalar	I4	I	128
NR2	(126)	I4	-	2					

Called subprograms:
ERROR

2.3.5.40 Subroutine DZMSCH

Problem solved:

Cross section values at energies with 6 similar digits will be replaced by mean values:

$$XNEW = 1/N*(X1+X2+...+XN)$$
$$YNEW = 1/N*(Y1+Y2+...+YN)$$

The accuracy with which the replaced points could have been interpolated is determined and ordered in classes.

Definition:
SUBROUTINE DZMSCH(X,Y,NBT,JNT,NDZ,EPS,SCRCLA,NUMSCR)

Explanation of the arguments:

name	mode	type	dimension	remarks
X	IO	R4	(N2X)	energy grid
Y	IO	R4	(N2X)	cross section values
NBT	IO	I4	(N1X)	interpolation ranges. NBT(1) must be equal N2
JNT	I	I4	(N1X)	interpolation codes. JNT(1) must be less or equal 2
NDZ	O	I4	scalar	total number of scratched values
EPS	I	R4	scalar	used for the construction of the scratch classes
SCRCLA	O	R4	(10)	scratch classes. Accuracies, with which the replaced points could have been interpolated (see 2.2.7).
NUMSCR	O	I4	(10)	number of points in a specified scratch class

Used common block:

name of the common block: RECS									
name	dimension	type	mode	word	name	dimension	type	mode	word
NR1	(7)	I4	-	1	N2	scalar	I4	IO	9
N1	scalar	I4	I	8					

Called subprograms:

ERROR

2.3.5.41 Subroutine SUPNEG

Problem solved:

For the cross section types σ_t and σ_e negative cross section values (mainly in the resolved resonance region) are diagnosed and optionally eliminated.

Definition:

SUBROUTINE SUPNEG(MT,N1,N2,NBT,JNT,X,Y,)

Explanation of the arguments:

name	mode	type	dimension	remarks
MT	I	I4	scalar	ENDF/B section number. Must be equal 1 or 2
N1	I	I4	scalar	number of interpolation ranges. Must be equal 1, if points must be eliminated
N2	IO	I4	scalar	number of energy points resp. cross section values
NBT	IO	I4	(N1X)	interpolation ranges. NBT(1) must be equal N2, if points must be eliminated
JNT	I	I4	(N1X)	interpolation codes. JNT(1) must be less or equal 2, if points must be eliminated
X	IO	R4	(N2X)	energy grid
Y	IO	R4	(N2X)	cross section values

Used common block:

name of the common block: INOUT									
name	dimension	type	mode	word	name	dimension	type	mode	word
NO	scalar	I4	I	1	ELINEG	scalar	I4	I	154
NR1	(152)	I4	-	2					

Called subprogram:

ERROR

2.3.5.42 Subroutine ISABEL

Problem solved:
Processing of ENDF/B MF=4 data (angular distributions of secondary neutrons).

Only elastic angular distributions are processed. If the distributions are given in tabulated form, only these data (SGNC or SGNL see /4/, /5/ or II) are written on logical unit number 11 for later insertion into KEDAK. If the data are given as Legendre polynomial coefficients, these coefficients are written as LEGNC or LEGNL (see /4/, /5/ or II) on logical unit number 11 together with the angular pointwise data calculated out of the coefficients (81 cosine values are used for each energy point).

Definition:
SUBROUTINE ISABEL(MATP,IF4,ITAB,TAB,CLEG,NBT,JNT,X,Y,B,A,ITAB2,TAB2)

Explanation of the arguments:

name	mode	type	dimension	remarks
MATP	I	I4	scalar	ENDF/B-number of the material to be converted
IF4	I	I4	scalar	see 2.2.6. If IF4=1 only ENDF/B-data of MF=4 are converted. Then ISABEL is the first subroutine writing on file 11. Then this file is initialized in this subroutine
ITAB	L	I4	(NBKX)	(array BK) used for storing Legendre polynomial coefficients in KEDAK representation
TAB	L	R4	(NBKX)	(array BK) see ITAB
CLEG	L	R4	(NBKX)	(array BK) see ITAB
NBT	L	I4	(N1X)	interpolation ranges
JNT	L	I4	(N1X)	interpolation codes
X	L	R4	(N2X)	cosine values
Y	L	R4	(N2X)	distribution probabilities
B	L	R4	(NBX)	Legendre coefficients
A	L	R4	(IADIM)	used for storing tabulated distribution probabilities in KEDAK representation

name	mode	type	dimension	remarks
ITAB2	L	I4	(IADIM)	(array A) see A
TAB2	L	R4	(IADIM)	(array A) see A

Used common blocks:

name of the common block: INOUT									
name	dimension	type	mode	word	name	dimension	type	mode	word
NO	scalar	I4	I	1	NR2	(45)	I4	-	84
NI	scalar	I4	-	2	N	scalar	I4	L	129
NT	scalar	I4	I	3	NBKX	scalar	I4	I	130
MODE	scalar	I4	I	4	JX	scalar	I4	-	131
NR1	(78)	I4	-	5	IADIM	scalar	I4	I	132
MATKDK	scalar	I4	I	83					

name of the common block: RECS									
MAT	scalar	I4	-	1	L1	scalar	I4	I	6
MF	scalar	I4	-	2	L2	scalar	I4	I	7
MT	scalar	I4	-	3	N1	scalar	I4	IO	8
C1	scalar	R4	-	4	N2	scalar	I4	IO	9
C2	scalar	R4	I	5					

name of the common block: TAB									
ENDTAB	(25)	I4	-	1	NCNAME	(39)	I4	O	316
KEDTAB	(50)	I4	-	26	NRIF3	scalar	I4	-	355
KEDR	(240)	I4	I	76	NRIF4	scalar	I4	O	356

Called subprograms:

A8FORM

ERROR

ITAB1

LEGP

PRIZEI

RREC

SEARCH

WREC

2.3.5.43 Function LEGP

Problem solved:

Calculation of Legendre polynomials.

Definition:

REAL FUNCTION LEGP(X,NL,CON)

Explanation of the arguments:

name	mode	type	dimension	remarks
X	I	R4	scalar	cosine value
NL	I	I4	scalar	order of the Legendre polynom.
CON	I	R4	(NL)	Legendre polynomial coefficients

Used common blocks:

None

Called subprograms:

None

2.3.5.44 Subroutine FILE5

Problem solved:
Processing of ENDF/B MF=5 data (energy distributions of secondary neutrons).

Following cross section types can be produced by the subroutine

KEDAK-names	ENDF section numbers
SED2N	MT = 16
SED3N	MT = 17
SEDF	MT = 18
SEDIC	MT = 91

These types can be produced, if the ENDF-representation is LF = 3 (K = 4 excitation of discrete levels), LF = 7 (K = 2 Maxwellian spectrum), LF = 9 (K = 1 evaporation spectrum) or LF = 10 (K = 3 Watt spectrum). (The values K refer to the KEDAK data types.) All interpolation is changed to linear-linear. The result of the subroutine is written on logical unit number 11 in a representation, which is used to built up the KEDAK-file.

Definition:
SUBROUTINE FILE5(MATP,IF5,BK,IBK,NBT,JNT,X,Y,B,TEMP)

Explanation of the arguments:

name	mode	type	dimension	remarks
MATP	I	I4	scalar	ENDF/B-number of the material, which is to be converted
IF5	-	I4	scalar	not used in FILE5
BK	L	R4	(NBKX)	used for building up the energy distributions in a KEDAK similar representation
IBK	L	I4	(NBKX)	(array BK) see BK
NBT	L	I4	(N1X)	interpolation ranges
JNT	L	I4	(N1X)	interpolation codes
X	L	R4	(N2X)	energy grids

name	mode	type	dimension	remarks
Y	L	R4	(N2X)	distributions or probabilities
B	L	R4	(NBX)	used for LF=10 (Watt spectrum) for a and b (see /2/ pages 10.3 and 10.8)
TEMP	P	R4	(NBX)	(array B) auxiliary array

Used common blocks:

name of the common block: INOUT									
name	dimension	type	mode	word	name	dimension	type	mode	word
NO	scalar	I4	I	1	NBX	scalar	I4	-	126
NI	scalar	I4	-	2	N1X	scalar	I4	I	127
NT	scalar	I4	I	3	N2X	scalar	I4	I	128
MODE	scalar	I4	I	4	N	scalar	I4	0	129
NR1	(78)	I4	-	5	NBKX	scalar	I4	I	130
MATKDK	scalar	I4	I	83	NR3	(10)	I4	-	131
NR2	(42)	I4	-	84	EPS	scalar	R4	I	141

name of the common block: RECS									
MAT	scalar	I4	-	1	L1	scalar	I4	-	6
MF	scalar	I4	-	2	L2	scalar	I4	I	7
MT	scalar	I4	-	3	N1	scalar	I4	I	8
C1	scalar	R4	I	4	N2	scalar	I4	I	9
C2	scalar	R4	I	5					

name of the common block: TAB									
ENDTAB	(25)	I4	-	1	NCNAME	(39)	I4	0	316
KEDTAB	(50)	I4	-	26	NRIF3	scalar	I4	-	355
KEDR	(240)	I4	I	76	NRIF4	scalar	I4	IO	356

Called subprograms:

A8FORM
CHGINT
ERROR
GENER } (entries of SCALE)
INITSC }
ITAB1
PRIZEI
RREC
SEARCH
WDA2 (entry of WDA)
WREC

2.3.5.45 Subroutine PRIZEI

Problem solved:
Prints date, elapsed time and CPU-time.

Definition:
SUBROUTINE PRIZEI

Used common block:

name of the common block: INOUT									
name	dimension	type	mode	word	name	dimension	type	mode	word
NO	scalar	I4	I	1	ANFZEI	scalar	R4	I	153
NR1	(151)	I4	-	2					

Called subprograms:

DATUM
ZEIT

2.3.5.46 Subroutine ERROR

Problem solved:

Prints the error number N, the quantities MAT, MF, and MT of the common RECS and the contents of NBT,JNT,X, and Y. Afterwards STOP1 is made.

Definition:

SUBROUTINE ERROR(N)

Explanation of the argument:

name	mode	type	dimension	remarks
N	I	I4	scalar	error number

Used common blocks:

name of the common block: RECS									
name	dimension	type	mode	word	name	dimension	type	mode	word
MAT	scalar	I4	I	1	MT	scalar	I4	I	3
MF	scalar	I4	I	2					

name of the common block: INOUT									
NOUT	scalar	I4	I	1					

Called subprogram:

WREC

2.3.5.47 Subroutine XTEXT

Problem solved:
The value of I is assigned to J. The contents of J is normally text of the form 'ABCD'.

Definition:
SUBROUTINE XTEXT(I,J)

Explanation of the arguments:

name	mode	type	dimension	remarks
I	I	I4	scalar	see problem solved
J	O	I4	scalar	see problem solved

Used common blocks:
None

Called subprograms:
None

2.3.5.48 Subroutine SEARCH

Problem solved:
Search ENDF/B tape NT for a particular section denoted by MATP,MFP,MTP. Search is in forward direction only. Zero values of MATP,MFP,MTP are read as (any). Thus, MATP,MFP,0 means any section in material MATP, file MFP. 0,MFP,0 means any (next) section in file MFP for any (next) material.

Definition:
SUBROUTINE SEARCH(NT,MODE,MATP,MFP,MTP,LNT)
(The subroutine is taken out of /3/ part R)

Explanation of the arguments:

name	mode	type	dimension	remarks
NT	I	I4	scalar	logical ENDF/B tape number
MODE	I	I4	scalar	tape mode. May be 1,2,3. (see /2/ chapter 4.)
MATP	I	I4	scalar	desired material number or zero
MFP	I	I4	scalar	desired file number or zero
MTP	I	I4	scalar	desired section number or zero
LNT	O	I4	scalar	= 0 section located. MAT,MF,MT stored in common RECS. = 1 section not located, because end of tape encountered, or because desired record is located prior to the current position of the tape

Used common block:

name of the common block: RECS									
name	dimension	type	mode	word	name	dimension	type	mode	word
MAT	scalar	I4	0	1	MT	scalar	I4	0	3
MF	scalar	I4	0	2					

Called subprograms:

ERROR

RREC

2.3.5.49 Subroutine RREC

Problem solved:

One record is read from the ENDF/B tape. The result is stored in the
common RECS for CONT-records
or in the
common RECS and the array B for
LIST-records
or in the
common RECS and the arrays NBT,JNT,
X,Y, for TAB1-records
or in the
common RECS and the arrays NBT,JNT,
for TAB2-records
or in the
common RECS and the array B for
HOL-records
or in the
quantities MAT,JF,MT of the common RECS
and the first 17 words of the array B
for TPID-records.

For explanation of the record-types see /2/ chapter 4.

Definition:
SUBROUTINE RREC(JT,NT,MODE,T)
ENTRY IRREC(NBT,JNT,X,Y,B)

(The entry is only used for passing the addresses of the arrays to the subroutine. It must be called before the first RREC-call.) (The subroutine has been taken with some changes out of /3/ part R.)

Explanation of the arguments:

name	mode	type	dimension	remarks
JT	I	I4	scalar	record type. May be 1,2,3,4,5,6. (see /2/)
NT	I	I4	scalar	logical ENDF/B-tape number
MODE	I	I4	scalar	ENDF/B-tape mode. May be 1,2,3 (see /2/)
T	I	R4	scalar	temperature (Kelvin). For T≥0 all data will be interpolated for temperature T, if there is a temperature dependance given in the record. For T<0 only the first record will be read
NBT	P+	I4	(N1X)	used to store retrieved data
JNT	P+	I4	(N1X)	" " " " "
X	P+	R4	(N2X)	" " " " "
Y	P+	R4	(N2X)	" " " " "
B	P+	R4	(NBX)	" " " " "

Used common blocks:

name of the common block: RECS									
name	dimension	type	mode	word	name	dimension	type	mode	word
MAT	scalar	I4	0	1	L1	scalar	I4	0	6
MF	scalar	I4	0	2	L2	scalar	I4	0	7
MT	scalar	I4	0	3	N1	scalar	I4	0	8
C1	scalar	R4	0	4	N2	scalar	I4	0	9
C2	scalar	R4	0	5					

+ The addresses are passed to the subroutine in the entry. These arrays will be filled by RREC-calls with respect to the rules given in the section "problem solved".

name of the common block: INOUT									
name	dimension	type	mode	word	name	dimension	type	mode	word
NR1	(125)	I4	-	1	N1X	scalar	I4	I	127
NBX	scalar	I4	I	126	N2X	scalar	I4	I	128

Called subprograms:

ERROR

TERP1

2.3.5.50 Subroutine WCONT

Problem solved:

Writes ENDF/B end-record on tape unit NT (see /2/ chapter 4.).

Definition:

SUBROUTINE WCONT(NT,MODE,MAT,MF,NS)

Explanation of the arguments:

name	mode	type	dimension	remarks
NT	I	I4	scalar	logical tape unit number
MODE	I	I4	scalar	ENDF/B-mode. May be 1,2,3. (see /2/ chapter 4.)
MAT	I	I4	scalar	ENDF/B mat number or zero or -1 (see /2/ chapter 4.)
MF	I	I4	scalar	ENDF/B file number or zero (see /2/ chapter 4.)
NS	I	I4	scalar	sequence number written in card columns 76-80

Used common blocks:

None

Called subprograms:

None

2.3.5.51 Subroutine WREC

Problem solved:
Writes one record in ENDF representation. The data in the common RECS together with the contents of the arrays NBT,JNT,X,Y, and B (depending on the record type) are written in the ENDF representation according to the given mode.

The subroutine has been taken out of /3/ chapter R (see R-34 to R-35 and R-63) but some modifications have been made before inserting it in BRIGITTE.

E.g.: the arrays NBT,KNT,X,Y, and B, which have been of fixed lenght and placed in the common RECS, are now of variable length, and are passed to the subroutine by an entry.

Definition:
SUBROUTINE WREC(JT,NT,MODE)
ENTRY IWREC(NBT,JNT,X,Y,B)

(A call of the entry only passes the addresses of the arrays to the subroutine)

Explanation of the arguments:

name	mode	type	dimension	remarks
JT	I	I4	scalar	record type. May be 1,2,3,4,5,6. (see /3/ chapter R-2.3)
NT	I	I4	scalar	logical output tape number. For NT $\leq$ 0 writing is ignored
MODE	I	I4	scalar	arrangement of the data. May be 1,2,3,4 (see /3/ chapter R-2.2)
NBT	I	I4	(N1X)	data to be written according to the record type
JNT	I	I4	(N1X)	" "
X	I	R4	(N2X)	" "
Y	I	R4	(N2X)	" "
B	I	R4	(NBX)	" "

Used common block:

name of the common block: RECS									
name	dimension	type	mode	word	name	dimension	type	mode	word
MAT	scalar	I4	I	1	L1	scalar	I4	I	6
MF	scalar	I4	I	2	L2	scalar	I4	I	7
MT	scalar	I4	I	3	N1	scalar	I4	I	8
C1	scalar	R4	I	4	N2	scalar	I4	I	9
C2	scalar	R4	I	5					

Called subprograms:

ERROR
PRCONT
PRHOL
PRLIST
PRTAB1
PRTAB2
PRTPID
PUCONT
PUHOL
PULIST
PUTAB1
PUTAB2
PUTPID

2.3.5.52 Subroutine PRCONT

Problem solved:
Called by the subroutine WREC to do the expanded interpreted print of a CONT-record. The subroutine has been taken from /3/ part R with the modifications mentioned in 2.3.5.51 (Original routine see /3/ page R-66).

Definition:
SUBROUTINE PRCONT(NT)

Explanation of the argument:

name	mode	type	dimension	remarks
NT	I	I4	scalar	print output unit number

Used common block:

name of the common block: RECS									
name	dimension	type	mode	word	name	dimension	type	mode	word
MAT	scalar	I4	I	1	L1	scalar	I4	I	6
MF	scalar	I4	I	2	L2	scalar	I4	I	7
MT	scalar	I4	I	3	N1	scalar	I4	I	8
C1	scalar	R4	I	4	N2	scalar	I4	I	9
C2	scalar	R4	I	5	NS	scalar	I4	IO	10

Called subprogramms:

None

2.3.5.53 Subroutine PRLIST

Problem solved:

Called by the subroutine WREC to do the expanded interpreted print of a LIST-record. The subroutine has been taken from /3/ part R with the modifications mentioned in 2.3.5.51 (Original routine see /3/ page R-67).

Definition:

SUBROUTINE PRLIST(NT,B)

Explanation of the arguments:

name	mode	type	dimension	remarks
NT	I	I4	scalar	print output unit number
B	I	R4	(NBX)	data to be printed

Used common blocks:

See 2.3.5.52

Called subprograms:

None

2.3.5.54 Subroutine PRTAB1

Problem solved:

Called by the subroutine WREC to do the expanded interpreted print of a TAB1-record. The subroutine has been taken from /3/ part R with the modifications mentioned in 2.3.5.51 (Original routine see /3/ page R-68).

Definition:

SUBROUTINE PRTAB1(NT,NBT,JNT,X,Y)

Explanation of the arguments:

name	mode	type	dimension	remarks
NT	I	I4	scalar	print output unit number
NBT	I	I4	(N1X)	data to be printed
JNT	I	I4	(N1X)	" " " "
X	I	R4	(N2X)	" " " "
Y	I	R4	(N2X)	" " " "

Used common blocks:

See 2.3.5.52

Called subprograms:

None

2.3.5.55 Subroutine PRTAB2

Problem solved:
Called by the subroutine WREC to do the expanded interpreted print of a TAB2-record. The subroutine has been taken from /3/ part R with the modifications mentioned in 2.3.5.51 (Original routine see /3/ page R-70).

Definition:
SUBROUTINE PRTAB2(NT,NBT,JNT)

Explanation of the arguments:

name	mode	type	dimension	remarks
NT	I	I4	scalar	print output unit number
NBT	I	I4	(N1X)	data to be printed
JNT	I	I4	(N1X)	" " " "

Used common blocks:
See 2.3.5.52

Called subprograms:
None

2.3.5.56 Subroutine PRHOL

Problem solved:
Called by the subroutine WREC to do the expanded interpreted print of a HOL-record. The subroutine has been taken from /3/ part R with the modifications mentioned in 2.3.5.51 (Original routine see /3/ page R-65).

Definition:
SUBROUTINE PRHOL(NT,B)

Explanation of the arguments:

name	mode	type	dimension	remarks
NT	I	I4	scalar	print output unit number
B	I	R4	(NBX)	data to be printed

Used common blocks:

See 2.3.5.52

Called subprograms:

None

2.3.5.57 Subroutine PRTPID

Problem solved:

Called by the subroutine WREC to do the expanded interpreted print of a TPID-record. The subroutine has been taken from /3/ part R with the modifications mentioned in 2.3.5.51 (Original routine see /3/ page R-65).

Definition:

SUBROUTINE PRTPID(NT,B)

Explanation of the arguments:

name	mode	type	dimension	remarks
NT	I	I4	scalar	print output unit number
B	I	R4	(NBX)	data to be printed

Used common block:

name of the common block: RECS									
name	dimension	type	mode	word	name	dimension	type	mode	word
MAT	scalar	I4	I	1	NS	scalar	I4	IO	10
NR1	(8)	I4	-	2					

Called subprograms:

None

2.3.5.58 Subroutine PUCONT

Problem solved:
Called by the subroutine WREC to produce in a BCD card image tape a CONT-record. The subroutine has been taken from /3/ part R with the modifications mentioned in 2.3.5.51 (Original routine see /3/ page R-67).

Definition:
SUBROUTINE PUCONT(NT)

Explanation of the argument:

name	mode	type	dimension	remarks
NT	I	I4	scalar	logical tape unit number

Used common block:
See 2.3.5.52

Called subprogram:
CXFP

2.3.5.59 Subroutine PULIST

Problem solved:
Called by the subroutine WREC to produce in a BCD card image tape a LIST-record. The subroutine has been taken from /3/ part R with the modifications mentioned in 2.3.5.51 (Original routine see /3/ page R-67).

Definition:
SUBROUTINE PULIST(NT,B)

Explanation of the arguments:

name	mode	type	dimension	remarks
NT	I	I4	scalar	logical tape unit number
B	I	R4	(NBX)	data to be written on NT

Used common block:
See 2.3.5.52

Called subprogram :
CXFP

2.3.5.60 Subroutine PUTAB1

Problem solved:
Called by the subroutine WREC to produce in a BCD card image tape a TAB1-record. The subroutine has been taken from /3/ part R with the modifications mentioned in 2.3.5.51 (Original routine see /3/ page R-69).

Definition:
SUBROUTINE PUTAB1(NT,NBT,JNT,X,Y)

Explanation of the arguments:

name	mode	type	dimension	remarks
NT	I	I4	scalar	logical tape unit number
NBT	I	I4	(N1X)	data to be written on tape NT
JNT	I	I4	(N1X)	" "
X	I	R4	(N2X)	" "
Y	I	R4	(N2X)	" "

Used common block:
See 2.3.5.52

Called subprogram:
CXFP

2.3.5.61 Subroutine PUTAB2

Problem solved:
Called by the subroutine WREC to produce in a BCD card image tape a TAB2-record. The subroutine has been taken from /3/ part R with the modifications mentioned in 2.3.5.51 (Original routine see /3/ page R-71).

Definition:
SUBROUTINE PUTAB2(NT,NBT,JNT)

Explanation of the arguments:

name	mode	type	dimension	remarks
NT	I	I4	scalar	logical tape unit number
NBT	I	I4	(N1X)	data to be written on tape NT
JNT	I	I4	(N1X)	" "

Used common block:
See 2.3.5.52

Called subprogram:
CXFP

2.3.5.62 Subroutine PUHOL

Problem solved:
Called by the subroutine WREC to produce in a BCD card image tape a HOL-record. The subroutine has been taken from /3/ part R with the modifications mentioned in 2.3.5.51 (Original routine see /3/ page R-65).

Definition:
SUBROUTINE PUHOL(NT,B)

Explanation of the arguments:

name	mode	type	dimension	remarks
NT	I	I4	scalar	logical tape unit number
B	I	R4	(NBX)	data to be written on tape NT

Used common block:
See 2.3.5.52

Called subprogram:
CXFP

2.3.5.63 Subroutine PUTPID

Problem solved:
Called by the subroutine WREC to produce in a BCD card image tape a TPID-record. The subroutine has been taken from /3/ part R with the modifications mentioned in 2.3.5.51 (Original routine see /3/ page R-65).

Definition:
SUBROUTINE PUTPID(NT,B)

Explanation of the arguments:

name	mode	type	dimension	remarks
NT	I	I4	scalar	logical tape unit number
B	I	R4	(NBX)	data to be written on tape NT

Used common block:

name of the common block: RECS									
name	dimension	type	mode	word	name	dimension	type	mode	word
MAT	scalar	I4	I	1	NS	scalar	I4	IO	10
NR1	(8)	I4	-	2					

Called subprograms:
None

2.3.5.64 Subroutine CXFP

Problem solved:

Convert X for punching. X is represented as F*10.0**SN with

F $0.999995 \leq F \leq 9.999995$

S sign either (EBCDIC) + or - .

N exponent

(see /3/ page R-66).

Definition:

SUBROUTINE CXFP(X,F,S,N)

Explanation of the arguments:

name	mode	type	dimension	remarks
X	I	R4	scalar	number to be converted
F	O	R4	scalar	see problem solved
S	O	R4	scalar	" " "
N	O	I4	scalar	" " "

Used common blocks:

None

Called subprograms:

None

2.3.5.65 Subroutine SCALE

Problem solved:

Two or more energy scales are combined. If the combined scale is shorter than NCMAX it is kept in main storage, otherwise datasets NIN, NOUT are used (unit numbers 3 and 4). An energy point XNEW is added to the combined scale if no old points XOLD exists with

$$XOLD \in [XNEW \cdot 0.99999,\ XNEW \cdot 1.00001]$$

Definition:

SUBROUTINE SCALE (This entry point is never used.)

ENTRY INITSC(NCMAX) (Initialisation. Must be called before the first GENER-call and for deletion of a generated scale.)

ENTRY GENER(MAX,E,TEMP,EC,ND,NCU) (This entry creates the sacle.)

Explanation of the arguments:

name	mode	type	dimension	remarks
NCMAX	I	I4	scalar	maximal dimension of array E.
MAX	I	I4	scalar	number of data in array E.
E	I	R4	(MAX)	energy points to be inserted in array EC.
TEMP	L	R4	(NCMAX)	auxiliary array.
EC	IO	R4	(NCMAX)	complete energy scale.
ND	O	I4	scalar	0 or number of the unit where the complete scale can be found.
NCU	O	I4	scalar	number of energy points in array EC.

Used common blocks:

None.

Called subprograms:

None.

2.3.5.66 Subroutine COMB

Problem solved:

Combine two TAB1 functions (see /3/ chapters R-2.3.3, R-9.2 and page R-87 for explanation).
Two TAB1 functions stored in dense storage (see 2.3.5.80 to 2.3.5.85) are combined and the result is stored in the arrays NBT, JNT, X and Y. Sufficient energy points are generated in order to guarantee the interpolation of the produced TAB1 record accurate to EPS. If FA(X) is the first TAB1 function and FB(X) is the second TAB1 function, then the resulting function FC(X) is defined by

FC(X) = FA(X) O FB(X)

O means any function OPER, which address had been passed to the subroutine COMB. This function must be defined by the statement

REAL FUNCTION oper*4(YA,YB,CON)

where oper is to be replaced by a name. YA and YB mean the input values. CON is an input array of the lenght 20, which can be used by the function. Examples for such functions can be found in 2.3.5.68 to 2.3.5.70.
The subroutine has been taken from /3/ part R with the changes mentioned in 2.3.5.51.

Definition:

SUBROUTINE COMB(OPER,CON,MA,MB,XL,XH,EPS,LOF,LNT)

ENTRY IICOMB(NBT,JNT,X,Y) (This entry passes the addresses of the arrays NBT, JNT, X and Y to the subroutine. It must be called before the first COMB-call and if the addresses are destroyed in the subroutine by using an overlay-structure.)

Explanation of the arguments:

name	mode	type	dimension	remarks
OPER	-	F4	scalar	name of a function, which combines the two TAB1 records (see problem solved)
CON	I	R4	(20)	array passed to the function OPER
MA	I	I4	scalar	identification of first TAB1-record in dense storage (see 2.3.5.80 to 2.3.5.85)
MB	I	I4	scalar	identification of second TAB1-record in dense storage (see 2.3.5.80 to 2.3.5.85)
XL	I	R4	scalar	lower limit of energy array X
XH	I	R4	scalar	upper limit of energy array X
EPS	I	R4	scalar	relative error criterion for generation of the energy grid of the combined TAB1 function
LOF	O	I4	scalar	overflow indicator. Normally 0. If 1, one or some of the arrays NBT, JNT, X or Y are too small
LNT	O	I4	scalar	normal exit = 0 = 1 record MA or MB not in dense storage
NBT	O	I4	(N1X)	produced interpolation ranges
JNT	O	I4	(N1X)	produced interpolation codes
X	O	R4	(N2X)	produced energy grid
Y	O	R4	(N2X)	produced Y (e.g. cross section) values

Used common blocks:

name of the common block: RECS									
name	dimension	type	mode	word	name	dimension	type	mode	word
NR1	(7)	I4	-	1	N2	scalar	I4	0	9
N1	scalar	I4	0	8					

name of the common block: INOUT									
name	dimension	type	mode	word	name	dimension	type	mode	word
NR2	(141)	I4	-	1	SCREPS	scalar	R4	I	142

Called subprograms:

COMBP
CROP
ERROR
FPDS
IPDS
LRIDS
OPER (The function name is defined at the subroutine call statement)

2.3.5.67 Subroutine COMBP

Problem solved:

Two functions, given in the interval $\lceil x_1, x_2 \rceil$ by values at the beginning and end point and by interpolation codes are combined by using the function OPER.
Sufficient points in the interval $\lceil x_1, x_2 \rceil$ are created in order to guarantee the interpolation of the produced function accurate to EPS.
The produced function is added to the function given in the arrays NBT, JNT, X and Y.
The subroutine has been taken from /3/ page R-85 with the following changes:

(1) NBT, JNT, X and Y are now passed to the subroutine by an entry-call instead of a common. The length of these arrays now depends only from the available main storage region.

(2) If $|x_1-x_2| \;/\; |x_1| \leq 10^{-6}$ no intermediate values are generated. The interpolation scheme of the combined function will be linear-linear.

(3) In the case of (2) or a discontinuity a jump to RETURN after adding the values to NBT, JNT, X and Y has been built in.

(4) The interpolation scheme of the produced function can be choosen by the program or can be always linear-linear.

(5) To avoid frequent switching of the interpolation code, following method is used:
Let EPS4 be the interpolation code of a previous interval and EPS5 the best interpolation code for the processed interval. If EPS4 $\leq$ FAICOD•EPS5 the code of the previous interval EPS4 is used. Otherwise EPS5 is used. FAICOD is passed to the program by a common and can be set in the input.

(6) The test on the elimination of points, which can be predicted by interpolation, has been replaced by a CROP-call in the

subroutine COMB (see 2.3.5.66 and 2.3.5.75). The old method had the incorrect effect to smooth faintly varying functions.

Definition:

```
SUBROUTINE COMBP(X1P,YA1P,YB1P,X2P,YA2P,YB2P,IAP,IBP,OPER,CON,
                 EPSP,LOF,NBT,JNT,X,Y)
```

Explanation of the arguments:

name	mode	type	dimension	remarks
X1P	I	R4	scalar	lower limit of the interval
YA1P	I	R4	scalar	value of function A at X1P
YB1P	I	R4	scalar	value of function B at point X1P
X2P	I	R4	scalar	upper limit of the interval
YA2P	I	R4	scalar	value of the function A at X2P
YB2P	I	R4	scalar	value of the function B at X2P
IAP	I	I4	scalar	interpolation code for the function A
IBP	I	I4	scalar	interpolation code for the function B
OPER	-	F4	scalar	name of the function, which combines the functions A and B
CON	I	R4	(20)	array passed to the function OPER
EPSP	I	R4	scalar	relative error criterion for the generation of the energy grid of the combined function
LOF	O	I4	scalar	overflow indicator. Normally 0. If 1, one or some of the arrays NBT, JNT, X or Y have been too small
NBT	O	I4	(N1X)	interpolation ranges
JNT	O	I4	(N1X)	interpolation codes
X	O	R4	(N2X)	energy grid
Y	O	R4	(N2X)	Y (e.g. cross section) values

Used common blocks:

name of the common block: RECS									
name	dimension	type	mode	word	name	dimension	type	mode	word
MAT	scalar	I4	-	1	L1	scalar	I4	-	6
MF	"	I4	-	2	L2	"	I4	-	7
MT	"	I4	-	3	N1	"	I4	IO	8
C1	"	R4	-	4	N2	"	I4	IO	9
C2	"	R4	-	5					

name of the common block: INOUT									
name	dimension	type	mode	word	name	dimension	type	mode	word
NR1	(124)	I4	-	1	N2X	scalar	I4	I	128
ICOMB	scalar	I4	I	125	NR2	(22)	I4	-	129
NBX	"	I4	-	126	FAICOD	scalar	R4	I	151
N1X	"	I4	I	127					

Called subprograms:

OPER (This name is defined at the subroutine call statement.)

TERP1

2.3.5.68 Function ADD

Problem solved:

The quantities YA and YB are added and the result is assigned to the function name (ADD=YA+YB).

Definition:

FUNCTION ADD(YA,YB,CON)

Explanation of the arguments:

name	mode	type	dimension	remarks
YA	I	R4	scalar	first operand
YB	I	R4	"	second operand
CON	-	R4	(20)	not used in ADD

Used common blocks:

None

Called subprograms:

None

2.3.5.69 Function SUB

Problem solved:

The quantities YA and YB are subtracted and the result is assigned to the function name (SUB=YA-YB).

Definition:

FUNCTION SUB(YA,YB,CON)

Explanation of the arguments:

name	mode	type	dimension	remarks
YA	I	R4	scalar	first operand
YB	I	R4	"	second operand
CON	-	R4	(20)	not used in SUB

Used common blocks:

None

Called subprograms:

None

2.3.5.70 Function MULT

Problem solved:

The quantities YA and YB are multiplicated and the result is assigned to the function name (MULT=YA*YB).

Definition:

REAL FUNCTION MULT(YA,YB,CON)

Explanation of the arguments:

name	mode	type	dimension	remarks
YA	I	R4	scalar	first operand
YB	I	R4	"	second operand
CON	-	R4	(20)	not used in MULT

Used common blocks:

None

Called subprograms:

None

2.3.5.71 Subroutine ITAB1

Problem solved:

Given a TAB1 function by the arrays NBT, JNT, X and Y (see /3/ chapter R-2.3.3). The subroutine interpolates the value YP at the x-value XP. If XP is outside the range of the TAB1 function, YP is set to zero.

Definition:

SUBROUTINE ITAB1(XP,YP)

ENTRY EITAB1(NBT,JNT,X,Y)
(An entry call must be made before the first ITAB1-call. By this entry call the addresses of NBT, JNT, X and Y are passed to the subroutine.)

Explanation of the arguments:

name	mode	type	dimension	remarks
XP	I	R4	scalar	x-value
YP	O	R4	scalar	desired y-value for x=XP
NBT	I	I4	(N1X)	interpolation ranges of the given TAB1-record
JNT	I	I4	(N1X)	interpolation codes of the given TAB1-record
X	I	R4	(N2X)	energy grid of the given TAB1-record
Y	I	R4	(N2X)	functional values of the given TAB1-record

Used common block:

name of the common block: RECS									
name	dimension	type	mode	word	name	dimension	type	mode	word
MAT	scalar	I4	-	1	L1	scalar	I4	-	6
MF	"	I4	-	2	L2	"	I4	-	7
MT	"	I4	-	3	N1	"	I4	-	8
C1	"	R4	-	4	N2	"	I4	I	9
C2	"	R4	-	5					

Called subprograms:

ERROR

TERP1

2.3.5.72 Subroutine TERP1

Problem solved:

Interpolate one point. An interpolated value Y at the desired X is computed by using the end points of the line (X1,Y1) and (X2,Y2) and the interpolation code I.

If X1 and/or X2 are negative or zero and interpolation code I=3 or 5 is desired, it is changed to I=2 or 4. If Y1 and/or Y2 are negative or zero and interpolation code I=4 or 5 is desired, it is changed to I=2 or 3. For negative or zero X and I=3 or 5 an error stop occurs.

The program given in /3/ pages R-30 resp. R-59 has been used after doing some changes. One of these changes is given in the following:
The interpolation formulaes for I=3 (y linear in log(x)) and
I=5 (log(4) linear in log (x)) are

$$y_p = y_a + \log\left(\frac{x_p}{x_a}\right) \cdot \frac{y_b - y_a}{\log\left(\frac{x_b}{x_a}\right)} \qquad (I=3)$$

$$y_p = y_a \cdot \mathrm{EXP}\left(\log\left(\frac{x_p}{x_a}\right) \cdot \frac{\log\left(\frac{y_b}{y_a}\right)}{\log\left(\frac{x_b}{x_a}\right)}\right) \qquad (I=5)$$

To avoid numerical difficulties, $\log\left(\frac{x_p}{x_a}\right)$ and $\log\left(\frac{x_b}{x_a}\right)$ is expanded in a series, if the quotients
$\frac{x_p}{x_a}$ and $\frac{x_b}{x_a}$ have values near by one.

Definition:

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

Explanation of the arguments:

name	mode	type	dimension	remarks
X1	I	R4	scalar	lower end of the interpolation interval
Y1	I	R4	scalar	functional value at the lower end of the interpolation interval
X2	I	R4	scalar	upper end of the interpolation interval
Y2	I	R4	scalar	functional value at the upper end of the interpolation interval
X	I	R4	scalar	point for which Y is to be calculated
Y	O	R4	scalar	interpolated value
I	I	I4	scalar	interpolation code in the interval $\boxed{X1,X2}$. May be 1,2,3,4,5

Used common blocks:

None

Called subprogram:

ERROR

2.3.5.73 Subroutine CHGINT

Problem solved:

The interpolation code of the function given in the TAB1-record (see /3/ chapter R-2.3.3)is changed to linear-linear. Discontinuities are eliminated by small changes in the x (energy)-values.

Definition:

SUBROUTINE CHGINT(EPS,N1,N2,NBT,JNT,X,Y,N1X,N2X,*)

Explanation of the arguments:

name	mode	type	dimension	remarks
EPS	I	R4	scalar	passed to subroutine DANI (see 2.3.5.74)
N1	IO	I4	scalar	number of interpolation ranges (array NBT) and interpolation codes (array JNT)
N2	IO	I4	scalar	number of energy points (array X) and functional values (array Y)
NBT	IO	I4	(N1X)	interpolation ranges
JNT	IO	I4	(N1X)	interpolation codes
X	IO	R4	(N2X)	energy grid
Y	IO	R4	(N2X)	functional values (e.g. cross sections)
N1X	I	I4	scalar	maximal dimension of the arrays NBT and JNT
N2X	I	I4	scalar	maximal dimension of the arrays X and Y
*	-	-	-	Return 1 is made, if the dimension of the arrays X and Y is too small

Used common block:

name of the common block: RE									
name	dimension	type	mode	word	name	dimension	type	mode	word
XR	(300)	R4	I	1	YR	(300)	R4	I	301

Called subprograms:

DANI
ERROR

2.3.5.74 Subroutine DANI

Problem solved:

For a given interval [X1,X2] with functional values Y1 resp. Y2 and any permitted interpolation code JNTX, an energy grid from X1 to X2 is generated and the associate functional values are calculated. The grid is made by bisections of the interval [X1,X2] and the produced subintervals.

Bisectioning is stopped, if the functional value at a point $x_h = 0.5 \cdot (x_n + x_m)$ - x_n and x_m are neighboured points - can be predicted by linear interpolation with an error less EPS. The interval from X1 to X2 can have maximally 300 points, including X1 and X2.

Definition:

SUBROUTINE DANI(X1,Y1,X2,Y2,JNTX,N,EPS)

Explanation of the arguments:

name	mode	type	dimension	remarks
X1	I	R4	scalar	lower limit of the interval
Y1	I	R4	scalar	functional value at the lower end of the interval
X2	I	R4	scalar	upper limit of the interval
Y2	I	R4	scalar	functional value at the upper end of the interval
JNTX	I	I4	scalar	predefined interpolation code
N	O	I4	scalar	number of produced energy points (and functional values) in the array XR (resp. YR). (XR and YR see common RE) ($2 \leq N \leq 300$)
EPS	I	R4	scalar	accuracy (see problem solved)

Used common block:

name of the common block: RE									
name	dimension	type	mode	word	name	dimension	type	mode	word
XR	(300)	R4	0	1	YR	(300)	R4	0	301

Called subprogram:

ERROR

2.3.5.75 Subroutine CROP

Problem solved:

A TAB1-record (see /3/ chapter R-2.3.3) stored in the arrays NBT, JNT, X and Y is cropped, and only the portion between XL and XH is retained. Portions of the range not defined by the original TAB1 record are set to zero. Unnecessary points are eliminated, if they can be predicted by interpolation between adjacent points to relative accuracy of EPS.

The subroutine originates from /3/ (see /3/ chapter R-6.4 and page R-75). Yet following changes have been made before using it:

(1) EPS is equal EPSMIN in the case of LOSSTE equal .FALSE. In this case the variables EPSMAX, SCRCLA and NUMSCR have no meaning. If LOSSTE is equal .TRUE., EPS has also normally the value EPSMIN. It has the value EPSMAX, if two adjacent points fulfil the condition $\left|\frac{X2-X1}{X1}\right| < 5.0 \cdot 10^{-6}$.

In this case the number of points which have been eliminated, will be counted. To do this, first so called scratch classes are built in the following way. If EPSMIN is less than EPSMAX, five intervals are made from 0.0 up to EPSMIN, and another five from EPSMIN to EPSMAX. If EPSMIN is equal EPSMAX ten intervals are made from zero to EPSMIN. In the subroutine CROP the accuracy, by which the cropped points could have been predicted, is calculated, and the number of these points is added up with respect to the associated scratch class. The array element NUMSCR(I) contains the number of cropped points that could have been predicted with an relative error in the range

SCRCLA(I-1) < EPS $\leq$ SCRCLA(I)

(SCRCLA(0) is defined to be zero).

(2) Points in an interval are only eliminated, if all points between can be obtained by interpolation with an error less EPS.

Definition:

SUBROUTINE CROP(XL,XH,EPSMIN,LOSSTE,EPSMAX,SCRCLA,NUMSCR,LOF,NBT,JNT,X,Y)

Explanation of the arguments:

name	mode	type	dimension	remarks
XL	I	R4	scalar	lower energy limit
XH	I	R4	scalar	upper energy limit
EPSMIN	I	R4	scalar	see "problem solved"
LOSSTE	I	L4	scalar	" " "
EPSMAX	I	I4	scalar	" " "
SCRCLA	O	R4	(10)	" " "
NUMSCR	O	I4	(10)	" " "
LOF	O	I4	scalar	= 0 normal return = 1 the dimension of the arrays X and Y is too small to add zeros at the lower or upper end
NBT	IO	I4	(N1X)	interpolation ranges
JNT	IO	I4	(N1X)	interpolation codes
X	IO	R4	(N2X)	energy grid
Y	IO	R4	(N2X)	functional values

Used common blocks:

name of the common block: RECS									
name	dimension	type	mode	word	name	dimension	type	mode	word
MAT	scalar	I4	-	1	L1	scalar	I4	-	6
MF	scalar	I4	-	2	L2	scalar	I4	-	7
MT	scalar	I4	-	3	N1	scalar	I4	IO	8
C1	scalar	R4	-	4	N2	scalar	I4	IO	9
C2	scalar	R4	-	5					

name of the common block: INOUT									
name	dimension	type	mode	word	name	dimension	type	mode	word
NR1	(125)	I4	-	1	N1X	scalar	I4	I	127
NBX	scalar	I4	-	126	N2X	scalar	I4	I	128

Called subprograms:

ERROR

TERP1

2.3.5.76 Subroutine WDA

Problem solved:

The contents of the array BK from I=1 to I=N (N given in common INOUT) are written on the direct access data set with the logical unit number NF=11. All written records have the lenght 400 4-Byte words. The data set has the following structure:

1. record (table of contents and addresses)

word number	contents	remarks
1	MATKDK	numerical material name
2	MIS	number of the isotope
3	NAME1	name of the (KEDAK) data type in alphamerical representation
4	NAME2	
5	IR	number of the record, where the data can be found
6	N	number of data measured in 4-Byte words
.		
.		
.		

word 1 until 6 are repeated for each entry

399	IR	first free record
400	IW	first free word in record number 1

2. and following records

contents of the array BK.

Definition:
SUBROUTINE WDA
(not used entry)

ENTRY WDA1(BK)
(passes the address of BK to the subroutine and
initializes the data set on unit NF=11)

ENTRY WDA2(MATKDK,MIS,NAME1,NAME2)
(does the writing of the contents of BK on unit
NF=11)

ENTRY WDA3(IT)
(prints the table of contents and addresses (record 1) of the
dataset on unit NF=11 and returns the length of the filled
portion of record 1.)

Explanation of the arguments:

name	mode	type	dimension	remarks
BK	I	R4	(NBX)	data to be written on logical unit number NF=11
MATKDK	I	I4	scalar	see problem solved
MIS	I	I4	scalar	" " "
NAME1	I	I4	scalar	" " "
NAME2	I	I4	scalar	" " "
IT	O	I4	scalar	lenght of the filled portion in record 1 (table of contents and addresses)

Used common block:

name of the common block: INOUT									
name	dimension	type	mode	word	name	dimension	type	mode	word
NO	scalar	I4	I	1	N	scalar	I4	I	129
NR1	(127)	I4	-	2	NBKX	scalar	I4	I	130

Called subprogram :
ERROR

2.3.5.77 Subroutine RDAB

Problem solved:
The data written by the subroutine WDA (see 2.3.5.76) on the direct access data set with the logical unit number NF=11 are read into the array BK. Organisation of the dataset see 2.3.5.76.

Definition:
SUBROUTINE RDAB
(This entry is never called.)

ENTRY RDA1(BK)
(This entry must be called before using all other entries. It passes the address of the array BK to the subroutine and reads the first record - table of contents and addresses - of the dataset.)

ENTRY RDA4(MATKDK,MIS,NAME1,NAME2,LNT,IR)
(The datatype "NAME1,NAME2" of the isotope number MIS of the material MATKDK is searched in the table of contents and addresses. If found the record number is stored in IR, the number of data in the quantity N of the common INOUT and RETURN is made.)

ENTRY RDA2(MATKDK,MIS,NAME1,NAME2,LNT)
(The datatype "NAME1,NAME2" of the isotope number MIS of the material MATKDK is searched in the table of contents and addresses. If found the data are read into the array BK.)

ENTRY RDA3(MATKDK)
(The number of datatypes, which are descenting from ENDF/B MF=1,2, and 3, and are to be inserted in KEDAK, are counted in the variable NTYP of the common INOUT.)

Explanation of the arguments:

name	mode	type	dimension	remarks
BK	O	R4	(NBKX)	used to store data read from the direct access data set NF=11
MATKDK	I	I4	scalar	numerical KEDAK material name
MIS	I	I4	scalar	number of the isotope
NAME1	I	I4	scalar	first part of the KEDAK alphamerical data type name
NAME2	I	I4	scalar	second part of the KEDAK alphamerical data type name
LNT	O	I4	scalar	0 normal RETURN. Data found. 1 data not found
IR	O	I4	scalar	number of the record of the direct access data set NF=11, where the desired data starts

Used common blocks:

name of the common block: TAB									
name	dimension	type	mode	word	name	dimension	type	mode	word
ENDTAB	(25)	I4	-	1	NCNAME	(39)	I4	-	316
KEDTAB	(50)	I4	-	26	NRIF3	scalar	I4	-	355
KEDR	(240)	I4	I	76	NRIF4	scalar	I4	-	356

name of the common block: INOUT									
NR1	(83)	I4	-	1	NR2	(44)	I4	-	85
NTYP	scalar	I4	0	84	N	scalar	I4	0	129

Called subprograms:

None

2.3.5.78 Subroutine WRECS

Problem solved:

TAB1 and LIST records (see /3/ chapter R-2.3.3 and chapter R-2.3.2) are stored on the direct access data set NF=12. One record has the lenght of 400 4-Byte words.

Organisation of the data set:

Record 1 until record BIB12 (table of contents and addresses)

word number	contents	remarks
1	MIS	number of the isotope for a specified material
2	NAME1 }	name of a datatype in alpha-
3	NAME2 }	merical form
4	EL for TAB1 records	lower energy limit
	0 for LIST records	
5	EH for TAB1 records	upper energy limit
	0 for LIST records	
6	IR	number of the record, where the data starts
7	N1	lenght of the arrays NBT and JNT for TAB1-records lenght of the array B for LIST-records
8	N2 for TAB1 records	lenght of the arrays X and Y
	0 for LIST records	

words number 1 until 8 are repeated for each entry.

Records BIB12+1, BIB12+2,... contain the data.

Definition:

SUBROUTINE WRECS

(This entry is never used)

ENTRY WRECS1(NBT,JNT,X,Y,B)

(This entry must be called first before using any other entry. The addresses of NBT,JNT,X,Y and B are passed to the subroutine and the dataset is initialized.)

ENTRY WRECS2(MIS,NAME1,NAME2,EL,EH)

(TAB1-records are written on the dataset.)

ENTRY WRECS4(MIS,NAME1,NAME2)

(LIST-records are written on the dataset.)

Explanation of the arguments:

name	mode	type	dimension	remarks
NBT	I	I4	(N1X)	interpolation ranges
JNT	I	I4	(N1X)	interpolation codes
X	I	R4	(N2X)	energy grid
Y	I	R4	(N2X)	functional values
B	I	R4	(NBX)	"LIST-records"
MIS	I	I4	scalar	number of the isotope
NAME1	I	I4	scalar	first part of the alphamerical data type name
NAME2	I	I4	scalar	second part of the alphamerical data-type name
EL	I	R4	scalar	lower energy limit of the TAB1-record
EH	I	R4	scalar	upper energy limit of the TAB1-record

Used common blocks:

name of the common block: INOUT									
name	dimension	type	mode	word	name	dimension	type	mode	word
NO	scalar	I4	I	1	BIB12	scalar	I4	I	79
NR1	(77)	I4	-	2					

name of the common block: RECS									
MAT	scalar	I4	-	1	L1	scalar	I4	-	6
MF	scalar	I4	-	2	L2	scalar	I4	-	7
MT	scalar	I4	-	3	N1	scalar	I4	I	8
C1	scalar	R4	-	4	N2	scalar	I4	I	9
C2	scalar	R4	-	5					

Called subprograms:

None

2.3.5.79 Subroutine RRECS

Problem solved:

The data written by the subroutine WRECS (see 2.3.5.78) on the direct access data set NF=12 are read into the arrays NBT,JNT,X, and Y for TAB1-records respectively B for LIST-records. Organisation of the dataset see 2.3.5.78.

Definition:

SUBROUTINE RRECS(MIS,NAME1,NAME2,EL,EH,LNT)

(Reads TAB1-records)

ENTRY RRECS4(MIS,NAME1,NAME2,LNT)

(Reads LIST-records)

ENTRY IRRECS(NBT,JNT,X,Y,B)

(This entry must be called first. The addresses of NBT,JNT,X,Y, and B are passed to the subroutine.)

Explanation of the arguments:

name	mode	type	dimension	remarks
MIS	I	I4	scalar	number of the isotope for a specified material
NAME1	I	I4	scalar	first part of the desired datatype name in alphamerical form
NAME2	I	I4	scalar	second part of the desired datatype name in alphamerical form
EL	O	R4	scalar	lower energy limit for the TAB1-record
EH	O	R4	scalar	upper energy limit for the TAB1-record
LNT	O	I4	scalar	0 data found 1 data not found
NBT	O	I4	(N1X)	interpolation ranges
JNT	O	I4	(N1X)	interpolation codes
X	O	R4	(N2X)	energy grids
Y	O	R4	(N2X)	functional values
B	O	R4	(NBX)	"LIST-records"

Used common blocks:

name of the common block: INOUT									
name	dimension	type	mode	word	name	dimension	type	mode	word
NR1	(78)	I4	-	1	NBX	scalar	I4	I	126
BIB12	scalar	I4	I	79	N1X	scalar	I4	I	127
NR2	(46)	I4	-	80	N2X	scalar	I4	I	128

name of the common block: RECS									
name	dimension	type	mode	word	name	dimension	type	mode	word
MAT	scalar	I4	-	1	L1	scalar	I4	-	6
MF	scalar	I4	-	2	L2	scalar	I4	-	7
MT	scalar	I4	-	3	N1	scalar	I4	0	8
C1	scalar	R4	-	4	N2	scalar	I4	0	9
C2	scalar	R4	-	5					

Called subprogram:
ERROR

2.3.5.80 Subroutine STORE

Problem solved:
The subroutines described in 2.3.5.80 until 2.3.5.85 work with the so called "Dense Record Storage" (see /3/ chapter R-3.2). This is a fast core storage to store all possible ENDF/B record types in a dense manner. To work with this storage the subroutines given in /3/ pages R-37, R-39, and R-40 and pages R-71, R-72, R-74, R-77, and R-78 have been used. Some arrays, located in a common in the original routines have yet been replaced by arrays, which are now passed to the subroutines by call-statements.

The subroutine STORE stores one record, which is given in the arrays NBT, JNT,X,Y and B, in the dense storage. The organisation of the dense storage is as following.

A table of contents and addresses is hold in the common DENS (see commons used in the program 2.3.4). The meaning of these quantities is:

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

JAT(M) starting location in the array A of the M^{th} record.

JTT(M) type of the M^{th} record (see /3/ chapter R-2.3)

JLT(M) number of words in array A used by the M^{th}-record

JNS next available location in the array A.

MNS next available location in the arrays JMT,JAT,JTT and JLT.

MX lenght of the arrays JMT,JAT,JTT and JLT (= 100)

The dense storage itself is realized in the array A (see 2.3.3). The organisation of this array can be found in /3/ chapter R-3.2.

Definition:
SUBROUTINE STORE(JT,MA,LOF)

ENTRY ISTORE(NBT,JNT,X,Y,B,A,LA)
(Passes the addresses of NBT,JNT,X,Y,B,A and LA to the subroutine.
Must be called first.)

Explanation of the arguments:

name	mode	type	dimension	remarks
JT	I	I4	scalar	record type. May be 1,2,3,4,5,6
MA	I	I4	scalar	identification of the record to be stored. If a record with the same identification is already stored, it is deleted
LOF	O	I4	scalar	0 normal RETURN 1 dimension of array A too small
NBT	I	I4	(N1X)	interpolation ranges
JNT	I	I4	(N1X)	interpolation codes
X	I	R4	(N2X)	energy grid
Y	I	R4	(N2X)	functional values
B	I	R4	(NBX)	"LIST-records"
A	O	R4	(JX)	dense storage
LA	O	I4	(JX)	dense storage. Same starting address as array A

Used common blocks:

name of the common block: DENS									
name	dimension	type	mode	word	name	dimension	type	mode	word
JMT	(100)	I4	IO	1	JNS	scalar	I4	IO	401
JAT	(100)	I4	0	101	MNS	scalar	I4	IO	402
JTT	(100)	I4	0	201	MX	scalar	I4	I	403
JLT	(100)	I4	0	301					

name of the common block: INOUT									
NR1	(130)	I4	-	1	JX	scalar	I4	I	131

name of the common block: RECS									
MAT	scalar	I4	I	1	L1	scalar	I4	I	6
MF	scalar	I4	I	2	L2	scalar	I4	I	7
MT	scalar	I4	I	3	N1	scalar	I4	I	8
C1	scalar	R4	I	4	N2	scalar	I4	I	9
C2	scalar	R4	I	5					

Called subprograms:

DELETE

ERROR

2.3.5.81 Subroutine FETCH

Problem solved:
One record from dense storage is fetched (see 2.3.5.80 for definitions and sources).

Definition:
SUBROUTINE FETCH(MA,LNT)
ENTRY IFETCH(NBT,JNT,X,Y,B,A,LA)
(Passes the addresses of the arrays NBT,JNT,X,Y,B,A and LA to the subroutine. Must be called first.)

Explanation of the arguments:

name	mode	type	dimension	remarks
MA	I	I4	scalar	identification of the record to be fetched
LNT	O	I4	scalar	0 record has been found and moved to the common RECS and some of the arrays NBT,JNT,X,Y,B 1 record not found
NBT	O	I4	(N1X)	interpolation ranges
JNT	O	I4	(N1X)	interpolation codes
X	O	R4	(N2X)	energy grid
Y	O	R4	(N2X)	functional values
B	O	R4	(NBX)	"LIST-record"
A	I	R4	(JX)	dense storage
LA	I	I4	(JX)	dense storage (Same starting address as array A.)

Used common blocks:

name of the common block: RECS									
name	dimension	type	mode	word	name	dimension	type	mode	word
MAT	scalar	I4	0	1	L1	scalar	I4	0	6
MF	scalar	I4	0	2	L2	scalar	I4	0	7
MT	scalar	I4	0	3	N1	scalar	I4	0	8
C1	scalar	R4	0	4	N2	scalar	I4	0	9
C2	scalar	R4	0	5					

name of the common block: DENS									
JMT	(100)	I4	I	1	JNS	scalar	I4	-	401
JAT	(100)	I4	I	101	MNS	scalar	I4	I	402
JTT	(100)	I4	I	201	MX	scalar	I4	-	403
JLT	(100)	I4	-	301					

name of the common block: INOUT									
NR1	(125)	I4	-	1	N1X	scalar	I4	I	127
NBX	scalar	I4	I	126	N2X	scalar	I4	I	128

Called subprogram:

ERROR

2.3.5.82 Subroutine DELETE

Problem solved:

Deletes one record from dense storage or clears total dense storage.

(See 2.3.5.80 for definitions and sources.)

Definition:

SUBROUTINE DELETE(MA,A)

Explanation of the arguments:

name	mode	type	dimension	remarks
MA	I	I4	scalar	identification of the record to be deleted. If MA=0 all records in dense storage are deleted
A	IO	R4	(JX)	dense storage

Used common blocks:

name of the common block: DENS									
name	dimension	type	mode	word	name	dimension	type	mode	word
JMT	(100)	I4	IO	1	JNS	scalar	I4	IO	401
JAT	(100)	I4	IO	101	MNS	scalar	I4	IO	402
JTT	(100)	I4	IO	201	MX	scalar	I4	I	403
JLT	(100)	I4	IO	301					

name of the common block: INOUT									
NR1	(130)	I4	-	1	JX	scalar	I4	I	131

Called subprograms:

None

2.3.5.83 Subroutine FPDS

Problem solved:

One point from a TAB1-record stored in dense storage is fetched. The data are stored in XP,YP and IP. The following output is produced:

NP	XP	YP	IP
≤ 0	$-1.0 \cdot 10^{20}$	0.0	1
> 0 and $< N2$	X(NP)	Y(NP)	interpolation code stored in dense storage
$= N2$	X(N2)	Y(N2)	2
$= N2+1$	X(N2)	0.0	2
$> N2+1$	$1.0 \cdot 10^{20}$	0.0	2

(TAB1-record see /3/ chapter R-2.3.3. For definition of dense storage and sources for the program see 2.3.5.80)

Definition:

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

ENTRY IFPDS(A,LA)

(Must be called first before any FPDS-call. Passes the address of A resp. LA to the subroutine.)

Explanation of the arguments:

name	mode	type	dimension	remarks
JA	I	I4	scalar	starting index in dense storage (realized by the array A resp. LA) of the TAB1-record, which is constructed out of the arrays NBT,JNT,X and Y
NP	I	I4	scalar	desired point number
XP	O	R4	scalar	= X(NP),X(n2),$-1.0 \cdot 10^{20}$, or $1.0 \cdot 10^{20}$ depending from the value of NP (see problem solved)
YP	O	R4	scalar	= Y(NP),Y(N2),or 0.0 depending from the value of NP (see problem solved)

name	mode	type	dimension	remarks
IP	O	I4	scalar	interpolation code used between X(NP) and X(NP+1), or 1 or 2 depending from the value of NP (see problem solved).
A	I	R4	(JX)	dense storage array
LA	I	I4	(JX)	dense storage array (Same starting address as array A.)

Used common blocks:

None

Called subprograms:

None

2.3.5.84 Subroutine IPDS

Problem solved:

Interpolates one point in a TAB1-record in the dense storage.

If the given value of X, which is named XP, is less than X(1), the corresponding (desired) value of Y named YP is set to zero, NP (address of X and Y arrays, such that XP lies between X(NP) and X(NP+1)) is also set to zero, and the interpolation code IP to 1. If XP is greater than X(N2), YP is set to zero, NP to N2+1, and IP to 1. (TAB1-record see /3/ chapter R-2.3.3. For definition of dense storage and sources for the program see 2.3.5.80.)

Definition:

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

ENTRY IIPDS(A,LA)

(Must be called first before any IPDS-call. Passes the address of A resp. LA to the subroutine.)

Explanation of the arguments:

name	mode	type	dimension	remarks
JA	I	I4	scalar	starting index in dense storage, which is realized by the array A resp. LA, of the TAB1-record
NP	IO	I4	scalar	index of X and Y arrays, such that XP lies between X(NP) and X(NP+1). If NP is greater than zero on input, it is used to start search. Otherwise search is started at NP=1
XP	I	R4	scalar	given value of X
YP	O	R4	scalar	corresponding (desired) value of Y
IP	O	I4	scalar	interpolation code used to compute YP, or 1 if XP is outside the TAB1-record range
A	I	R4	(JX)	dense storage array
LA	I	I4	(JX)	dense storage array. (Same starting address as array A.)

Used common blocks:

None

Called subprograms:

ERROR

TERP1

2.3.5.85 Subroutine LRIDS

Problem solved:

Locates record MA in the dense storage. (For definition of dense storage and sources for the program see 2.3.5.80.)

Definition:

SUBROUTINE LRIDS(MA,JA,LNT)

Explanation of the arguments:

name	mode	type	dimension	remarks
MA	I	I4	scalar	identification of the record
JA	O	I4	scalar	starting index in array A for the record identified by MA
LNT	O	I4	scalar	= 0 address JA of the record identified by MA found = 1 record identified by MA not in dense storage

Used common block:

name of the common block: DENS									
name	dimension	type	mode	word	name	dimension	type	mode	word
JMT	(100)	I4	I	1	JNS	scalar	I4	-	401
JAT	(100)	I4	I	101	MNS	scalar	I4	I	402
JTT	(100)	I4	-	201	MX	scalar	I4	-	403
JLT	(100)	I4	-	301					

Called subprograms:

None

2.3.5.86 Subroutine SICKY

Problem solved:

Inserts the data in a KEDAK library[+] on the direct access data set IFILE=1. If no library exists, one is built up. If an isotope is yet present in the library and IMOD (see 2.2.6) has been set to 1, the isotope is replaced.

[+]This KEDAK library must not have been processed with or created by KEMA (see II or /4/). It must have been created by BRIGITTE itself.

If IMOD has the value zero, and an isotope is yet present in the library, the data are not inserted.

Deleting a specified isotope and compressing the library is done by the subroutine LISA (see 2.3.5.87). The data to be inserted in the KEDAK library are read from direct access dataset NF=11 by the subroutine RDAB.

Definition:
SUBROUTINE SICKY(NIS,NISO,IF4,IF5,BK,TAB,ITAB)

Explanation of the arguments:

name	mode	type	dimension	remarks
NIS	IO	I4	scalar	number of isotopes. Is set to 1 for NISO equal zero
NISO	I	I4	scalar	number of isotopes. Only used to set NIS to 1 for NISO equal zero
IF4	I	I4	scalar	= 1 only data from ENDF/B MF=4 have been converted ≠ 1 data from all files can be found in direct access data set NF=11
IF5	I	I4	scalar	= 1 only data from ENDF/B MF=5 have been converted ≠ 1 data from all files can be found in direct access data set NF=11
BK	L	R4	(NBKX)	used to store data read from direct access dataset NF=11
TAB	L	R4	(ITDIM)	used to built up the data to be written in KEDAK
ITAB	L	I4	(ITDIM)	same array as TAB

Used common blocks:

name of the common block: TAB									
name	dimension	type	mode	word	name	dimension	type	mode	word
ENDTAB	(25)	I4	-	1	NCNAME	(39)	I4	I	316
KEDTAB	(50)	I4	-	26	NRIF3	scalar	I4	I	355
KEDR	(240)	I4	I	76	NRIF4	scalar	I4	I	356

name of the common block: INOUT									
name	dimension	type	mode	word	name	dimension	type	mode	word
NRE1	(80)	I4	-	1	MATISO	(10)	I4	I	105
NAMKDK	(2)	I4	IO	81	NRE2	(14)	I4	-	115
MATKDK	scalar	I4	I	83	NKED	scalar	I4	IO	129
NTYP	scalar	I4	IO	84	NRE3	(3)	I4	-	130
NAMISO	(20)	I4	I	85	ITDIM	scalar	I4	I	133

Called subprograms:

DATUM
LISA
PRIZEI
RDA1 }
RDA2 } (entries of RDAB)
RDA3 }
RDA4 }
SPLIT
TABPRT

2.3.5.87 Subroutine LISA

Problem solved:
The material "INAM1,INAM2" in the KEDAK library on the direct access data set on NT=1 is deleted. The library is compressed[+].

Definition:
SUBROUTINE LISA(INAM1,INAM2,IFIRST)

[+]This subroutine can not be used for KEDAK libraries, which have been created by or processed with KEMA (see II), because the libraries generated by BRIGITTE have a special form. First in word 880 of the first record the number of the next available record can be found. Second each material starts on a new record.

Explanation of the arguments:

name	mode	type	dimension	remarks
INAM1	I	I4	scalar	first part of the alphanumerical KEDAK material name to be deleted
INAM2	I	I4	scalar	second part of the alphanumerical KEDAK material name to be deleted
IFIRST	O	I4	scalar	0 normal RETURN 1 the deleted material is the only one in the library. The library is regarded as empty and must be initialized in the calling program

Used common blocks:

None

Called subprograms:

None

2.3.5.88 Subroutine TABPRT

Problem solved:

A table of contents of the KEDAK library, which contains the inserted material, is printed out. This routine can also be used for libraries, which have been created by KEMA (see II).

Definition:

SUBROUTINE TABPRT(MAT,NMAT,IFIL)

Explanation of the arguments:

name	mode	type	dimension	remarks
MAT	IO	R8	(NMAT)	MAT(1) = 'ALL':the material conversion table, the type conversion table, the material address table and the type address tables of all materials are printed out = 'material names': like MAT(1)= 'ALL' with the exception that only the type address tables of the specified materials are printed out
NMAT	I	I4	scalar	number of material names in the array MAT
IFIL	I	I4	scalar	logical unit number of the KEDAK library, for which the table of contents is to be printed out

Used common blocks:

None

Called subprograms:

INITPG (entry of PAGE)

PAGE

PRIZEI

2.3.5.89 Subroutine PAGE

Problem solved:

Called by TABPRT (see 2.3.5.88). Prints the heading of each page created by TABPRT.

Definition:

SUBROUTINE PAGE(TITLE,IL)

ENTRY INITPG(IFIL)

(Must be called first. The page counter is set to zero and the value of IFIL is passed to the subroutine.)

Explanation of the arguments:

name	mode	type	dimension	remarks
TITLE	I	R8	scalar	text printed in the heading of each page (e.g. a material name)
IL	O	I4	scalar	number of lines per page printed in the subroutine
IFIL	I	I4	scalar	logical file number of the KEDAK dataset

Used common blocks:

None

Called subprograms:

None

2.3.5.90 Subroutine SPLIT

Problem solved:

The data given in the form dd.mm.jj in EBCDIC are splitted up in three integer values dd, mm, and jj.

Definition:

SUBROUTINE SPLIT(A,I1,I2,I3)

Explanation of the arguments:

name	mode	type	dimension	remarks
A	I	R8	scalar	date in the form dd.mm.jj
I1	O	I4	scalar	dd
I2	O	I4	scalar	mm
I3	O	I4	scalar	jj

Used common blocks:

None

Called subprogram:

CON

2.3.5.91 Subroutine CON

Problem solved:
A number in the range 0 until 99 represented in EBCDIC, is converted in a machine internal representation.

Definition:
SUBROUTINE CON(IA,IN)

Explanation of the arguments:

name	mode	type	dimension	remarks
IA	I	I2	scalar	number in EBCDIC
IN	O	I2	scalar	number in internal representation

Used common blocks:
None

Called subprogram:
CONVX

2.3.5.92 Subroutine A8FORM

Auxiliary program (member of a local library) used by BRIGITTE (see /13/).

Problem solved:
Prints texts with "big" letters. One "line" can contain up to 8 characters. One page can contain 3 lines.

Because the subroutine is not a part of the program BRIGITTE, only examples how to use the program together with an explanation of the arguments will be given in the following.

Program call:

CALL A8FORM(IA,T)

Effect of the call:

The text T is written on unit IA.

Explanation of the arguments:

name	mode	type	dimension	remarks
IA	I	I4	scalar	print output unit number
T	I	R8	scalar	text (8 characters) to be printed

Program call:

CALL RAMANF(IA)

CALL RAMEND(IA)

Effect of the call:

The top and the bottom of a frame of asterisk is produced. All A8FORM-calls between will produce addionally with the text the left and right parts of the frame.

Explanation of the argument:

name	mode	type	dimension	remarks
IA	I	I4	scalar	print output unit number

Examples for the use of the routine in the source program:

See subroutine TITEL (2.3.5.2).

2.3.5.93 Subroutine DEFI

Auxiliary program (member of a local library) used by BRIGITTE (see /14/).

Problem solved:
The static IBM-FORTRAN IV DEFINE FILE statement (see /9/) is made dynamic.

Because the subroutine is not a part of the program BRIGITTE, only an example how to use the program together with an explanation of the arguments will be given in the following.

Program call:
CALL DEFI(UNIT,NPRIQ,FORMAT,NBLK,NASS)+

Effect of the call:
The direct access dataset on unit number UNIT is defined.

Explanation of the arguments:

name	mode	type	dimension	remarks
UNIT	I	I4	scalar	logical unit number
NPRIQ	I	I4	scalar	number of records for the direct access data set (see 2.3.5.93)
FORMAT	I	R4	scalar	format of the records. May be 'L','E' or 'U' (see /9/)
NBLK	I	I4	scalar	number of words for FORMAT='U' or number of bytes for FORMAT='L' or ='E' of one record
NASS	P	I4	scalar	address of the associated variable (see /9/)

Example for the use of the routine in the source program:

See MAIN program (2.3.5.1)

+The argument list can be repeated

2.3.5.94 Subroutine DINF

Auxiliary program (member of a local library) used by BRIGITTE (see /14/).

Problem solved:
The data block lenght and the primary quantity, both given in the SPACE parameter of a DD-card (see /10/), are fetched and passed to the calling program.

Because the subroutine is not a part of the program BRIGITTE, only examples how to use the program together with an explanation of the arguments will be given in the following.

Program call:
CALL DINF(DDNAME,NBLK,NPRIQ)

Effect of the call:
NBLK is filled with the data block lenght and NPRIQ with the primary quantity of the SPACE parameter of the DD-card DDNAME.

Explanation of the arguments:

name	mode	type	dimension	remarks
DDNAME	I	R8	scalar	DD-name of the DD-card in interest
NBLK	O	I4	scalar	data block lenght (in byte) given in the SPACE parameter
NPRIQ	O	I4	scalar	primary quantity of the SPACE parameter. Number of records for the dataset

Examples for the use of the routine in the source program:

See subroutine DADEFI (2.3.5.7)

2.3.5.95 Subroutine FREESP

Auxiliary program (member of a local library) used by BRIGITTE (see /15/).

Problem solved:
The size of the unused part of the region (see /10/) - measured in units of 2 k-bytes - is fetched.

Because the subroutine is not a part of the program BRIGITTE, only an example how to use the program together with an explanation of the argument will be given in the following.

Program call:
CALL FREESP(I)

Effect of the call:
I contains the size of the unused region, measured in units of 2 k-bytes.

Explanation of the argument:

name	mode	type	dimension	remarks
I	O	I4	scalar	unused part of the region measured in units of 2 k-bytes

Example for the use of the routine in the source program:

See subroutine GSPACE (2.3.5.5).

2.3.5.96 Subroutine XTAREA

Auxiliary program (member of a local library) used by BRIGITTE (see /16/).

Problem solved:
Portions of the unused region or all of it are placed at the program disposal on execution time by using the IBM/370 macros GETMAIN and FREEMAIN (see /11/).

Because the subroutine is not a part of the program BRIGITTE, only an example how to use the program together with an explanation of the arguments will be given in the following.

Program call:
CALL XTAREA(K1,K2,K3,AREA)

Effect of the call:
The absolute address of the desired space is filled in K1. K2 is filled with zero, if the desired space was not available. K3 returns the address of the array AREA.

Explanation of the arguments:

name	mode	type	dimension	remarks
K1	O	I4	scalar	absolute address - measured in bytes - of the beginning of the desired space
K2	IO	I4	scalar	on input: number of bytes of the desired space. on output: if greater zero: normal RETURN. If equal zero: space was not available
K3	O	I4	scalar	absolute address of the beginning of the array AREA. Used to calculate the relative address of the desired space in the FORTRAN program
AREA	P	-	(1)	used as reference point (see K3)

Program call:
CALL REXTAR(K1,K2)

Effect of the call:
K2 bytes starting on the absolute address K1 are released

Explanation of the arguments:

name	mode	type	dimension	remarks
K1	I	I4	scalar	absolute address of area to be released
K2	I	I4	scalar	number of bytes to be released

Example for the use of the routine in the source program:

See subroutine GSPACE (2.3.5.5)

2.3.5.97 Function ZEIT

Auxiliary program (member of a local library) used by BRIGITTE (see /17/).

Problem solved:
The used CPU-time - relative to a fixed zero point - is measured.

Because the function is not a part of the program BRIGITTE, only examples how to use the program together with an explanation of the arguments will be given in the following.

Program call:

```
UHRZEI=0.0
ANFZEI=ZEIT(UHRZEI)
```

Effect of the call:
This call defines a reference point. All following calls will give times relative to this call.

Explanation of the argument:

name	mode	type	dimension	remarks
UHRZEI	I(O)	R4	scalar	used to define the reference point UHRZEI must have the value 0.0. It can be changed by this first ZEIT-call

Program call:
DURA=ZEIT(ANFZEI)

Effect of the call:
The CPU-time passed since the ZEIT(UHRZEI)-call is stored in DURA. It is given in seconds as REAL*4 number.

Explanation of the argument:

name	mode	type	dimension	remarks
ANFZEI	I	R4	scalar	value produced at the first ZEIT-call

Examples for the use of the routine in the source program:

See MAIN-program (2.3.5.1) and subroutine PRIZEI (2.3.5.45).

2.3.5.98 Subroutine DATUM

Auxiliary program (member of a local library) used by BRIGITTE (see /18/).

Problem solved:
The date and time of day is returned.

Because the subroutine is not a part of the program BRIGITTE, only an example how to use the program together with an explanation of the arguments will be given in the following.

Program:call:
CALL DATUM(DATE,UHRZEI)

Effect of the call:
The date and time of day is returned.

Explanation of the arguments:

name	mode	type	dimension	remarks
DATE	O	R8	scalar	date in the form dd.mm.yy (day, month, year)
UHRZEI	O	R8	scalar	time of day in the form hh.mm.ss (hour, minute, second)

Example for the use of the routine in the source program:

See subroutine PRIZEI (2.3.5.45)

2.3.5.99 Subroutine CONVX

Auxiliary program (member of a local library) used by BRIGITTE (see /19/).

Problem solved:
Conversion of machine internal numbers in alphanumerical representation (EBCDIC).

Because the subroutine is not a part of the program BRIGITTE, only examples how to use the program together with an explanation of the arguments will be given in the following.

Program call:
CALL CONVX(ARG,ALPH,FIELDD)

Effect of the call:
The machine internal number ARG is stored in alphanumerical representation by using the format FIELDD in the variable ALPH.

Explanation of the arguments:

name	mode	type	dimension	remarks
ARG	I	-	scalar	machine internal number
ALPH	O	-+	-+	alphanumerical representation
FIELDD	I	-+	-+	conversion format

+If no type or dimension is given, several possibilities are permitted.

Example for the use of the routine in the source program:

See subroutine CON (2.3.5.91).

2.4 Literature

/1/ J.C. Schepers
BRIGITTE. A computer program for translating neutron cross section data from the ENDF/B Evaluated Nuclear Data File to the KEDAK-format. Work document presented at the meeting on "Conversion of Formats and Related problems".
Bologna, June 7-9, 1972

/2/ M.K. Drake
Data Formats and Procedures for the ENDF Neutron Cross Section Library.
National Neutron Cross Section Center. Brookhaven National Laboratory, Upton, New York
BNL 50274, October 1970

/3/ Henry C. Honeck
Retrieval Subroutines for the ENDF/B System.
ENDF-110.
National Neutron Cross Section Center. Brookhaven National Laboratory, Upton, New York
BNL 13582, September 1967. Revised April 1969.

/4/ B. Krieg
Handling and Service Programs for the Karlsruhe Nuclear Data File KEDAK.
Part I: Management and Retrieval Programs.
Institut für Neutronenpyhsik und Reaktortechnik, Gesellschaft für Kernforschung mbH., Karlsruhe
KFK 1725, Juni 1973

/5/ B. Goel, B. Krieg
Status of the Nuclear Data Library KEDAK-3. October 1975.
Institut für Neutronenpyhsik und Reaktortechnik, Gesellschaft für Kernforschung mbH., Karlsruhe
KFK 2234. NEANDC(E)171 >>U<<, Dezember 1975

/6/ Nuclear Program Abstracts. Edited by the NEA Computer Program Library. Nuclear Energy Agency. OECD (Organisation for Economic Co-Operation and Development, Ispra (Varese), 1975

/7/ I.P. Mysovskih
Leningrad State University
Lectures on numerical methods.
Wolters-Noordhoff Publishing
Groningen 1969

/8/ G.I. Bell and S. Glasstone
Nuclear reactor theory
Van Nostrand Reinhold Company, October 1970.

/9/ IBM System /360 and System /370. FORTRAN IV Language.
File No. S360-25
Order No. GC28-6515-9
31.03.1973

/10/ IBM System /360 Operating System
FORTRAN IV (G+H)
Programmer's Guide
Order No. C28-6817

/11/ IBM System /360 Operating System
Supervisor and Data Management
Macro-Instructions.
Order No. C28-6647

/12/ DIN Taschenbuch 25
Informationsverarbeitung
Beuth-Vertrieb GmbH, ISBN 3-410-10037-7, 1972

/13/ Otto Eggenberger
A8FORM. Ausdruck von Texten in Großformat.
February 1970. Not published.

/14/ G. Arnecke, H. Bachmann
DINF und DEFI. Programme zur Dynamisierung des DEFINE FILE Statements im IBM-FORTRAN IV.
September 1972. Not published.

/15/ Ch. Hinze
FREESP. Eine Subroutine zur Bestimmung des noch freien Kernspeichers für FORTRAN-Benutzer an der IBM/360-65.
Not published.

/16/ W. Höbel
XTAREA/REXTAR. Eine Subroutine zur volldynamischen Dimensionierung von IBM/360-FORTRAN-Programmen.
October 1969. Not published.

/17/ Ch. Hinze
ZEIT(Z). Zeitkontrollroutine für FORTRAN-Benutzer an der Rechenanlage IBM/360-65.
Not published.

/18/ Ch. Hinze
DATUM(DDAT,DZEIT). Datums- und Uhrzeitroutine für FORTRAN-Benutzer an der Rechenanlage IBM/360-65.
Not published.

/19/ K. Gogg
CONVX. Fortran-Unterprogramm für die IBM/360 zur Umwandlung von in maschineninterner Darstellung vorliegenden Fest- und Gleitkommazahlen in alphanumerische Darstellung.
Not published.

2.5 Source program

In the section 2.5.3 a complete listing of the source program is given. This listing includes all programs described in 2.3.5.1 until 2.3.5.91. No listing of the auxiliary library programs (2.3.5.92 - 2.3.5.99) is given.

To facilitate the use of the source program listing two tabulations will be given. In the first (2.5.1) you will find the subroutine and function names in the sequence of their appearance in the list of the source program. The subroutines and functions are numbered (first column). The names are given in the second column. The type, listed in the third column, means:

M	main program
SF	FORTRAN subroutine
B	BLOCK DATA program
RF4	FORTRAN function delivering REAL*4 results
SA	ASSEMBLER subroutine

In the fourth column the number of the section, where the description is given, can be found. Finally the fifth column contains entry names, if such one are existing. In section 2.5.2 an alphabetic listing of all subroutine-, function-, and entry names is given (excluding the auxiliary programs (see 2.3.5.92 - 2.3.5.99), which are contained in program libraries). The number defined in 2.5.1 is associated to each name. If the specified name is an entry of a subroutine, the subroutine name is listed in column 2.

2.5.1 List of the subroutine and function names in the sequence of their appearance in the source program list

number	name	type	described	entry names
1	MAIN	M	2.3.5.1	-
2	TITEL	SF	2.3.5.2	-
3	PRIZEI	SF	2.3.5.45	-
4	DADEFI	SF	2.3.5.7	-
5	GSPACE	SF	2.3.5.5	-
6	BLOCK DATA	B	2.3.5.4	-
7	PART1	SF	2.3.5.8	-
8	INITNL	SF	2.3.5.9	-
9	INGRID	SF	2.3.5.10	-
10	REGINE	SF	2.3.5.11	-
11	MARY	SF	2.3.5.12	-
12	SORTRE	SF	2.3.5.14	-
13	MARCEL	SF	2.3.5.20	MARGOT
14	MILLI	SF	2.3.5.22	-
15	MILLI1	SF	2.3.5.23	-
16	RESERR	SF	2.3.5.18	-
17	MELA	SF	2.3.5.24	-
18	MARYO1	SF	2.3.5.15	-
19	MARYO2	SF	2.3.5.19	-
20	FACTS	SF	2.3.5.13	-
21	FACPHI	SF	2.3.5.21	-
22	ANNICK	SF	2.3.5.25	-
23	ANNIO1	SF	2.3.5.30	-
24	ERF	SF	2.3.5.27	-
25	GAUSS	SF	2.3.5.28	-

number	name	type	described	entry names
26	BINT	RF4	2.3.5.29	-
27	GNRL	SF	2.3.5.26	-
28	ADD	RF4	2.3.5.68	-
29	SUB	RF4	2.3.5.69	-
30	MULT	RF4	2.3.5.70	-
31	MARION	SF	2.3.5.35	-
32	MYRIAM	SF	2.3.5.33	-
33	SOPHIE	SF	2.3.5.36	-
34	ELIMZ	SF	2.3.5.38	-
35	SUZY	SF	2.3.5.37	-
36	DZMSCH	SF	2.3.5.40	-
37	ADDPNT	SF	2.3.5.39	-
38	SUPNEG	SF	2.3.5.41	-
39	ISABEL	SF	2.3.5.42	-
40	ITAB1	SF	2.3.5.71	EITAB1
41	WCONT	SF	2.3.5.50	-
42	BETSY	SF	2.3.5.34	-
43	LEGP	RF4	2.3.5.43	-
44	RDAB	SF	2.3.5.77	RDA1
	-	-	-	RDA2
	-	-	-	RDA3
	-	-	-	RDA4
45	FILE5	SF	2.3.5.44	-
46	CHGINT	SF	2.3.5.73	-
47	DANI	SF	2.3.5.74	-
48	COMB	SF	2.3.5.66	IICOMB
49	ERROR	SF	2.3.5.46	-
50	MAGGY	SF	2.3.5.31	-

number	name	type	described	entry names
51	SEARCH	SF	2.3.5.48	-
52	TERP1	SF	2.3.5.72	-
53	WDA	SF	2.3.5.76	WDA1
	-	-	-	WDA2
	-	-	-	WDA3
54	WREC	SF	2.3.5.51	IWREC
55	WRECS	SF	2.3.5.78	WRECS1
	-	-	-	WRECS2
	-	-	-	WRECS3
56	CROP	SF	2.3.5.75	-
57	LUCY	SF	2.3.5.16	-
58	RRECS	SF	2.3.5.79	IRRECS
	-	-	-	RRECS4
59	PRTPID	SF	2.3.5.57	-
60	PUTPID	SF	2.3.5.63	-
61	RREC	SF	2.3.5.49	IRREC
62	PRHOL	SF	2.3.5.56	-
63	PUHOL	SF	2.3.5.62	-
64	PRCONT	SF	2.3.5.52	-
65	PUCONT	SF	2.3.5.58	-
66	PRLIST	SF	2.3.5.53	-
67	PULIST	SF	2.3.5.59	-
68	PRTAB1	SF	2.3.5.54	-
69	PUTAB1	SF	2.3.5.60	-
70	PRTAB2	SF	2.3.5.55	-
71	PUTAB2	SF	2.3.5.61	-
72	CXFP	SF	2.3.5.64	-
73	XTEXT	SF	2.3.5.47	-

number	name	type	described	entry names
74	EPT	SF	2.3.5.17	-
75	SCALE	SF	2.3.5.65	GENER
	-	-	-	INITSC
76	COMBP	SF	2.3.5.67	-
77	DENISE	SF	2.3.5.32	-
78	FETCH	SF	2.3.5.81	IFETCH
79	STORE	SF	2.3.5.80	ISTORE
80	DELETE	SF	2.3.5.82	-
81	FPDS	SF	2.3.5.83	IFPDS
82	IPDS	SF	2.3.5.84	IIPDS
83	LRIDS	SF	2.3.5.85	-
84	SICKY	SF	2.3.5.86	-
85	LISA	SF	2.3.5.87	-
86	TABPRT	SF	2.3.5.88	-
87	PAGE	SF	2.3.5.89	INITPG
88	SPLIT	SF	2.3.5.90	-
89	CON	SF	2.3.5.91	-
90	PRIEIN	SF	2.3.5.3	-
91	BUFREG	SA	2.3.5.6	-

The meaning of the abbreviations in the type column may be found in section 2.5.

2.5.2 Index of the subroutine, function and entry names in alphabetic order

name	entry of	number in section 2.5.1
ADD	-	28
ADDPNT	-	37
ANNICK	-	22
ANNIO1	-	23
BETSY	-	42
BINT	-	26
BLOCK DATA	-	6
CHGINT	-	46
COMB	-	48
COMBP	-	76
CON	-	89
CROP	-	56
CXFP	-	72
DADEFI	-	4
DANI	-	47
DELETE	-	80
DENISE	-	77
DZMSCH	-	36
EITAB1	ITAB1	40
ELIMZ	-	34
EPT	-	74
ERF	-	24
ERROR	-	49
FACPHI	-	21
FACTS	-	20
FETCH	-	78

name	entry of	number in section 2.5.1
FILE5	-	45
FPDS	-	81
GAUSS	-	25
GENER	SCALE	75
GNRL	-	27
GSPACE	-	5
IFETCH	FETCH	78
IFPDS	FPDS	81
IICOMB	COMB	48
IIPDS	IPDS	82
INGRID	-	9
INITNL	-	8
INITPG	PAGE	87
INITSC	SCALE	75
IPDS	-	82
IRREC	RREC	61
IRRECS	RRECS	58
ISABEL	-	39
ISTORE	STORE	79
ITAB1	-	40
IWREC	WREC	54
LEGP	-	43
LISA	-	85
LRIDS	-	83
LUCY	-	57
MAGGY	-	50
MAIN	-	1

name	entry of	number in section 2.5.1
MARCEL	-	13
MARGOT	MARCEL	13
MARION	-	31
MARY	-	11
MARYO1	-	18
MARYO2	-	19
MELA	-	17
MILLI	-	14
MILLI1	-	15
MULT	-	30
MYRIAM	-	32
PAGE	-	87
PART1	-	7
PRIEIN	-	90
PRIZEI	-	3
PRCONT	-	64
PRHOL	-	62
PRLIST	-	66
PRTAB1	-	68
PRTAB2	-	70
PRTPID	-	59
PUCONT	-	65
PUHOL	-	63
PULIST	-	67
PUTAB1	-	69
PUTAB2	-	71
PUTPID	-	60

name	entry of	number in section 2.5.1
RDAB	-	44
RDA1	RDAB	44
RDA2	RDAB	44
RDA3	RDAB	44
RDA4	RDAB	44
REGINE	-	10
RESERR	-	16
RREC	-	61
RRECS	-	58
RRECS4	RRECS	58
SCALE	-	75
SEARCH	-	51
SICKY	-	84
SOPHIE	-	33
SORTRE	-	12
SPLIT	-	88
STORE	-	79
SUB	-	29
SUPNEG	-	38
SUZY	-	35
TABPRT	-	86
TERP1	-	52
TITEL	-	2
WCONT	-	41
WDA	-	53
WDA1	WDA	53
WDA2	WDA	53

name	entry of	number in section 2.5.1
WDA3	WDA	53
WREC	-	54
WRECS	-	55
WRECS1	WRECS	55
WRECS2	WRECS	55
WRECS4	WRECS	55
XTEXT	-	73

2.5.3 Source program listing

The following pages contain a listing of the whole source program. This is a listing of all the subroutines and functions described in 2.3.5.1 until 2.3.5.91. A listing of the auxiliary programs described in 2.3.5.92 until 2.3.5.99 is not given.

```
C*****************************************************************      10
C*                                                               *      20
C*                  P R O G R A M   B R I G I T T E              *      30
C*                                                               *      40
C*       TRANSLATION OF ENDF/B NUCLEAR DATA                      *      50
C*                  IN KEDAK FORMAT                              *      60
C*                                                               *      70
C*                  WRITTEN BY J.C.SCHEPERS AND J.DANIELS        *      80
C*                            JUNE 1971                          *      90
C*                  CHANGED BY E.STEIN                           *     100
C*        D A T E   O F   L A S T   C H A N G E :   ALGUST  1976 *     110
C*                                                               *     120
C*****************************************************************     130
      DIMENSION IF(6),F(1)                                              140
      EQUIVALENCE (IF(4),NRIQ1),(IF(5),NRIQ11),(IF(6),NRIQ12),          150
     1(IF(1),NBLK1),(IF(2),NBLK11),(IF(3),NBLK12)                       160
      COMMON/INOUT/NR1(133),IVER,IVGENR,IVJNT,IVX,IVY,IVB,IVDENS,NR2(12) 170
     *,ANFZEI                                                           180
      DATA IU1,IU11,IU12,FORMAT/1,11,12,'U   '/                         190
      CALL ESPIE                                                        200
      UHRZEI=0.0                                                        210
      ANFZEI=ZEIT(UHRZEI)                                               220
      CALL TITEL                                                        230
      CALL PRIEIN(5,6,60)                                               240
      CALL GSPACE(F)                                                    250
      CALL DADEFI(IF)                                                   260
      CALL DEFI(IU1,NRIQ1,FORMAT,NBLK1,K1,                              270
     1          IU11,NRIQ11,FORMAT,NBLK11,K11,                          280
     2          IU12,NRIQ12,FORMAT,NBLK12,K12)                          290
      IBK=1+IVER                                                        300
      ITAB=IBK+IVGENR                                                   310
      INBT=ITAB                                                         320
      IJNT=IBK+IVJNT                                                    330
      IX=IBK+IVX                                                        340
      IY=IBK+IVY                                                        350
      IB=IBK+IVB                                                        360
      IA=IBK+IVDENS                                                     370
    1 CALL PART1(NIS,NISO,IF4,IF5,F(IBK),F(INBT),F(IJNT),F(IX),         380
     *             F(IY),F(IB),F(IA))                                   390
      CALL SICKY(NIS,NISO,IF4,IF5,F(IBK),F(ITAB),F(ITAB))               400
      GOTO 1                                                            410
      END                                                               420
```

```
      SUBROUTINE TITEL                                                   10
C     DRUCKT TITELSEITE.                                                 20
      COMMON/INOUT/ NO                                                   30
      I=1                                                                40
      WRITE(NO,450)                                                      50
  450 FORMAT('1')                                                        60
    1 CALL RAMANF(NO)                                                    70
      CALL A8FORM(NO,'BRIGITTE')                                         80
      CALL RAMEND(NO)                                                    90
      IF(I.GT.1) GOTO 333                                               100
      WRITE(NO,99)                                                      110
   99 FORMAT('0')                                                       120
      WRITE(NO,100)                                                     130
  100 FORMAT(36X,60('*')/36X,60('*'))                                   140
      WRITE(NO,101)                                                     150
  101 FORMAT(36X,'**',56X,'**')                                         160
      WRITE(NO,102)                                                     170
  102 FORMAT(36X,'**',13X,'C O N V E R S I O N   F R O M',14X,'**')     180
      WRITE(NO,101)                                                     190
      WRITE(NO,103)                                                     200
  103 FORMAT(36X,'**',22X,'E N D F / B',23X,'**')                       210
      WRITE(NO,101)                                                     220
      WRITE(NO,104)                                                     230
  104 FORMAT(36X,'**',21X,'TO   K E D A K',21X,'**')                    240
      WRITE(NO,101)                                                     250
      WRITE(NO,100)                                                     260
      WRITE(NO,105)                                                     270
  105 FORMAT(//)                                                        280
      I=I+1                                                             290
      GOTO 1                                                            300
  333 RETURN                                                            310
      END                                                               320
```

```
      SUBROUTINE PRIZEI                                                  10
C     DRUCKT DATUM, VERWEILZEIT UND CPU-ZEIT.                            20
      REAL*8 UHRZEI,DATE                                                 30
      COMMON/INOUT/ NO,NR1(151),ANFZEI                                   40
C                                                                        50
      WRITE(NO,100)                                                      60
  100 FORMAT(' ',120('*'))                                               70
      CALL DATUM(DATE,UHRZEI)                                            80
      DURA=ZEIT(ANFZEI)                                                  90
      WRITE(NO,101) DATE,UHRZEI,DURA                                    100
  101 FORMAT(' ',31('*'),' DATE: ',A8,'. TIME: ',A8,'. CPU-TIME: ',F9.3, 110
     *' SEC. ',31('*'))                                                 120
      WRITE(NO,100)                                                     130
      RETURN                                                            140
      END                                                               150
```

```
      SUBROUTINE DADEFI(IF)                                              10
      REAL*8 DD1,DD11,DD12                                               20
      DIMENSION IF(6)                                                    30
      DATA DD1,DD11,DD12/'FT01F001','FT11F001','FT12F001'/               40
      CALL DINF(DD1,IF(1),IF(4))                                         50
      CALL DINF(DD11,IF(2),IF(5))                                        60
      CALL DINF(DD12,IF(3),IF(6))                                        70
      DO 1 I=1,3                                                         80
    1 IF(I)=IF(I)/4                                                      90
```

```
      RETURN                                                            100
      END                                                               110
      SUBROUTINE GSPACE(F)                                               10
C     DIE GROESSE DES VON PROGRAMMEN UND PUFFERN NICHT BENUTZTEN TEILS   20
C     DER REGION WIRD BESTIMMT. DAVON ABGEZOGEN WIRD DER PLATZ, DER      30
C     SPATER FUER PUFFER VERWENDUNG FINDEN KOENNTE, SOWIE DER VOM        40
C     SYSTEM BENOETIGTE PLATZ. DER SO VERBLEIBENDE TEIL WIRD DEM         50
C     PROGRAMM ALS DATENSPEICHER ZUR VERFUEGUNG GESTELLT.                60
C                                                                        70
      DIMENSION F(1)                                                     80
      COMMON/INOUT/ NO,NI,NR1(123),NBX,N1X,N2X,N,NBKX,JX,IADIM,ITDIM,    90
     *              IVER,IVGENR,IVJNT,IVX,IVY,IVB,IVDENS                100
      NAMELIST/SPACE/ NBX,N1X,N2X,NBKX,IADIM,ITDIM                      110
      WRITE(NO,100)                                                     120
  100 FORMAT('0DYNAMIC WORKING SPACE ALLOCATION.'/'+',33('_'))          130
      CALL FREESP(J)                                                    140
      WRITE(NO,101) J                                                   150
  101 FORMAT(' FREE REGION=',I4,' K-BYTE.')                             160
      IF(J.GE.10) GOTO 1                                                170
    4 WRITE(NO,102)                                                     180
  102 FORMAT(' ***** NOT ENOUGH REGION.')                               190
    2 WRITE(NO,103)                                                     200
  103 FORMAT(' ***** PROGRAM ENDED DUE TO ERROR.')                      210
      STOP                                                              220
    3 WRITE(NO,111) IBUF                                                230
  111 FORMAT(' ***** ERROR NUMBER ',I2,' IN BUFREG.')                   240
      GOTO 2                                                            250
C                                                                       260
C     PLATZ FUER PUFFER                                                 270
C                                                                       280
    1 CALL BUFREG(IBUF,83)                                              290
      ISYS=6                                                            300
      WRITE(NO,104) IBUF,ISYS                                           310
  104 FORMAT(' ',I4,' K-BYTE WILL BE NEEDED AS SPACE FOR BUFFERS.'/     320
     1       ' ',I4,' K-BYTE ARE FOR SYSTEM USE.')                      330
      IDATA=J-(ISYS+IBUF)                                               340
      IF(IDATA.LT.10) GOTO 4                                            350
      WRITE(NO,105) IDATA                                               360
  105 FORMAT('0',I4,' K-BYTE ARE USED FOR DATA FIELDS.')                370
      K2=1024*IDATA                                                     380
      CALL XTAREA(K1,K2,K3,F(1))                                        390
      IF(K2.NE.0) GOTO 5                                                400
      WRITE(NO,106)                                                     410
  106 FORMAT(' ***** ERROR IN XTAREA.')                                 420
      GOTO 2                                                            430
    5 IVER=(K1-K3)/4                                                    440
      KW=256*IDATA                                                      450
C                                                                       460
C     VERTEILUNG DER DATEN REGION.                                      470
C                                                                       480
      READ(NI,SPACE)                                                    490
      WRITE(NO,SPACE)                                                   500
C     JX WIRD GLEICH IADIM GESETZT.                                     510
C     NBKX MUSS EIN VIELFACHES VON 2 SEIN.                              520
C     ITDIM WIRD GLEICH MAXO(ITDIM,NBX+2*(N1X+N2X)+IADIM) GESETZT.      530
C     NBX=ITDIM-IADIM-2*(N1X+N2X)                                       540
C     ITDIM WIRD AUF VIELFACHES VON 880 ABGERUNDET.                     550
C                                                                       560
      IDUR=1                                                            570
    6 JX=IADIM                                                          580
      NBKX=(NBKX/2)*2                                                   590
      ITDIM=MAXO(ITDIM,NBX+2*(N1X+N2X)+IADIM)                           600
      NBX=ITDIM-IADIM-2*(N1X+N2X)                                       610
      ITDIM=(ITDIM/880)*880                                             620
      IF(IDUR.EQ.1) WRITE(NO,112)                                       630
  112 FORMAT(' PLANNED DIMENSIONS:')                                    640
      IF(IDUR.EQ.2) WRITE(NO,113)                                       650
  113 FORMAT(' NEW DIMENSIONS:')                                        660
      WRITE(NO,107) NBX,N1X,N1X,N2X,N2X,NBKX,IADIM,ITDIM                670
  107 FORMAT(' B(',I5,'),NBT(',I4,'),JNT(',I4,'),X(',I6,'),Y(',         680
     2I6,'),BK(',I6,'),A(',I6,'),TAB(',I6,')')                          690
      ISUM=NBKX+NBX+2*(N1X+N2X)+IADIM                                   700
      IF(ISUM.LE.KW) GOTO 7                                             710
      WRITE(NO,108)                                                     720
  108 FORMAT('0DATA REGION NOT LARGE ENOUGH TO FULFILL PLANNED DIMENSION 730
     1S.'/' DIMENSIONS WILL BE SHORTENED.')                             740
    9 FAK=(1.0*KW)/ISUM                                                 750
      NBX=NBX*FAK                                                       760
      N1X=N1X*FAK                                                       770
      N2X=N2X*FAK                                                       780
      NBKX=NBKX*FAK                                                     790
      JX=JX*FAK                                                         800
      IADIM=IADIM*FAK                                                   810
      ITDIM=ITDIM*FAK                                                   820
      IDUR=2                                                            830
      GOTO 6                                                            840
    7 IF(1.1*ISUM.GT.KW.OR.IDUR.GE.2) GOTO 8                            850
      WRITE(NO,114)                                                     860
  114 FORMAT('0PLANNED DIMENSIONS WILL BE INCREASED, BECAUSE THERE IS EN 870
     1OUGH DATA REGION.')                                               880
      GOTO 9                                                            890
C                                                                       900
C     DISPLACEMENTS RELATIV ZU BK(1).                                   910
C                                                                       920
    8 IVGENR=NBKX                                                       930
      IVJNT=IVGENR+N1X                                                  940
      IVX=IVJNT+N1X                                                     950
      IVY=IVX+N2X                                                       960
      IVB=IVY+N2X                                                       970
      IVDENS=IVB+NBX                                                    980
      WRITE(NO,109) NBKX,IVGENR,ITDIM,IVGENR,N1X,IVJNT,N1X,IVX,N2X,IVY, 990
     1N2X,IVB,NBX,IVDENS,IADIM                                         1000
  109 FORMAT('0OVERLAY OF DATA REGION.'//                               1010
     1' DISPLACEMENT RELATIV TO BK(1)  ARRAY NAME  LENGHT'/             1020
     2/'+',62('_')//                                                   1030
     3'            =        0            BK          NBKX=',           1040
```

```
     3I7/                                                                    1050
     4'       IVGENR=',I9,12X,'TAB',8X,'ITDIM=',I6/                          1060
     5'       IVGENR=',I9,12X,'NBT',8X,'N1X=',I8/                            1070
     6'       IVJNT =',I9,12X,'JNT',8X,'N1X=',I8/                            1080
     7'       IVX   =',I9,12X,'X  ',8X,'N2X=',I8/                            1090
     8'       IVY   =',I9,12X,'Y  ',8X,'N2X=',I8/                            1100
     9'       IVB   =',I9,12X,'B  ',8X,'NBX=',I8/                            1110
     A'       IVDENS=',I9,12X,'A  ',8X,'IADIM=',I6//)                        1120
      WRITE(NO,110)                                                          1130
  110 FORMAT(' END OF DYNAMIC WORKING SPACE ALLOCATION.'/'1')                 1140
      RETURN                                                                 1150
      END                                                                    1160
```

```
      BLOCK DATA                                                              10
      COMMON/TAB/ENDTAB(25),KEDTAB(50),KEDR(240),NCNAME(39),NRIF3,NRIF4       20
      DIMENSION KDR1(123),KDR2(84)                                            30
      EQUIVALENCE (KEDR(1),KDR1(1)),(KEDR(124),KDR2(1))                       40
      INTEGER ENDTAB                                                          50
      DATA ENDTAB/1,2,3,4,16,17,18,22,23,24,25,27,28,29,                      60
     *           102,103,104,105,106,107,108,201,206,207,251/                 70
      DATA KEDTAB/'SGT ',' ','SGN ',' ',                                      80
     *            'SGX ',' ','SGI ',' ',                                      90
     *            'SG2N',' ','SG3N',' ',                                     100
     *            'SGF ',' ','SGIA',' ',                                     110
     *            'SGI3','A ','SG2N','A ',                                   120
     *            'SG3N','A ','SGA ',' ',                                    130
     *            'SGIP',' ','SGNI',' ',                                     140
     *            'SGG ',' ','SGP ',' ',                                     150
     *            'SGD ',' ','SGH3',' ',                                     160
     *            'SGHE','3 ','SGAL','P ',                                   170
     *            'SG2H','E ','SGTR',' ',                                    180
     *            'ETA ',' ','ALPH','A ',                                    190
     *            'MUEL',' '/                                                200
      COMMON/XBAR/XX(4,10)                                                   210
      DATA XX/0.005252,0.051755,0.112925,0.169150,0.237171,0.163095,         220
     *  0.265600,0.340780,0.103126,0.288421,0.404385,0.480571,0.207836,      230
     *  0.431766,0.547724,0.617825,0.359852,0.599210,0.704048,0.762381,      240
     *  0.574283,0.800560,0.882440,0.922898,0.879334,1.053224,1.096835,      250
     *  1.111387,1.334810,1.393010,1.374373,1.350285,2.105227,1.916230,      260
     *  1.786357,1.697511,4.390800,3.301643,2.824583,2.546602/               270
      DATA KDR1/'RANG','RES ',14511,'ISOT','1 ',14580,'ISOT','2 ',           280
     X    14590,'ISOT','3 ',14600,'PLNL','E ',14570,                        290
     X          'CHIC','R ',14560,'RES ',' ',21520,                         300
     X          'ST  ',' ',21530,'STD ',' ',21540,                          310
     X          'STGF',' ',21550,'SGT ',' ',30010,                          320
     X          'SGN ',' ',30020,'SGX ',' ',30030,                          330
     X          'SGI ',' ',30040,'SGIZ',' ',30050,'SGIZ','C ',             340
     X    30051,'SG2N',' ',30160,'SG3N',' ',30170,                          350
     X          'SGF ',' ',30190,'SGIA',' ',30220,                          360
     X          'SGI3','A ',30230,'SG2N','A ',30240,                        370
     X          'SG3N','A ',30250,'SGA ',' ',30270,                         380
     X          'SGIP',' ',30280,'SGNI',' ',30290,                          390
     X          'SGG ',' ',31020,'SGP ',' ',31030,                          400
     X          'SGD ',' ',31040,'SGH3',' ',31050,                          410
     X          'SGHE','3 ',31060,'SGAL','P ',31070,                        420
     X          'SG2H','E ',31080,'SGTR',' ',32010,                         430
     X          'ETA ',' ',32060,'ALPH','A ',32070,                         440
     X          'MUEL',' ',32510,'NUE ',' ',34520,                          450
     X          'NUEP',' ',34550,'CHIF',' ',34610,                          460
     X          'CHIF','D ',34620/                                          470
      DATA KDR2/                    'SGNL',' ',40021,                       480
     *          'SGNC',' ',40022,'SGIL',' ',40041,                          490
     X          'SGIC',' ',40042,'SGIL','Z ',40051,                         500
     X          'SGIC','Z ',40052,'SGNI','L ',40291,                        510
     X          'SGNI','C ',40292,'LEGN','L ',44631,                        520
     X          'LEGN','C ',44632,'LEGI','L ',44641,                        530
     X          'LEGI','C ',44642,'LEGI','LZ',44651,                        540
     X          'LEGI','CZ',44652,'LGNI','L ',44661,                        550
     X          'LGNI','C ',44662,'CHII',' ',50040,                         560
     X          'CHII','C ',50051,'SED1','C ',50053,                        570
     X          'CHI2','N ',50160,'SED2','N ',50163,                        580
     X          'CHI3','N ',50170,'SED3','N ',50173,                        590
     X          'SEDF',' ',54523,'CHIF','Z ',54610,                         600
     X          'SEDF','P ',54613,'CHIF','CZ',54620,                        610
     X          'SEDF','D ',54623/                                          620
      COMMON/INOUT/ NO,NI,NR1(123),NBX,N1X,N2X,N,NBKX,JX,IADIM,ITDIM,        630
     *             NR2(21)                                                  640
      DATA NO,NI,NBX,N1X,N2X,NBKX,JX,IADIM,ITDIM/6,5,3000,500,5000,          650
     *12000,20000,20000,32560/                                               660
      END                                                                    670
```

```
      SUBROUTINE PART1(NIS,MISO,KIF,KIF5,BK,NBT,JNT,X,Y,B,A)                  10
      COMMON/INOUT/NR1(2),NT,NR2(139),                                        20
     *LABEL,LNER,ITAPE,MATP,EB,NISO,IF4,IF5                                   30
      DIMENSION BK(1),NBT(1),JNT(1),X(1),Y(1),B(1),A(1)                       40
      CALL INITNL(MISO,&21)                                                   50
      CALL IWREC(NBT,JNT,X,Y,B)                                               60
      CALL IRREC(NBT,JNT,X,Y,B)                                               70
      CALL EITAB1(NBT,JNT,X,Y)                                                80
      CALL WRECS1(NBT,JNT,X,Y,B)                                              90
      CALL WDA1(BK)                                                          100
      KIF=IF4                                                                110
      KIF5=IF5                                                               120
      CALL INGRID(MATP,LABEL,LRP,LFI,LNU,IF4,IF5)                            130
      IF(IF4.EQ.1)GOTO 11                                                    140
      IF(IF5.EQ.1)GOTO 12                                                    150
    3 CALL REGINE(MATP,LNER,LISRES,LISUNR,NIS,LFWRES,LFW,EB,LRP,             160
     *     NISO,BK,NBT,JNT,X,Y,B,A)                                          170
      CALL DENISE(MATP,LRP,LISRES,LISUNR,NIS,LFW,LNU,BK,NBT,JNT,X,Y,B,A)     180
      CALL SUZY(LNU,ITAPE,BK,BK,BK,BK,NBT,JNT,X,Y,B)                         190
      IF(IF4.EQ.-1)GOTO 14                                                   200
   11 CALL ISABEL(MATP,IF4,BK,BK,BK,NBT,JNT,X,Y,B,A,A,A)                     210
   14 IF(IF4.EQ.1.OR.IF5.EQ.-1) GOTO 13                                      220
   12 CALL FILES(MATP,IF5,BK,BK,NBT,JNT,X,Y,B,B)                             230
```

```
   13 CALL WDA3(IW)                                                     240
      RETURN                                                            250
   21 REWIND NT                                                         260
      STOP                                                              270
      END                                                               280
```

```
      SUBROUTINE INITNL(MISO,*)                                          10
      LOGICAL*4 SALINE                                                   20
      COMMON/INOUT/NO,NI,NT,MODE,LEDIT1,LEDIT2,LEDITS,KEDAVE,ABIS(20),   30
     *            NOTTRL(25),BIB12,SALINE,                               40
     *NAMKDK,MATKDK,NTYP,NAMISO(10),MATISO(10),EBISO(10),ICOMB,NR1(15),   50
     *EPSRES,SCREPS,LABEL,LNER,ITAPE,MATP,EB,NISO,IF4,IF5,FAICOD,MLSLSW,  60
     *NR3(1),ELINEG                                                      70
      REAL*8 NAMKDK,NAMISO,NOTTRL,BLANC                                  80
      DATA BLANC/'        '/                                             90
      COMMON/TAB/NR2(315),NCNAME(39),NRIF3,NRIF4                        100
      INTEGER BIB12,ELINEG                                              110
      NAMELIST/INPUT/MODE,LABEL,LNER,LEDIT1,LEDIT2,LEDITS,KEDAVE,       120
     *     ITAPE,MATP,NAMKDK,MATKDK,EB,NISO,IF4,ICOMB,IF5,NOTTRL,       130
     *     BIB12,SALINE,EPSRES,SCREPS,FAICOD,MLSLSW,ELINEG              140
      NAMELIST/ISO/NAMISO,MATISO,EBISO                                  150
      NI=5                                                              160
      NO=6                                                              170
      NT=9                                                              180
      DO 20 I=1,39                                                      190
   20 NCNAME(I)=0                                                       200
      DO 25 I=1,25                                                      210
   25 NOTTRL(I)=BLANC                                                   220
C-----MODE=DATA INPUT FORMAT DEFINED AS FOLLOWS---                      230
C          =1 BINARY TAPE, STANDARD ARRANGEMENT---                      240
C          =3 BCD CARD IMAGE TAPE---                                    250
C-----LABEL=IDENTIFICATION NUMBER OF THE INPUT TAPE,TREATED AS FOLLOWS  260
C           =GREATER THAN ZERO---READ TPID AND CHECK LABEL              270
C           =EQUAL ZERO---READ TPID. DO NOT CHECK LABEL                 280
C           =LESS THAN ZERO---DO NOT READ TPID OR CHECK LABEL           290
C-----LNER=0 CALCULATE ENERGIES USING 51 ENERGY POINTS PER RESONANCE     300
C          >1 CALCULATE ENERGIES USING LNER ENERGY POINTS PER RESONANCE  310
C-----KEDAVE=0  O L D   KEDAK VERSION.                                  320
C            1  N E W   KEDAK VERSION.                                  330
C-----ICOMB=0  THE INTERPOLATION SCHEME IN COMBP IS CHOOSEN             340
C              BY THE ROUTINE.                                          350
C            1 THE INTERPOLATION SCHEME USED IN COMBP IS                360
C              ALWAYS LINEAR-LINEAR.                                    370
C-----BIB12=1,2,....,9,10 NUMBER OF BLOCKS OF 400 WORDS FOR TABLE OF    380
C                         CONTENTS FOR FILE 12.                         390
C-----MLSLSW=SWITCH FOR USING MULTILEVEL OR SINGLELEVEL BREIT-WIGNER     400
C            FORMALISM IN RESOLVED RESONANCE CALCULATING SECTION.       410
C           =0 THE ENDF/B RECOMMENDED FORMALISM IS USED.                420
C           =1 SINGLE LEVEL FORMALISM IS USED INDEPENDENTLY FROM        430
C              THE ENDF/B RECOMMENDED FORMALISM.                        440
      ICOMB=0                                                           450
      MODE=3                                                            460
      LABEL=-1                                                          470
      LNER=0                                                            480
      LEDIT1=1                                                          490
      LEDIT2=1                                                          500
      LEDITS=0                                                          510
      KEDAVE=1                                                          520
      ITAPE=0                                                           530
      NISO=0                                                            540
      IF4=0                                                             550
      IF5=0                                                             560
      NRIF3=0                                                           570
      NRIF4=0                                                           580
      BIB12=1                                                           590
      SALINE=.FALSE.                                                    600
      EPSRES=1.0E-2                                                     610
      SCREPS=5.0E-3                                                     620
      FAICOD=10.0                                                       630
      MLSLSW=0                                                          640
      ELINEG=0                                                          650
      READ(NI,INPUT,END=21)                                             660
      WRITE(NO,INPUT)                                                   670
      BIB12=MAXO(1,MINO(BIB12,10))                                      680
   10 IF(MODE.EQ.1.OR.MODE.EQ.3)GOTO 7                                  690
      WRITE(NO,6)MODE                                                   700
    6 FORMAT(1X,'MODE =',I4,'  OUT OF RANGE')                           710
      STOP                                                              720
    7 IF(NISO.LE.1)GOTO 9                                               730
      READ(NI,ISO)                                                      740
      WRITE(NO,ISO)                                                     750
    9 MISO=NISO                                                         760
      RETURN                                                            770
   21 RETURN 1                                                          780
      END                                                               790
```

```
      SUBROUTINE INGRID(MATP,LABEL,LRP,LFI,LNU,IF4,IF5)                  10
      DIMENSION IORECS(9)                                                20
      EQUIVALENCE (IORECS(1),MAT)                                        30
      COMMON/INOUT/NO,NI,NT,MODE,LEDIT1,NR1(2),KEDAVE,NR2(72),NAMKDK,    40
     *MATKDK,NR3(70),ELINEG                                              50
      COMMON/RECS/MAT,MF,MT,C1,C2,L1,L2,N1,N2,NS                         60
      INTEGER ELINEG                                                     70
      REAL*8 NAMKDK                                                      80
      DO 1 I=1,9                                                         90
    1 IORECS(I)=0                                                       100
      NS=1                                                              110
      IF((MODE.LT.1).OR.(MODE.GT.3)) CALL ERROR(400)                    120
      REWIND NT                                                         130
      IF(LABEL.LT.0)GOTO 100                                            140
      CALL RREC(6,NT,MODE,-1.0)                                         150
      GOTO(120,130,120),MODE                                            160
  120 LABE=MAT                                                          170
      GOTO 140                                                          180
```

```
  130 LABE=IABS(MF)
  140 IF(LABEL.EQ.0)GOTO 100
      IF(LABEL.NE.LABE) CALL ERROR(401)
  100 WRITE(NO,450)
  450 FORMAT('1')
      CALL RAMANF(NO)
      CALL A8FORM(NO,'BRIGITTE')
      CALL RAMEND(NO)
      WRITE(NO,150)MATP,NAMKDK,MATKDK
  150 FORMAT(///' MATERIAL ',6X,I8/1X,
     *'KEDAK NAME ',4X,A8/1X, 'KEDAK NUMBER ',4X,I6)
      WRITE(NO,160)LABEL,MODE
  160 FORMAT(1X,'ENDF/B TAPE       ',I6,' IN MODE ',I2)
      WRITE(NO,451)
  451 FORMAT(' NEGATIVE CROSS SECTION VALUES FOR SGT AND SGN WILL')
      IF(ELINEG.NE.1) WRITE(NO,452)
      IF(ELINEG.EQ.1) WRITE(NO,453)
  452 FORMAT('+',51X,'   N O T   BE CHANGED.'///)
  453 FORMAT('+',51X,' BE   D E L E T E D .'///)
      CALL RAMANF(NO)
      CALL A8FORM(NO,NAMKDK)
      CALL RAMEND(NO)
      WRITE(NO,450)
      CALL RAMANF(NO)
      IF(KEDAVE.EQ.0) CALL A8FORM(NO,' O L D  ')
      IF(KEDAVE.NE.0) CALL A8FORM(NO,' N E W  ')
      CALL A8FORM(NO,'FORM OF ')
      CALL A8FORM(NO,' KEDAK  ')
      CALL RAMEND(NO)
      CALL SEARCH(NT,MODE,MATP,1,451,LNT)
      IF(LNT.EQ.1) CALL ERROR(201)
      LRP=L1
      LFI=L2
      NC=N2
      CALL RREC(5,NT,MODE,-1.0)
      IF452=0
      WRITE(NO,400)
  400 FORMAT(1H1 //30X,'ENDF/B FILE 1'//30X,13('*')//)
      CALL PRIZEI
      IF(LEDIT1.NE.1)GOTO 165
      CALL WREC(5,NC,4)
      WRITE(NO,410)
  410 FORMAT(//20X,'CONTENTS OF THE FILES'//1X,'FILE',3X,'MAT',3X,
     *    'NUMB. OF REC.'/)
  165 DO 170 I=1,NC
      CALL RREC(1,NT,MODE,-1.0)
      IF(L2.EQ.452) IF452=1
      IF(LEDIT1.NE.1)GOTO 170
      WRITE(NO,420)L1,L2,N1
  420 FORMAT(I4,4X,I3,6X,I4)
  170 CONTINUE
      IF(IF4.EQ.1.OR.IF5.EQ.1)GOTO 180
C
C   NUE
C
      IF(LFI.EQ.1.AND.IF452.NE.1) WRITE(NO,1452)
 1452 FORMAT('0',120('+')/' ',29('+'),' WARNING: FISSIONABLE MATERIAL. N
     *O SECTION 452 ( NUE ) FOUND. ',29('+')/' ',120('+'))
      IF(LFI.NE.1.OR.IF452.EQ.0) GOTO 180
      CALL SEARCH(NT,MODE,MATP,1,452,LNT)
      IF(LNT.EQ.1) CALL ERROR(202)
      LNU=L2
      WRITE(NO,250)
  250 FORMAT(1HI,'NUE DATA'/)
      GOTO(190,200),LNU
C DATA IS POLYNOMIAL COEFFICIENTS
  190 CALL RREC(2,NT,MODE,-1.0)
      WRITE(NO,350)
  350 FORMAT(1HO,'DATA IS POLYNOMIAL COEFFICIENTS'/)
      CALL WRECS4(0,4H NUE,4H    )
      CALL WREC(2,NO,4)
      GOTO 240
C LNU=2   DATA IS TABULAR
  200 CALL RREC(3,NT,MODE,-1.0)
      WRITE(NO,260)
  260 FORMAT(1HO,'DATA IS TABULAR'/)
      CALL WRECS2(0,1,452,0.0,0.0)
      CALL WREC(3,NC,4)
  240 CONTINUE
      WRITE(NO,251)
  251 FORMAT(' NUE PROCESSED')
  180 RETURN
      END
```

```
      SUBROUTINE REGINE(MATP,LNER,LISRES,LISUNR,NIS,LFWRES,LFW,EB,
     *     LRP,NISO,BK,NBT,JNT,X,Y,B,AH)
C
C PROCESS RESONANCE DATA
C
      COMMON/INOUT/NO,NI,NT,MODE,NR1(4),ABIS(20),NR2(52),NAMKDK,MATKDK,
     *NR3(21),MATISO(10),EBISO(10),NR4(4),N,NBKX,JX
      COMMON/RECS/MAT,MF,MT,C1,C2,L1,L2,N1
      COMMON/FLCT/ NR5(6),NDIM6,NPEMAX
      DIMENSION AWRISO(20),NAM1(2),BK(1),NBT(1),JNT(1),X(1),Y(1),B(1),
     *AH(1)
      REAL*8 NAM8,NAMKDK
      LOGICAL*1 LNAM(8)
      EQUIVALENCE (NAMKDK,LNAM(1))
      EQUIVALENCE (NAM8,NAM1(1))
      NISMAX=10
      JRES=0
      WRITE(NO,20)
   20 FORMAT(1H1,30X,'ENDF/B FILE 2'//31X,13('*')///)
      CALL PRIZEI
      CALL SEARCH(NT,MODE,MATP,2,151,LNT)
      IF(LNT.EQ.0)GOTO 300
      WRITE(NO,301)
```

```
  301 FORMAT(1X,'FILE 2 NOT PRESENT FOR THIS MATERIAL')
      NIS=1
      LFW=0
      LFWRES=0
      NER=0
      LISRES=1
      LISUNR=1
      BACKSPACE NT
      RETURN
  300 AWR=C2
      NIS=N1
      IF(NIS.GT.NISMAX)CALL ERROR(210)
      WRITE(NO,1)NIS
    1 FORMAT(1HO,'NUMBER OF ISOTOPES IN FILE 2,151',I6)
      IF(NIS.EQ.1)GOTO 450
      IF(NIS.NE.NISO)CALL ERROR(219)
C
C-----LOOP ON ISOTOPES
C
  450 DO 103 MIS=1,NIS
      CALL RREC(1,NT,MODE,-1.0)
C     (Z,A)DESCRIPTION FOR AN ISOTOPE IN A MATERIAL
      ZAI=C1
C ABUNDANCE(WEIGHT FRACTION) OF AN ISOTOPE IN THIS MATERIAL
      ABN=C2
      ABIS(MIS)=ABN
C IF=1 FISSION WIDTHS ARE GIVEN
      LFW=L2
C NUMBER OF ENERGY RANGES FOR THIS ISOTOPE
      NER=N1
      IF(LRP.GT.0) GOTO 251
  250 WRITE(NO,252)
  252 FORMAT(1HO,'NO RESONANCE INFORMATION FOR THIS ISOTOPE')
      GOTO 7
  251 IF(NER.EQ.1) WRITE(NO,5)
    5 FORMAT(1HO,'ONLY ONE ENERGY RANGE FOR RESONANCE PARAMETERS')
      IF(NER.LE.2) GOTO 7
      WRITE(NO,6)
    6 FORMAT(1HO,'MORE THAN TWO ENERGY RANGES---FATAL ERROR...')
      STOP
C
C-----LOOP ON ENERGY RANGES
C
    7 LISRES=1
      LISUNR=1
      DO 2 I=1,NER
      JRES=JRES+1
      CALL RREC(1,NT,MODE,-1.0)
      EL=C1
      EH=C2
C=====LRU=0 NO INFORMATION DATA FOR THE CORRESPONDING ENERGY RANGE
C         =1 RESOLVED RESONANCES
C         =2 UNRESOLVED RESONANCES
      LRU=L1
      LRF=L2
      IF(LRF.LE.2)GOTO 50
      IF(LRF.EQ.3)WRITE(NO,30)
   30 FORMAT(1HO,'REICH-MOORE MULTILEVEL PARAMETERS')
      IF(LRF.EQ.4)WRITE(NO,40)
   40 FORMAT(1HO,'ADLER-ADLER FORMALISM')
      STOP
   50 LRU1=LRU+1
      GOTO(10,3,4),LRU1
C NO INFORMATION
   10 CALL RREC(1,NT,MODE,-1.0)
      SPI=C1
      AP=C2
      AWRI=AWR
      LISRES=1
      LISUNR=1
      GOTO 2
C RESONANCES ARE RESOLVED
    3 LISRES=0
      JXRES=JX/200
      JXEV=10*JXRES
      JXRED=JX-JXEV
      CALL MARY(EL,EH,LRF,ZAI,ABN,LFWRES,NIS,LNER,MIS,LIS1,SPI,AWRI,AP,
     *BK,NBT,JNT,X,Y,B,AH(1+JXEV),JXRED,AH(1),AH(1+JXRES),AH(1+2*JXRES),
     *AH(1+3*JXRES),AH(1+4*JXRES),AH(1+5*JXRES),AH(1+6*JXRES),
     *AH(1+7*JXRES),AH(1+8*JXRES),AH(1+9*JXRES),JXRES)
      LISRES=LIS1
      GOTO 2
C RESONANCES ARE NOT RESOLVED
    4 LISUNR=0
      NDIM6=6
      NPEMAX=JX/45
      CALL ANNICK(EL,EH,LRF,ZAI,ABN,LFW,NIS,MIS,LIS2  ,SPI,AWRI,AP,BK,
     *NBT,JNT,X,Y,B,AH(1),AH(1+NPEMAX),AH(1+2*NPEMAX),AH(1+3*NPEMAX),
     *AH(1+4*NPEMAX),AH(1+5*NPEMAX),AH(1+6*NPEMAX),AH(1+7*NPEMAX),
     *AH(1+8*NPEMAX),AH(1+9*NPEMAX),AH(1+15*NPEMAX),AH(1+21*NPEMAX),
     *AH(1+27*NPEMAX),AH(1+33*NPEMAX),AH(1+39*NPEMAX))
      LISUNR=LIS2
    2 CONTINUE
      A=AMOD(ZAI,1000.)
      AWRISO(MIS)=A
C
C----ISOT1 AND ISOT2
C
      WRITE(NO,302)
  302 FORMAT('1')
      CALL XTEXT(4HISOT,NAM1(1))
      CALL XTEXT(4H    ,NAM1(2))
      CALL AGFORM(NO,NAM8)
      IF(NIS.GT.1.AND.MIS.EQ.1)GOTO 60
      IBCL=1
      GOTO 61
   60 IBCL=2
      WRITE(NO,210)
  210 FORMAT(1HO,'ISOT1 AND ISOT2 APPEAR TWO TIMES'/
     * 1X,' -FOR THE WHOLE MATERIAL'/
```

```
     * 1X,' -FOR THE FIRST ISOTOPE')                                    1340
   61 DO 103 IJ=1,IBCL                                                  1350
      Z=IFIX(ZAI/1000.0)                                                1360
      AWRIS=AWRI                                                        1370
      IF(NIS.GT.1.AND.MIS.EQ.1.AND.IJ.EQ.1)AWRI=AWR                     1380
      ANEUT=1.0086654                                                   1390
      BK(1)=AWRI*ANEUT                                                  1400
      BK(2)=Z                                                           1410
      BK(3)=SPI                                                         1420
      N=3                                                               1430
      WRITE(NO,102)                                                     1440
  102 FORMAT(1HO,'ISOT1'/1X,5('*')//)                                   1450
      WRITE(NO,104)(BK(KB),KB=1,3)                                      1460
  104 FORMAT(1X,'ATOMIC WEIGHT',T35,1PE15.6/                            1470
     *     1X,'ATOMIC NUMBER',T35,1PE15.6/                            1480
     *     1X,'NUCLEAR SPIN OF GROUND STATE',T35,1PE15.6/)            1490
      MATK=MATKDK                                                       1500
      IF(NIS-1)62,62,63                                                 1510
   62 MMIS=1                                                            1520
      GOTO 64                                                           1530
   63 MMIS=MIS                                                          1540
      IF(MIS.GT.1.OR.IJ.EQ.2)GOTO 108                                   1550
      GOTO 64                                                           1560
  108 MATK=MATISO(MIS)                                                  1570
      EB=EBISO(MIS)                                                     1580
      MMIS=MIS+1                                                        1590
   64 CALL WDA2(MATK  ,MMIS,4HISOT,4H1   )                              1600
      WL=455.197*(AWRI+1.0)/AWRI                                        1610
      WRITE(NO,105)                                                     1620
  105 FORMAT(1HO,'ISOT2'/1X,5('*')//)                                   1630
      WRITE(NO,106)WL,AP,EB                                             1640
  106 FORMAT(1X,'REDUCED NEUTRON WAVE LENGTH',T35,1PE15.6,3X,           1650
     *     '10**(-12)CM'/                                             1660
     *     1X,'NUCLEAR RADIUS',T35,1PE15.6,3X,                        1670
     *     '10**(-12)CM'/                                             1680
     *     1X,'BINDING ENERGY OF THE LAST NEUTRON',T38,1PE12.6,3X,    1690
     *     'EV'/)                                                     1700
      BK(1)=WL                                                          1710
      BK(2)=AP                                                          1720
      BK(3)=EB                                                          1730
      N=3                                                               1740
      CALL WDA2(MATK  ,MMIS,4HISOT,4H2   )                              1750
      AWRI =AWRIS                                                       1760
  103 CONTINUE                                                          1770
C                                                                       1780
C-----ISOT3                                                             1790
C                                                                       1800
      IF(NIS.EQ.1) GOTO 402                                             1810
      CALL XTEXT(4HISOT,NAM1(1))                                        1820
      CALL XTEXT(4H3   ,NAM1(2))                                        1830
      CALL A8FORM(NO,NAM8)                                              1840
      WRITE(NO,206)                                                     1850
  206 FORMAT(1HO,'ISOT3'/1X,5('*')/)                                    1860
      WRITE(NO,107)(AWRISO(MIS),ABIS(MIS),MIS=1,NIS)                    1870
  107 FORMAT(1X,'ATOMIC WEIGHTS',T25,'ISOTOPIC ABUNDANCE'//             1880
     * (3X,1PE12.6,T26,1PE12.6))                                       1890
      I=1                                                               1900
      DO 401 MIS=1,NIS                                                  1910
      BK(I)=AWRISO(MIS)                                                 1920
      BK(I+1)=ABIS(MIS)*100.0                                           1930
      I=I+2                                                             1940
      IF(I.GT.N8KX) CALL ERROR(204)                                     1950
  401 CONTINUE                                                          1960
      N=I-1                                                             1970
      CALL WDA2(MATKDK,1,4HISOT,4H3   )                                 1980
      GOTO 400                                                          1990
  402 IF((MATKDK/1000)*1000.NE.MATKDK) GOTO 400                         2000
      AWR=AWRISO(1)+0.00001                                             2010
      WRITE(NO,109) NAMKDK,ABIS(1),LNAM(1),LNAM(2),AWR,(LNAM(I),I=6,8)  2020
  109 FORMAT('0THE MATERIAL ',A8,' CONSISTS OF ',2PE7.2,'%  ',2A1,      2030
     *0PI3,3A1,'.')                                                     2040
  400 RETURN                                                            2050
      END                                                               2060
```

```
      SUBROUTINE MARY(EL,EH,LRF,ZAI,ABN,LFWRES,NIS,LNER,MIS,LIS,SPI,AWRI   10
     *,AP,BK,NBT,JNT,ENS,SIGS,B,AH,JXRED,ER,AJ,GT,GN,GG,GF,AL,G,SER,PER,   20
     *NDIM)                                                               30
      COMMON/INOUT/NO,NI,NT,MODE,LEDIT1,LEDIT2,NREF1(78),                 40
     *NAMISO(10),NREF2(22),N1X,N2X,N,NBKX,NREF4(21),MLSLSW                50
      COMMON/RECS/NREF3(3),C1,C2,L1,L2,N1,N2                              60
      REAL*8 NAMISO                                                       70
      DIMENSION ER(1),AJ(1),GT(1),GN(1),GG(1),GF(1),AL(1),                80
     *          G(1),BK(1),NBT(1),JNT(1),B(1),SER(1),PER(1),              90
     *SIGS(1),ENS(1),LLSTR(5),AH(1)                                      100
      PI=3.14159265                                                      110
      PIM4=4.0*PI                                                        120
      CON1=2.196771E-3                                                   130
      CON2=PIM4/(CON1**2)                                                140
      CALL RREC(1,NT,MODE,-1.0)                                          150
      SPI=C1                                                             160
      AP=C2                                                              170
C-----LIS=0 CALCULATE CROSS SECTION FROM RESONANCE PARAMETERS            180
C           PLUS SMOOTH CONTRIBUTION FROM FILE 3                         190
C        =1 USE SMOOTH CROSS SECTION IN FILE 3 ONLY                      200
      LIS=L1                                                             210
      NLS=N1                                                             220
      WRITE(NO,1)                                                        230
    1 FORMAT(1H1,20X,'RESOLVED RESONANCE DATA')                          240
      IF(NIS.GT.1)WRITE(NO,360)NAMISO(MIS)                               250
  360 FORMAT('+',60X,'ISOTOPE',3X,A8)                                    260
      CALL PRIZEI                                                        270
      IZ=0.001*ZAI                                                       280
      IA=ZAI-1000.*IZ                                                    290
      WRITE(NO,2)IZ,IA,ABN,EL,EH,SPI,AP,NLS                              300
    2 FORMAT(1HO,'Z---------------------------',I11/                     310
     *       1X,'A---------------------------',I11/                     320
     *       1X,'FRACTIONNAL ABUNDANCE--------',1PE11.4/                330
```

```
     *          1X,'LOWER ENERGY LIMIT-----------',E11.4/                  340
     *          1X,'UPPER ENERGY LIMIT-----------',E11.4/                  350
     *          1X,'NUCLEAR SPIN-----------------',E11.4/                  360
     *          1X,'SPIN SCATTERING LENGTH (A+)--',E11.4/                  370
     *          1X,'NUMBER OF L STATES-----------',I11/)                   380
      WRITE(NO,1000)                                                       390
 1000 FORMAT(' FORMALISM RECOMMENDED BY ENDF/B TO CALCLLATE THE RESONANC   400
     *ES:')                                                                410
      IF(LRF.EQ.1) WRITE(NO,1001)                                          420
      IF(LRF.EQ.2) WRITE(NO,1002)                                          430
 1001 FORMAT('+',60X,' SINGLE LEVEL BREIT WIGNER.')                        440
 1002 FORMAT('+',60X,' MULTI LEVEL BREIT WIGNER.')                         450
      WRITE(NO,1003)                                                       460
 1003 FORMAT(' FORMALISM USED IN CALCULATING THE RESONANCES:')             470
      IF(MLSLSW.GE.1) LRF=1                                                480
      IF(LRF.EQ.1) WRITE(NO,1001)                                          490
      IF(LRF.EQ.2) WRITE(NO,1002)                                          500
      IF(NLS.GT.5) CALL ERROR(207)                                         510
      NRSI=1                                                               520
      DO 3 I=1,NLS                                                         530
      J=1                                                                  540
      CALL RREC(2,NT,MODE,-1.0,B)                                          550
      AWRI=C1                                                              560
      C=2.196771E-3*AP*AWRI/(AWRI+1.0)                                     570
      AM=C2                                                                580
      ARAT=AWRI/(AWRI+1.0)                                                 590
      RAD=0.1*(0.8+1.23*AWRI**0.333333333)                                 600
      CON3=CON1*ARAT*RAD                                                   610
C     RHO=CON3*SQRT(E)                                                     620
      CON4=CON2/(ARAT**2)                                                  630
C     4PI/K**2 = CON4/E                                                    640
C     RHOCF = C * SQRT(E)                                                  650
      L=L1                                                                 660
      LLSTR(I)=L                                                           670
      NRS=N2                                                               680
      NRSF=NRSI+N2-1                                                       690
      WRITE(NO,5)L,NRS,AM                                                  700
    5 FORMAT(1HO,'L VALUE----------------------',I11/                      710
     *          1X,'NUMBER OF RESONANCES---------',I11/                   720
     *          1X,'SPIN SCATTERING LENGTH-------',1PE11.4//)             730
      IF(LEDIT2.EQ.1)WRITE(NO,120)                                         740
  120 FORMAT(   44X,'RESONANCE WIDTHS (EV)'/                               750
     *          6X,'ENERGY (EV)',2X,'J VALUE',8X,                         760
     *          'TOTAL',10X,'NEUTRON',8X,'RADIATION',6X,                  770
     *          'FISSION',8X,'L VALUE',8X,'G VALUE')                      780
      DO 4 K=NRSI,NRSF                                                     790
      IF(K.GT.NDIM) CALL ERROR(205)                                        800
      ER(K)=B(J)                                                           810
      AJ(K)=B(J+1)                                                         820
      GT(K)=B(J+2)                                                         830
      GN(K)=B(J+3)                                                         840
      GG(K)=B(J+4)                                                         850
      GF(K)=B(J+5)                                                         860
      AL(K)=L                                                              870
      G(K)=(2.*AJ(K)+1.)/(2.*(2.*SPI+1.))                                  880
C     CALCULATE PENETRATION AND SHIFT FACTOR AT RESONANCE ENERGY ER(K).    890
      RHO=CON3*SQRT(ABS(ER(K)))                                            900
      CALL FACTS(L,RHO,SER(K),PER(K))                                      910
      J=J+6                                                                920
      IF(LEDIT2.EQ.1)                                                      930
     *      WRITE(NO,6)ER(K),AJ(K),GT(K),GN(K),GG(K),GF(K),AL(K),G(K)      940
    6 FORMAT(1P8E15.4)                                                     950
    4 CONTINUE                                                             960
      NRSI=NRSF+1                                                          970
    3 CONTINUE                                                             980
C-----NUMBER OF RESONANCES                                                 990
      NRES=NRSF                                                           1000
C-----SORT RESONANCE DATA ON INCREASING ENERGY                            1010
      IF(NLS.GT.1) CALL SORTRE(NRES,ER,AJ,GT,GN,GG,GF,AL,G,SER,PER)       1020
C-----STORE RESONANCE DATA IN THE ARRAY BK.                               1030
      CALL MARY01(EL,EH,NRES,BK,ER,AL,AJ,G,GT,GN,GG,GF,NIS,MIS,LIS,       1040
     *LFWRES,49)                                                          1050
C=====CALCULATE ENERGY POINTS USING NPTR POINTS PER RESONANCE             1060
  140 IF(LNER.LE.0)NPTR=51                                                1070
      IF(LNER.GT.1)NPTR=LNER                                              1080
      NPTR=MIN0(NPTR,200)                                                 1090
      EUP=EH                                                              1100
      CALL EPT(EL,EUP,NPTR,NSE,NRES,ER,GT,ENS,N2X)                        1110
C     CHECK IF THERE ARE POINTS WITH THE SAME VALUE.                      1120
C     IF YES, PUT ONLY ONE OF THIS POINTS.                                1130
      CALL MARY02(ENS,NSE)                                                1140
      WRITE(NO,300)NSE,(ENS(I),I=1,NSE)                                   1150
  300 FORMAT(1H1,I6,3X,'ENERGIES SELECTED BY EPT'//(1P8E15.5))            1160
      CALL PRIZEI                                                         1170
C     CALCULATE SGN, SGG AND IF LFWRES=1 SGF.                             1180
      CALL MARCEL(LFWRES,LRF,C,CON3,CON4,PIM4,AP,NRES,NLS,                1190
     *LLSTR,ER,GN,GG,GF,G,AL,SER,PER)                                     1200
      IF((LFWRES.EQ.1.AND.3*NSE.GT.JXRED).OR.                             1210
     *   (LFWRES.NE.1.AND.2*NSE.GT.JXRED)) CALL ERROR(208)                1220
      DO 40 K=1,NSE                                                       1230
      CALL MARGOT(ENS(K),AH(K),AH(NSE+K),SGF)                             1240
      IF(LFWRES.EQ.1) AH(2*NSE+K)=SGF                                     1250
   40 CONTINUE                                                            1260
C     CHECK IF THERE ARE ENOUGH ENERGY POINTS FOR DESCRIBING              1270
C     POINT RESONANCE CROSS SECTIONS. CALCULATE NBT AND JNT.              1280
      IF((6*N1X).GT.NBKX) CALL ERROR(206)                                 1290
      CALL MILLI(ENS,NSE,AH,BK(1),BK(1+N1X),BK(1+2*N1X),BK(1+3*N1X),      1300
     *BK(1+4*N1X),BK(1+5*N1X),LFWRES,N1SGN,N1SGG,N1SGF,JXRED)             1310
      CALL MELA(LFWRES,BK,AH,NBT,JNT,SIGS,N1SGN,N1SGG,N1SGF,NSE,MIS,      1320
     *EL,EH)                                                              1330
    9 RETURN                                                              1340
      END                                                                 1350
```

```
      SUBROUTINE SORTRE(NRES,ER,AJ,GT,GN,GG,GF,AL,G,SER,PER)                10
C     SORTIERT RESONANZDATEN NACH AUFSTEIGENDEN ENERGIEN.                   20
      DIMENSION ER(1),AJ(1),GT(1),GN(1),GG(1),GF(1),AL(1),G(1),             30
     *SER(1),PER(1)                                                         40
      NR1=NRES-1                                                            50
      DO 8 IR=1,NR1                                                         60
      IX=IR+1                                                               70
      DO 8 I=IX,NRES                                                        80
      IF(ER(IR).LT.ER(I)) GOTO 8                                            90
      T1=ER(IR)                                                            100
      T2=AJ(IR)                                                            110
      T3=GT(IR)                                                            120
      T4=GN(IR)                                                            130
      T5=GG(IR)                                                            140
      T6=GF(IR)                                                            150
      T7=AL(IR)                                                            160
      T8=G(IR)                                                             170
      T9=SER(IR)                                                           180
      T10=PER(IR)                                                          190
      ER(IR)=ER(I)                                                         200
      AJ(IR)=AJ(I)                                                         210
      GT(IR)=GT(I)                                                         220
      GN(IR)=GN(I)                                                         230
      GG(IR)=GG(I)                                                         240
      GF(IR)=GF(I)                                                         250
      AL(IR)=AL(I)                                                         260
      G(IR)=G(I)                                                           270
      SER(IR)=SER(I)                                                       280
      PER(IR)=PER(I)                                                       290
      ER(I)=T1                                                             300
      AJ(I)=T2                                                             310
      GT(I)=T3                                                             320
      GN(I)=T4                                                             330
      GG(I)=T5                                                             340
      GF(I)=T6                                                             350
      AL(I)=T7                                                             360
      G(I)=T8                                                              370
      SER(I)=T9                                                            380
      PER(I)=T10                                                           390
    8 CONTINUE                                                             400
      RETURN                                                               410
      END                                                                  420
```

```
      SUBROUTINE MARCEL(LFWRES,LRF,C,CON3,CON4,PIM4,AP,NRES,NLS,            10
     *LLSTR,ER,GN,GG,GF,G,AL,SER,PER)                                       20
C     CALCULATES ELASTIC, RADIATIVE CAPTURE AND (IF LFWRES=1) FISSION       30
C     CROSS SECTIONS. DEPENDING ON LRF SINGLE LEVEL OR MULTI LEVEL          40
C     BREIT WIGNER IS USED.                                                 50
      REAL*8 SGN8,SGG8,SGF8,SML8                                            60
      DIMENSION LLSTR(5),SINP2(5),SIN2P(5),COS2P(5),AL(1),ER(1),            70
     *GN(1),GG(1),GF(1),G(1),SER(1),PER(1)                                  80
      GOTO 333                                                              90
      ENTRY MARGOT(E,SGN,SGG,SGF)                                          100
      SGN8=0.0                                                             110
      SGG8=0.0                                                             120
      SGF8=0.0                                                             130
      RHOCF=C*SQRT(E)                                                      140
      DO 50 LL=1,NLS                                                       150
      LLS=LLSTR(LL)                                                        160
      CALL FACPHI(LLS,RHOCF,PHIL)                                          170
      LLSI=LLS+1                                                           180
      SINP2(LLSI)=SIN(PHIL)**2                                             190
      SIN2P(LLSI)=SIN(2.0*PHIL)                                            200
      COS2P(LLSI)=COS(2.0*PHIL)                                            210
      SPOT=0.0                                                             220
      IF(ABS(PHIL).GT.5.0E-4.OR.LLS.GT.0)                                  230
     *SPOT=CON4*(2.0*FLOAT(LLS)+1.0)*SINP2(LLSI)/E                         240
      IF(ABS(PHIL).LE.5.0E-4.AND.LLS.EQ.0)                                 250
     *SPOT=PIM4*AP**2                                                      260
      SGN8=SGN8+SPOT                                                       270
   50 CONTINUE                                                             280
      CON5=CON4/E                                                          290
      RHOE=CON3*SQRT(E)                                                    300
      DO 54 J=1,NRES                                                       310
      LLS=AL(J)+0.001                                                      320
      LLSI=LLS+1                                                           330
      IF(LLS) 55,56,57                                                     340
   55 CALL ERROR(209)                                                      350
   56 ERE=ER(J)                                                            360
      GN1=GN(J)*SQRT(E/ABS(ER(J)))                                         370
      GOTO 22                                                              380
   57 CALL FACTS(LLS,RHOE,SLE,PLE)                                         390
      ERE=ER(J)+(SER(J)-SLE)*GN(J)/(2.0*PER(J))                            400
      GN1=GN(J)*PLE/PER(J)                                                 410
   22 GX=GF(J)+GG(J)                                                       420
      GT1=GN1+GX                                                           430
      XZ=2.0*(E-ERE)/GT1                                                   440
      SIGO=CON5*G(J)                                                       450
      GN1GT1=GN1/GT1                                                       460
      COMFAC=SIGO*GN1GT1/(1.0+XZ**2)                                       470
      SIG1=COMFAC*(GN1GT1*COS2P(LLSI)-2.0*SINP2(LLSI)*GX/GT1               480
     #+XZ*SIN2P(LLSI))                                                     490
      SGN8=SGN8+SIG1                                                       500
      SIGMA=COMFAC/GT1                                                     510
      SGG8=SGG8+SIGMA*GG(J)                                                520
      IF(LFWRES.EQ.1) SGF8=SGF8+SIGMA*GF(J)                                530
      IF(LRF.NE.2) GOTO 54                                                 540
C     MULTILEVEL INTERFERENCE                                              550
      IF(J.EQ.1) GOTO 54                                                   560
      JJ=J-1                                                               570
      SML8=0.0                                                             580
      DO 58 JS=1,JJ                                                        590
C     CHECK L-VALUE AND G-VALUE TO CALCULATE INTERFERENCE                  600
C     ONLY FOR SAME SPIN STATES.                                           610
      IF(G(JS).NE.G(J).OR.AL(JS).NE.AL(J)) GOTO 58                         620
      IF(LLS) 59,60,61                                                     630
   59 CALL ERROR(209)                                                      640
```

```
   60 ERES=ER(JS)                                                        650
      GNS=GN(JS)*SQRT(E/ABS(ER(JS)))                                     660
      GOTO 23                                                            670
   61 ERES=ER(JS)+(SER(JS)-SLE)*GN(JS)/(2.0*PER(JS))                     680
      GNS=GN(JS)*PLE/PER(JS)                                             690
   23 GTS=GNS+GF(JS)+GG(JS)                                              700
      XS=2.0*(E-ERES)/GTS                                                710
      SML8=SML8+GNS/GTS*(1.0+XZ*XS)/(1.0+XS**2)                          720
   58 CONTINUE                                                           730
      SGN8=SGN8+2.0*SML8*COMFAC                                          740
   54 CONTINUE                                                           750
      SGN=SGN8                                                           760
      SGG=SGG8                                                           770
      SGF=SGF8                                                           780
  333 RETURN                                                             790
      END                                                                800
```

```
      SUBROUTINE MILLI(ENS,NSE,AH,NBTN,JNTN,NBTG,JNTG,NBTF,JNTF,          10
     *LFWRES,N1SGN,N1SGG,N1SGF,JXRED)                                     20
C     CHECKS IF THERE ARE ENUOGH POINTS DESCRIBING POINT RESONANCE        30
C     CROSS SECTIONS. CALCULATES NBT ANT JNT.                             40
C     NSE MAY HAVE BEEN ENLARGED AFTER RETURN. IN NBT AND JNT THERE ARE   50
C     AFTER RETURN N1 VALUES FOR INTERPOLATING THE POINT CROSS SECTION.   60
C     N MEANS SGN, G MEANS SGG AND F MEANS SGF.                           70
C     IN AH YOU WILL FIND THE FOLLOWING DATA.                             80
C           1 ... NSE   SGN                                               90
C       NSE+1 ... 2*NSE   SGG                                            100
C     2*NSE+1 ... 3*NSE   SGF                                            110
C                                                                        120
      COMMON/INOUT/ IOUT,NR1(126),N2X,NR2(12),EPS                        130
      DIMENSION ENS(1),AH(1),NBTN(1),JNTN(1),NBTG(1),JNTG(1),            140
     *NBTF(1),JNTF(1),EPSI(5),NIC(5),CST(3)                              150
      DATA CST/'SGN ','SGG ','SGF '/                                     160
      LOGICAL*4 T                                                        170
      T=.FALSE.                                                          180
      IF(LFWRES.EQ.1) T=.TRUE.                                           190
      IEND=2                                                             200
      IF(T) IEND=3                                                       210
      N1SGN=1                                                            220
      N1SGG=1                                                            230
      N1SGF=1                                                            240
      NBTN(1)=2                                                          250
      JNTN(1)=2                                                          260
      NBTG(1)=2                                                          270
      JNTG(1)=2                                                          280
      IF(.NOT.T) GOTO 17                                                 290
      NBTF(1)=2                                                          300
      JNTF(1)=2                                                          310
C     LOOP FOR ALL ENERGY-INTERVALS.                                     320
   17 K=2                                                                330
    1 E1=ENS(K-1)                                                        340
      E2=ENS(K)                                                          350
      E=(E2+E1)*0.5                                                      360
      IF(E.LT.E1) E=0.5*E2+0.5*E1                                        370
      CALL MARGOT(E,SGN,SGG,SGF)                                         380
C     CALCULATE FOR SGN,SGG AND SGF INTERPOLATIONCODE AND EPSI           390
      EPSSGF=0.0                                                         400
      DO 2 I=1,IEND                                                      410
      GOTO (3,4,5),I                                                     420
    3 IP=0                                                               430
      SIGMA=SGN                                                          440
      ICALT=JNTN(N1SGN)                                                  450
      GOTO 6                                                             460
    4 IP=NSE                                                             470
      SIGMA=SGG                                                          480
      ICALT=JNTG(N1SGG)                                                  490
      GOTO 6                                                             500
    5 IP=2*NSE                                                           510
      SIGMA=SGF                                                          520
      ICALT=JNTF(N1SGF)                                                  530
    6 S1=AH(IP+K-1)                                                      540
      S2=AH(IP+K)                                                        550
      DO 7 IC=2,5                                                        560
      IF((IC.GE.4.AND.(S1.LE.0.0.OR.S2.LE.0.0)).OR.                      570
     *((IC.EQ.3.OR.IC.EQ.5).AND.((E2/E1.EQ.1.0).OR.                      580
     *(E.LT.E1.OR.E2.LT.E1)))) GOTO 23                                   590
      CALL TERP1(E1,S1,E2,S2,E,YINT,IC)                                  600
      EPSI(IC)=ABS((YINT-SIGMA)/SIGMA)                                   610
      GOTO 7                                                             620
   23 EPSI(IC)=1.0E+50                                                   630
    7 CONTINUE                                                           640
C     INCREASING ORDER OF EPSI IS MADE.                                  650
      DO 18 IC=2,5                                                       660
   18 NIC(IC)=IC                                                         670
      DO 19 IC=2,4                                                       680
      ICP1=IC+1                                                          690
      DO 20 ICI=ICP1,5                                                   700
      IF(EPSI(IC).LE.EPSI(ICI)) GOTO 20                                  710
      IP=NIC(IC)                                                         720
      NIC(IC)=NIC(ICI)                                                   730
      NIC(ICI)=IP                                                        740
      SIGMA=EPSI(IC)                                                     750
      EPSI(IC)=EPSI(ICI)                                                 760
      EPSI(ICI)=SIGMA                                                    770
   20 CONTINUE                                                           780
   19 CONTINUE                                                           790
      IF(EPSI(2).GT.EPS.AND.E.NE.E1) GOTO 11                             800
      IF(EPSI(2).GT.EPS.AND.E.EQ.E1) WRITE(IOUT,1000) EPS,E1,E2,CST(I),  810
     *EPSI(2)                                                            820
 1000 FORMAT('0***** WARNING IN SUBROUTINE MILLI. WISHED ACCURACY=',     830
     *F5.4,' FOR INTERPOLATING POINTS'/                                  840
     *' ***** IN THE ENERGY INTERVAL E1=',1PE15.7,' E2=',1PE15.7,        850
     *' COULD NOT BE FULFILLED FOR'/                                     860
     *' ***** CROSS-SECTION TYPE ',A4,'. BEST ACCURACY FOR THIS TYPE IN  870
     * THIS INTERVAL IS=',1PE15.7,'.')                                   880
      III=2                                                              890
      IF(K.EQ.2) GOTO 21                                                 900
```

```
      DO 22 II=2,5                                                          910
      IF(NIC(II).NE.ICALT) GOTO 22                                          920
      IF(II.EQ.2) GOTO 21                                                   930
      IF(EPSI(II).LE.EPS.AND.EPSI(II).LE.(6*EPSI(2)/II)) III=II             940
      GOTO 21                                                               950
   22 CONTINUE                                                              960
   21 ICODE=NIC(III)                                                        970
      GOTO (8,9,10),I                                                       980
    8 ICSGN=ICODE                                                           990
      GOTO 2                                                               1000
    9 ICSGG=ICODE                                                          1010
      GOTO 2                                                               1020
   10 ICSGF=ICODE                                                          1030
    2 CONTINUE                                                             1040
C     EPSSGN,EPSSGG AND EPSSGF ARE LESS EPS. STORE INTERPOLATIONCODE       1050
C     SET K=K+1 AND BRANCH TO 1 IF(K.LE.NSE).                              1060
C     FIRST INTERVAL?                                                      1070
      IF(K.GT.2) GOTO 12                                                   1080
      JNTN(1)=ICSGN                                                        1090
      JNTG(1)=ICSGG                                                        1100
      IF(.NOT.T) JNTF(1)=ICSGF                                             1110
      GOTO 13                                                              1120
   12 CONTINUE                                                             1130
      CALL MILLII(NBTN,JNTN,N1SGN,ICSGN,K,NSE)                             1140
      CALL MILLII(NBTG,JNTG,N1SGG,ICSGG,K,NSE)                             1150
      IF(T) CALL MILLII(NBTF,JNTF,N1SGF,ICSGF,K,NSE)                       1160
   13 K=K+1                                                                1170
      IF(K.LE.NSE) GOTO 1                                                  1180
      GOTO 333                                                             1190
   11 CONTINUE                                                             1200
C     ENLARGE ENS BY POINT E. ENLARGE AH BY SIGMAS.                        1210
      NSE=NSE+1                                                            1220
      IF(NSE.GT.N2X) CALL RESERR(105,K,NSE,1)                              1230
      IF((T.AND.(3*NSE).GT.JXRED).OR.                                      1240
     *   (.NOT.T.AND.(2*NSE).GT.JXRED)) CALL ERROR(208)                    1250
C     MAKING A GAP.                                                        1260
      KK=IEND*NSE                                                          1270
      IVER=IEND                                                            1280
      IM=IEND-1                                                            1290
   14 I1=KK-IVER                                                           1300
      AH(KK)=AH(I1)                                                        1310
      IF(KK.LE.NSE) ENS(KK)=ENS(I1)                                        1320
   15 KK=KK-1                                                              1330
      IF(KK.NE.(IM*NSE+K)) GOTO 14                                         1340
      IF(IM.LE.0) GOTO 16                                                  1350
      IVER=IVER-1                                                          1360
      IM=IM-1                                                              1370
      IF(IVER.LT.0) CALL ERROR(777)                                        1380
      GOTO 15                                                              1390
   16 ENS(K)=E                                                             1400
      AH(K)=SGN                                                            1410
      AH(NSE+K)=SGG                                                        1420
      IF(T) AH(2*NSE+K)=SGF                                                1430
      GOTO 1                                                               1440
  333 RETURN                                                               1450
      END                                                                  1460
```

```
      SUBROUTINE MILLII(NBT,JNT,N1,IC,K,NSE)                                10
      COMMON/INOUT/ NR1(126),N1X                                            20
      DIMENSION JNT(1),NBT(1)                                               30
      IF(JNT(N1).EQ.IC) GOTO 1                                              40
      N1=N1+1                                                               50
      IF(N1.GT.N1X) CALL RESERR(103,K,NSE,1)                                60
      JNT(N1)=IC                                                            70
    1 NBT(N1)=K                                                             80
  333 RETURN                                                                90
      END                                                                  100
```

```
      SUBROUTINE RESERR(N,K,NSE,IERR)                                       10
      COMMON/INOUT/ IO                                                      20
      WRITE(IO,100)                                                         30
  100 FORMAT(' ***** ERROR IN RESOLVED RESONANCE CALCULATING SECTION.')     40
      IF(IERR.EQ.1) WRITE(IO,102) K,NSE                                     50
  102 FORMAT(' ***** ',I5,' POINTS OUT OF ',I5,' PROCESSED.')               60
      IF(IERR.EQ.2) WRITE(IO,103) K,NSE                                     70
  103 FORMAT(' ***** ',I5,' RESONANCES OUT OF ',I5,' PROCESSED.')           80
      I=0                                                                   90
      IF(N.EQ.103) I=1                                                     100
      IF(N.EQ.105) I=2                                                     110
      WRITE(IO,101) I                                                      120
  101 FORMAT(' ***** INCREASE N',I1,'X OR TOTAL REGION.')                  130
      STOP 2                                                               140
      END                                                                  150
```

```
      SUBROUTINE MELA(LFWRES,IBK,AH,NBT,JNT,SIGS,N1SGN,N1SGG,N1SGF,NSE,     10
     *MIS,EL,EH)                                                            20
      COMMON/INOUT/ NO,NR1(5),LEDITS,NR2(119),N1X,N2X                       30
      COMMON/RECS/ NR3(3),C1,NR4(3),N1,N2                                   40
      DIMENSION IBK(1),AH(1),NBT(1),JNT(1),SIGS(1),TK(3)                    50
      REAL*8 T(6)                                                           60
      DATA T/'SCATTERI','NG      ','CAPTURE ','        ',                   70
     *'FISSION ','        '/,TK/'SCAT','CAPT','FISS'/                      80
      K3=2                                                                  90
      IF(LFWRES.EQ.1) K3=3                                                 100
      C1=0.0                                                               110
      DO 77 K2=1,K3                                                        120
      GOTO (70,71,72),K2                                                   130
   70 N1=N1SGN                                                             140
      IP=0                                                                 150
      GOTO 73                                                              160
   71 N1=N1SGG                                                             170
      IP=2*N1X                                                             180
```

```
      GOTO 73                                                          190
   72 N1=N1SGF                                                         200
      IP=4*N1X                                                         210
   73 N2=NSE                                                           220
      DO 74 I=1,N1                                                     230
      NBT(I)=IBK(IP+I)                                                 240
   74 JNT(I)=IBK(IP+N1X+I)                                             250
      IF(N2.GT.N2X) CALL ERROR(105)                                    260
      DO 75 I=1,N2                                                     270
   75 SIGS(I)=AH((K2-1)*NSE+I)                                         280
      IF(LEDITS.NE.1) GOTO 101                                         290
      WRITE(NO,250) T(2*K2-1),T(2*K2)                                  300
  250 FORMAT('1POINT WISE ',A8,A3,'CROSS SECTION CALCULATED FROM RESOLVE 310
     *D RESONANCE PARAMETER'//)                                        320
      CALL WREC(3,NO,4)                                                330
  101 CONTINUE                                                         340
      CALL WRECS2(MIS,TK(K2),4H RES,EL,EH)                             350
   77 CONTINUE                                                         360
      RETURN                                                           370
      END                                                              380
```

```
      SUBROUTINE MARY01(EL,EH,NRES,BK,ER,AL,AJ,G,GT,GN,GG,GF,NIS,MIS,    1C
     *LIS,LFWRES,*)                                                      20
      DIMENSION NAM1(2),BK(1),ER(1),AL(1),AJ(1),G(1),GT(1),GN(1),GG(1),   3C
     *GF(1)                                                              4C
      REAL*8 NAMISO,NAM8                                                 50
      COMMON/INOUT/NO,NR1(4),LEDIT2,NR2(76),MATKDK,NR3(1),NAMISO(10),    6C
     *MATISO(10),NR4(14),N,NBKX                                          70
      EQUIVALENCE (NAM8,NAM1(1))                                         80
C                                                                        90
      J=1                                                               100
      DO 17 K=1,NRES                                                    110
      IF(J.GT.NBKX) CALL ERROR(206)                                     120
      BK(J)=ER(K)                                                       130
      BK(J+1)=AL(K)                                                     140
      BK(J+2)=AJ(K)                                                     150
      BK(J+3)=G(K)                                                      160
      BK(J+4)=GT(K)                                                     170
      BK(J+5)=GN(K)                                                     180
      IF(ER(K).LT.0.0)BK(J+5)=GN(K)/SQRT(ABS(ER(K)))                    190
      BK(J+6)=GG(K)                                                     200
      BK(J+7)=GF(K)                                                     210
      BK(J+8)=0.0                                                       220
      BK(J+9)=0.0                                                       230
      BK(J+10)=0.0                                                      240
      J=J+11                                                            250
   17 CONTINUE                                                          260
      N=J-1                                                             270
      IF(LEDIT2.NE.1)GOTO 100                                           280
      WRITE(NO,21)                                                      290
   21 FORMAT(1H1,'RESONANCE DATA IN KEDAK FORMAT')                      300
      IF(NIS.GT.1)WRITE(NO,360)NAMISO(MIS)                              310
```

```
  360 FORMAT(1H+,60X,'ISOTOPE',3X,A8)                                  320
      CALL XTEXT(4HRES ,NAM1(1))                                       330
      CALL XTEXT(4H    ,NAM1(2))                                       340
      CALL A8FORM(NC,NAM8)                                             350
      CALL LUCY(2,BK)                                                  360
  100 CONTINUE                                                         370
      WRITE(NO,350)                                                    380
  350 FORMAT(1H1,'KEDAK REACTION TYPE RANGRES')                        390
      IF(NIS.GT.1)WRITE(NO,360)NAMISO(MIS)                             400
      CALL XTEXT(4HRANG,NAM1(1))                                       410
      CALL XTEXT(4HRES ,NAM1(2))                                       420
      CALL A8FORM(NC,NAM8)                                             430
      MMIS=1                                                           440
      MATK=MATKDK                                                      450
      IF(NIS.EQ.1)GOTO 301                                             460
      MATK=MATISO(MIS)                                                 470
      MMIS=MIS+1                                                       480
  301 CALL WDA2(MATK  ,MMIS,4HRES ,4H    )                             490
C=====RANGRES                                                          500
      BK(1)=EL                                                         510
      BK(2)=EH                                                         520
      BK(3)=NRES                                                       530
      BK(4)=1.0                                                        540
      N=4                                                              550
      WRITE(NO,351)(BK(K),K=1,4)                                       560
  351 FORMAT(' LOWER ENERGY BOUNDARY OF THE RESOLVED RESONANCE REGION', 570
     *T58,1PE15.6,' EV'/                                               580
     *       ' UPPER ENERGY BOUNDARY OF THE RESOLVED RESONANCE REGION', 590
     *T58,1PE15.6,' EV'/                                               600
     *       1X,'NUMBER OF RESONANCES',T58,0PF15.0/                    610
     *       1X,'CONSISTENCY TEST PARAMETER',T58,F15.0)                620
      CALL WDA2(MATK,MMIS,4HRANG,4HRES )                               630
      IF(LIS.NE.0)RETURN 1                                             640
C=====DETERMINE IF THE FISSION WIDTHS ARE ZERO                         650
   10 LFWRES=1                                                         660
      DO 130 IR=1,NRES                                                 670
      IF(GF(IR).GT.1.E-6)GOTO 333                                      680
  130 CONTINUE                                                         690
      LFWRES=0                                                         700
      WRITE(NO,150)                                                    710
  150 FORMAT(1H0,'NO FISSION WIDTHS FOR THIS ISOTOPE')                 720
  333 RETURN                                                           730
      END                                                              740
```

```
      SUBROUTINE MARY02(ENS,NSE)                                        10
C     CHECKS IF THERE ARE POINTS WITH THE SAME VALUE.                   20
C     IF YES, ALL BUT ONE OF THESE VALUES ARE DELETED.                  30
C                                                                       40
      DIMENSION ENS(1)                                                  50
      I=2                                                               60
    1 IF(ENS(I).EQ.ENS(I-1)) GOTO 2                                     70
      I=I+1                                                             80
```

```
    5 IF(I.LE.NSE) GOTO 1                                                 90
  333 RETURN                                                              100
    2 IA=I+1                                                              110
      IF(IA.GT.NSE) GOTO 4                                                120
      DO 3 II=IA,NSE                                                      130
    3 ENS(II-1)=ENS(II)                                                   140
    4 NSE=NSE-1                                                           150
      GOTO 5                                                              160
      END                                                                 170
```

```
      SUBROUTINE FACTS(L,RHO,SE,PE)                                       10
C*********************************************************************   20
C*    CALCULATES PENETRATION AND SHIFT FACTORS                       *    30
C*    LIBERATED FROM GREGSON ET. AL.                                 *    40
C*********************************************************************   50
      LL=L+1                                                              60
      GO TO(10,20,30,40,50),LL                                            70
C-----S-WAVE                                                              80
   10 SE=0.0                                                              90
      PE=RHO                                                             100
      GO TO 100                                                          110
C-----P-WAVE                                                             120
   20 R2=RHO*RHO                                                         130
      DEN=1.0+R2                                                         140
      PE=R2*RHO/DEN                                                      150
      SE=-1.0/DEN                                                        160
      GO TO 100                                                          170
C-----D-WAVE (L=2)                                                       180
   30 R2=RHO*RHO                                                         190
      R4=R2*R2                                                           200
      DEN=3.0*R2+R4+9.0                                                  210
      PE=R4*RHO/DEN                                                      220
      SE=-(18.0+3.0*R2 )/DEN                                             230
      GO TO 100                                                          240
C-----F-WAVE (L=3)                                                       250
   40 R2=RHO*RHO                                                         260
      R4=R2*R2                                                           270
      R6=R4*R2                                                           280
      DEN=225.0+45.0*R2+6.0*R4+R6                                        290
      PE=R6*RHO/DEN                                                      300
      SE=-(675.0+90.0*R2+6.0*R4)/DEN                                     310
      GO TO 100                                                          320
C-----G-WAVE (L=4)                                                       330
   50 R2=RHO*RHO                                                         340
      R4=R2*R2                                                           350
      R6=R4*R2                                                           360
      R8=R4*R4                                                           370
      DEN=11025.0+1575.0*R2+135.0*R4+10.0*R6+R8                          380
      PE=R8*RHO/DEN                                                      390
      SE=-(44100.0+4725.0*R2+270.0*R4+10.0*R6)/DEN                       400
  100 RETURN                                                             410
      END                                                                420
```

```
      SUBROUTINE FACPHI(L,RHO,PHI)                                        10
C*********************************************************************   20
C*    CALCULATES PHASE SHIFT                                         *    30
C*********************************************************************   40
      LL=L+1                                                              50
      GO TO (10,20,30,40,50),LL                                           60
C-----S-WAVE                                                              70
   10 PHI=RHO                                                             80
      GO TO 100                                                           90
   20 PHI=RHO-ATAN(RHO)                                                  100
      GO TO 100                                                          110
   30 R2=RHO*RHO                                                         120
      PHI=RHO-ATAN(3.0*RHO/(3.0-R2))                                     130
   32 IF((PHI/RHO).LT.0.000001) PHI=0.0                                  140
      GO TO 100                                                          150
   40 R2=RHO*RHO                                                         160
      PHI=RHO-ATAN((15.0*RHO-RHO*R2)/(15.0-6.0*R2))                      170
      GO TO 32                                                           180
   50 R2=RHO*RHO                                                         190
      R4=R2*R2                                                           200
      TOP=105.0*RHO-10.0*R2*RHO                                          210
      BOT=105.0-45.0*R2+R4                                               220
      PHI=RHO-ATAN(TOP/BOT)                                              230
      GO TO 32                                                           240
  100 RETURN                                                             250
      END                                                                260
```

```
      SUBROUTINE ANNICK(EL,EH,LRF,ZAI,ABN,LFW,NIS,MIS,LIS,SPI,AWRI,AP,     10
     *BK,NBT,JNT,BK2,Y,B,SIGS,E,SIGG,SIGF,SIGT,ESC,XT,YT,TEMP,             20
     *C,GX,GNO,GG,GF,AJ)                                                   30
      COMMON/INOUT/NO,NI,NT,MODE,LEDIT1,LEDIT2,NR1(76),                    40
     *MATKOK,NTYP,NAMISO(10),MATISO(10),NR2(11),NBX,N1X,N2X,N,NBKX,        50
     *NR3(10),EPS                                                          60
      COMMON/RECS/MAT,MF,MT,C1,C2,L1,L2,N1,N2                              70
      REAL*8 NAMISO,NAM8                                                   80
      COMMON/ELCT/ZYN,ZYF,U,V,RHON,RHOV,NDIM6,NPEMAX                       90
      DIMENSION SIGS(NPEMAX),E(NPEMAX),SIGG(NPEMAX),SIGF(NPEMAX),         100
     *SIGT(NPEMAX),D(NDIM6,NPEMAX),GX(NDIM6,NPEMAX),                      110
     *GNO(NDIM6,NPEMAX),GG(NDIM6,NPEMAX),GF(NDIM6,NPEMAX),                120
     *  AMUX(6),AMUN(6),AMUG(6),AMUF(6),AJ(NDIM6,NPEMAX),INTS(18),         130
     *ESC(NPEMAX),XT(NPEMAX),YT(NPEMAX),TEMP(NPEMAX),                     140
     *BK2(1),BK(1),NAM1(2),NBT(1),JNT(1),Y(1),B(1)                        150
      EQUIVALENCE (NAM8,NAM1(1))                                          160
      NB=1                                                                170
      NB2=1                                                               180
      IWT=0                                                               190
      LS=1                                                                200
      N=1                                                                 210
```

```
      IDIMBK=NBKX
      JLI=1
      DO 31 K=1,NPEMAX
      SIGS(K)=0.0
      SIGG(K)=0.0
      SIGF(K)=0.0
      SIGT(K)=0.0
   31 CONTINUE
      WRITE(NO,1)
    1 FORMAT(1H1,30X,'UNRESOLVED RESONANCE DATA')
      IF(NIS.GT.1)WRITE(NO,110)NAMISO(MIS)
  110 FORMAT(1H+,60X,'ISOTOPE',3X,A8)
      CALL PRIZEI
      CALL INITSC(NPEMAX)
      CALL RREC(1,NT,MODE,-1.0)
      SPI=C1
      A=C2
      LIS=L1
C
C     BEGINNING OF LOOP ON L STATES
C
   39 IF(LRF.EQ.2)GOTO 4
      IF(LFW)2,2,3
C-----FISSION-WIDTHS NOT GIVEN
C----- ALL PARAMETERS ARE ENERGY INDEPENDENT
    2 IF(LS.GT.1)GOTO 40
      LFCHG=0
      NLS=N1
C-----CALCULATES ENERGIES USING EQUAL LETHARGY
      E(1)=EL
      NPUNR=20
      DE=NPUNR-1
      DE=ALOG(EH/EL)/DE
      DE=EXP(DE)
      DO 5 K=2,NPUNR
      E(K)=E(K-1)*DE
    5 CONTINUE
      E(NPUNR)=EH
      ISTGF=0
   40 CALL RREC(2,NT,MODE,-1.0,B)
      AWRI=C1
      C=2.19685E-3*A*AWRI/(AWRI+1.0)
      L=L1
      NJS=N2
      IF(NJS.GT.6)CALL ERROR(600)
      JS=1
      NE=NPUNR
      DO 10 J=1,NJS
      DO 11 K=1,NE
      D(J,K)=B(JS)
      AJ(J,K)=B(JS+1)
      GNO(J,K)=B(JS+3)
      GG(J,K)=B(JS+4)
      GF(J,K)=0.0
      GX(J,K)=0.0
   11 CONTINUE
      AMUN(J)=B(JS+2)
      AMUX(J)=0.0
      AMUG(J)=0.0
      AMUF(J)=0.0
      INTS(JLI)=5
      JLI=JLI+1
      JS=JS+6
   10 CONTINUE
      GOTO 12
C-----FISSION-WIDTHS GIVEN
C-----ONLY FISSION-WIDTHS ARE ENERGY-DEPENDENT
    3 IF(LS.GT.1)GOTO 41
      LFCHG=1
      BACKSPACE NT
      CALL RREC(2,NT,MODE,-1.0,B)
      NE=N1
      IF(NE.GT.NPEMAX)CALL ERROR(220)
      NLS=N2
      DO 13 K=1,NE
      E(K)=B(K)
   13 CONTINUE
      ISTGF=1
   41 CALL RREC (1,NT,MODE,-1.0,B)
      AWRI=C1
      C=2.19685E-3*A*AWRI/(AWRI+1.0)
      L=L1
      NJS=N1
      IF(NJS.GT.6)CALL ERROR(600)
      DO 17 J=1,NJS
      CALL RREC(2,NT,MODE,-1.0,B)
      DO 18 K=1,NE
      D(J,K)=B(1)
      AJ(J,K)=B(2)
      GNO(J,K)=B(4)
      GG(J,K)=B(5)
      GX(J,K)=0.0
      GF(J,K)=B(K+6)
   18 CONTINUE
      AMUN(J)=B(3)
      AMUX(J)=0.0
      AMUG(J)=0.0
      AMUF(J)=L2
      INTS(JLI)=5
      JLI=JLI+1
   17 CONTINUE
      GOTO 12
C-----ALL ENERGY-DEPENDENT PARAMETERS
    4 IF(LS.GT.1)GOTO 42
      LFCHG=2
      NLS=N1
      ISTGF=1
   42 CALL RREC(1,NT,MODE,-1.0)
      AWRI=C1
      C=2.19685E-3*A*AWRI/(AWRI+1.0)
```

```
      L=L1                                                              1320
      NJS=N1                                                            1330
      IF(NJS.GT.6)CALL ERROR(600)                                       1340
      DO 22 J=1,NJS                                                     1350
      CALL RREC(2,NT,MODE,-1.0,B)                                       1360
      NE=N2                                                             1370
      IF(NE.GT.NPEMAX)CALL ERROR(220)                                   1380
      JK=1                                                              1390
      DO 23 K=1,NE                                                      1400
      E(K)=B(JK+6)                                                      1410
      D(J,K)=B(JK+7)                                                    1420
      AJ(J,K)=C1                                                        1430
      GNO(J,K)=B(JK+9)                                                  1440
      GG(J,K)=B(JK+10)                                                  1450
      GX(J,K)=B(JK+8)                                                   1460
      GF(J,K)=B(JK+11)                                                  1470
      JK=JK+6                                                           1480
   23 CONTINUE                                                          1490
      AMUN(J)=B(4)                                                      1500
      AMUX(J)=B(3)                                                      1510
      AMUG(J)=B(5)                                                      1520
      AMUF(J)=B(6)                                                      1530
      INTS(JLI)=L1                                                      1540
      JLI=JLI+1                                                         1550
   22 CONTINUE                                                          1560
C                                                                       1570
C=====PRINTING OF UNRESOLVED RESONANCE DATA                             1580
C                                                                       1590
   12 IF(IWT.GE.1)GOTO 25                                               1600
      WRITE(NO,24)EL,EH,SPI,NLS                                         1610
   24 FORMAT(1HO,'LOWER ENERGY LIMIT---------',1PE11.4/                 1620
     *       1X,'UPPER ENERGY LIMIT---------',E11.4/                    1630
     *       1X,'SPIN SCATTERING LENGTH-----',E11.4/                    1640
     *       1X,'NUMBER OF L STATES---------',I11///)                   1650
      IWT=IWT+1                                                         1660
   25 IF(LEDIT2.NE.1)GOTO 100                                           1670
      WRITE(NO,26)L                                                     1680
   26 FORMAT(1H1,10X,'L =',I3/11X,6('=')///)                            1690
      DO 27 J=1,NJS                                                     1700
      JINT=AJ(J,1)                                                      1710
      WRITE(NO,28)JINT                                                  1720
   28 FORMAT(//' J = ',I3/1X,7('-')//)                                  1730
      WRITE(NO,32) INTS(JLI-1-NJS+J)                                    1740
   32 FORMAT(' INTERPOLATION-CODE=',I1,'.')                             1750
      WRITE(NO,29)AMUN(J),AMUX(J),AMUG(J),AMUF(J)                       1760
   29 FORMAT(1X,'NUMBER OF DEGREES OF FREEDOM',                         1770
     *1X,'NEUTRON WIDTH-----' ,F5.1/                                    1780
     *30X,'COMPETITIVE WIDTH-',F5.1/                                    1790
     *30X,'RADIATION WIDTH---',F5.1/                                    1800
     *30X,'FISSION WIDTH-----',F5.1//)                                  1810
      WRITE(NO,30)                                                      1820
   30 FORMAT(3X,'ENERGY (EV)',6X,'MEAN LEVEL',5X,'AV NEUTRON',5X,       1830
     *  'AV RADIAT.',5X,'AV FISSION',5X,'AV COMPET.'/20X,               1840
     *'SPACING',10X,4('WIDTH',10X))                                     1850
      WRITE(NO,50)((E(K),D(J,K),GNO(J,K),GG(J,K),GF(J,K),GX(J,K)),      1860
     *    K=1,NE)                                                       1870
   50 FORMAT(1X,1P6E15.5)                                               1880
   27 CONTINUE                                                          1890
C                                                                       1900
C=====CALCULATION OF POINT-WISE CROSS-SECTIONS                          1910
C                                                                       1920
  100 T=C/A                                                             1930
      AWR=AWRI                                                          1940
      CONST=19.7392088/(T*T)                                            1950
      PIFOUR=12.566371                                                  1960
      ASS=AWR*1.008665                                                  1970
      CRT=ASS**(1.0/3.0)                                                1980
      TERM=0.123*CRT+0.08                                               1990
      APEN=TERM                                                         2000
      CPEN=C/A*APEN                                                     2010
      LI=L+1                                                            2020
      DO 33 J=1,NJS                                                     2030
      MU=IFIX(AMUN(J))                                                  2040
      IF(LFCHG.EQ.0)GOTO 300                                            2050
      N1=1                                                              2060
      DO 301 IE=1,LFCHG                                                 2070
      DO 302 K=1,NE                                                     2080
      XT(K)=E(K)                                                        2090
      IF(IE.EQ.2)GOTO 303                                               2100
      YT(K)=GF(J,K)                                                     2110
      GOTO 302                                                          2120
  303 YT(K)=GNO(J,K)                                                    2130
  302 CONTINUE                                                          2140
      NBT(1)=NE                                                         2150
      JNT(1)=INTS(JLI-1-NJS+J)                                          2160
      N2=NE                                                             2170
      CALL CHGINT(EPS,N1,N2,NBT,JNT,XT,YT,N1X,NPEMAX,&304)              2180
      GOTO 305                                                          2190
  304 CALL ERROR(221)                                                   2200
  305 CALL GENER(N2,XT,TEMP,ESC,ND,NCU)                                 2210
      IF(NCU.GT.NPEMAX)CALL ERROR(222)                                  2220
  301 CONTINUE                                                          2230
  300 NU=IFIX(AMUF(J))                                                  2240
      LAMBDA=IFIX(AMUX(J))                                              2250
      DO 34 K=1,NE                                                      2260
      GJ=(2.0*AJ(J,K)+1.0)/(4.0*SPI+2.0)                                2270
      E2=SQRT(E(K))                                                     2280
      EJ=E(K)                                                           2290
      WAVE=C/A*E2                                                       2300
      RHO=C*E2                                                          2310
      RHOPEN=CPEN*E2                                                    2320
      GOTO (35,36,37),LI                                                2330
   35 VL=E2*AMUN(J)                                                     2340
      PS=RHO                                                            2350
      GOTO 38                                                           2360
   36 VL=E2*RHOPEN**2/(1.0+RHOPEN**2)*AMUN(J)                           2370
      PS=RHO-ATAN(RHO)                                                  2380
      GOTO 38                                                           2390
   37 VL=E2*RHOPEN**4/(RHOPEN**4+3.0*RHOPEN**2+9.0)*AMUN(J)             2400
      PS=RHO-ATAN(3.0*RHO/(3.0-RHO**2))                                 2410
```

```
   38 SIGPOT=(SIN(PS)/WAVE)**2                                                  2420
      SIGPOT=SIGPOT*PIFOUR*(2.0*L+1.0)                                          2430
      GN=GNO(J,K)*VL                                                            2440
      GAMMA=GG(J,K)                                                             2450
      GALPHA=GN                                                                 2460
      GBETA=GF(J,K)                                                             2470
      DIFF=GX(J,K)                                                              2480
      GT=GN+GAMMA+GBETA+DIFF                                                    2490
      TERG=(CONST*GJ*GN*GAMMA)/(EJ*D(J,K))                                      2500
      TERS=(CONST*GJ*GN*GN)/(EJ*D(J,K))                                         2510
      TERF=(CONST*GJ*GN*GBETA)/(EJ*D(J,K))                                      2520
      CALL GNRL(GALPHA,GBETA,GAMMA,MU,NU,LAMBDA,GS,DIFF,1)                      2530
      CALL GNRL(GALPHA,GBETA,GAMMA,MU,NU,LAMBDA,GC,DIFF,2)                      2540
      CALL GNRL(GALPHA,GBETA,GAMMA,MU,NU,LAMBDA,GFF,DIFF,3)                     2550
      GS=GS*TERS                                                                2560
      GC=GC*TERG                                                                2570
      GFF=GFF*TERF                                                              2580
      CORR=(CONST*GJ*2.0*GN*SIN(PS)*SIN(PS))/(EJ*D(J,K))                        2590
      GS=GS-CORR                                                                2600
      SIGG(K)=SIGG(K)+GC                                                        2610
      SIGS(K)=SIGS(K)+GS                                                        2620
      SIGF(K)=SIGF(K)+GFF                                                       2630
      IF(ISTGF.EQ.0)GOTO 80                                                     2640
      IF(NB2.GT.N2X) CALL ERROR(105)                                            2650
      BK2(NB2)=E(K)                                                             2660
      BK2(NB2+1)=FLOAT(L)                                                       2670
      BK2(NB2+2)=AJ(J,K)                                                        2680
      BK2(NB2+3)=AMUF(J)                                                        2690
      BK2(NB2+4)=GF(J,K)                                                        2700
      BK2(NB2+5)=GG(J,K)                                                        2710
      BK2(NB2+6)=GN                                                             2720
      ZYN=AMUN(J)                                                               2730
      ZYF=AMUF(J)                                                               2740
      U=GN/GG(J,K)/AMUN(J)                                                      2750
      V=GF(J,K)/GG(J,K)                                                         2760
      RHOV=1.                                                                   2770
      RHON=0.                                                                   2780
      JINT=AJ(J,1)                                                              2790
      IF(V.NE.0.0.AND.ZYF.EQ.0.0) WRITE(NO,1000) L,JINT,K,V                     2800
 1000 FORMAT(' ***** WARNING IN SUBROUTINE ANNICK. NUMBER OF DEGREES OF         2810
     *FREEDOM FOR FISSION WIDTH'/                                               2820
     *' ***** IN THE UNRESOLVED RESONANCE REGION FOR L=',I2,' J=',I2,          2830
     *' K=',I3,' IS EQUAL ZERO AND'/                                            2840
     *' ***** V=GF(J,K)/GG(J,K) HAS THE VALUE=',1PE15.7/                        2850
     *' ***** THE STATISTICAL FLUCTUATION FACTORS ARE SET TO ZERO.')             2860
      IF(V.NE.0.0.AND.ZYF.NE.0.0) GOTO 200                                      2870
      PEER=0.                                                                   2880
      XEER=0.                                                                   2890
      RG=0.                                                                     2900
      RF=0.                                                                     2910
      GOTO 220                                                                  2920
  200 CALL ERF(PEER)                                                            2930
      RHOV=0.                                                                   2940
      CALL ERF(XEER)                                                            2950
      RF=(V*PEER*(ZYN+ZYF+2.)+ZYF*(XEER-1.))/(V*ZYN-L*ZYF)                      2960
      RHON=1.                                                                   2970
      CALL ERF(RG)                                                              2980
  220 BK2(NB2+7)=PEER                                                           2990
      BK2(NB2+8)=XEER                                                           3000
      BK2(NB2+9)=RF                                                             3010
      BK2(NB2+10)=RG                                                            3020
      NB2=NB2+11                                                                3030
   80 IF(J.GT.1)GOTO 34                                                         3040
      SIGS(K)=SIGS(K)+SIGPOT                                                    3050
   34 CONTINUE                                                                  3060
      IF(NB.GT.NBKX) CALL ERROR(204)                                            3070
      BK(NB)=FLOAT(L)                                                           3080
      BK(NB+1)=AJ(J,1)                                                          3090
      BK(NB+2)=GG(J,1)                                                          3100
      BK(NB+3)=D(J,1)                                                           3110
      BK(NB+4)=GNO(J,1)                                                         3120
      BK(NB+5)=GNO(J,1)/D(J,1)                                                  3130
      BK(NB+6)=AMUF(J)                                                          3140
      BK(NB+7)=AMUN(J)                                                          3150
      NB=NB+8                                                                   3160
   33 CONTINUE                                                                  3170
      LS=LS+1                                                                   3180
C                                                                               3190
C      END OF LOOP ON L STATES                                                  3200
C                                                                               3210
      IF(NLS.GT.3) CALL ERROR(601)                                              3220
      IF(LS.LE.NLS)GOTO 39                                                      3230
      CALL ANNI01(NE,SIGT,SIGG,SIGS,SIGF,NIS,MIS,E,NB,BK,ISTGF,NB2,             3240
     *BK2,LFCHG,NCU,ESC,IDIMBK,INTS,NBT,JNT,Y,EL,EH,LFW)                        3250
      RETURN                                                                    3260
      END                                                                       3270
```

```
      SUBROUTINE ANNI01(NE,SIGT,SIGG,SIGS,SIGF,NIS,MIS,E,NB,BK,ISTGF,          10
     *NB2,BK2,LFCHG,NCU,ESC,IDIMBK,INTS,NBT,JNT,Y,EL,EH,LFW)                    20
      DIMENSION SIGT(1),SIGG(1),SIGS(1),SIGF(1),E(1),NAM1(2),BK(1),             30
     *BK2(1),ESC(1),INTS(1),NBT(1),JNT(1),Y(1)                                  40
      REAL*8 NAMISO,NAM8                                                        50
      EQUIVALENCE (NAM8,NAM1(1))                                                60
      COMMON/INOUT/ NO,NI,NT,MODE,LEDIT1,LEDIT2,LEDITS,NR1(75),                 70
     *MATKOK,NTYP,NAMISO(10),MATISO(10),NR2(11),N8X,N1X,N2X,N,NBKX              80
      COMMON/RECS/ MAT,MF,MT,C1,C2,L1,L2,N1,N2                                  90
C                                                                               100
C     CALCULATION OF TOTAL CROSS-SECTION                                        110
C                                                                               120
      DO 44 K=1,NE                                                              130
      SIGT(K)=SIGG(K)+SIGS(K)+SIGF(K)                                           140
   44 CONTINUE                                                                  150
      MMIS=1                                                                    160
      MATK=MATKOK                                                               170
      IF(NIS.EQ.1)GOTO 105                                                      180
      MATK=MATISO(MIS)                                                          190
      MMIS=MIS+1                                                                200
```

```
  105 IF(LEDITS.NE.1)GOTO 101
      WRITE(NO,43)
   43 FORMAT(1H1,'POINT-WISE CROSS SECTIONS FROM UNRESOLVED PARAMETERS'
     *  ////1X,'ENERGY',13X,
     *       1X,'SCATTERING',5X,'CAPTURE',8X,'FISSION',8X,'TOTAL'/)
      WRITE(NO,60)(E(K),SIGS(K),SIGG(K),SIGF(K),SIGT(K),K=1,NE)
   60 FORMAT(1X,1P5E15.5)
  101 CALL XTEXT(4H    ,NAME2)
      N=NB-1
C
C     PRINTING OF KEDAK DATA TYPE ST
C
      IF(LEDIT2.NE.1)GOTO 102
      WRITE(NO,81)
   81 FORMAT(1H1,'KEDAK DATA TYPE ST')
      IF(NIS.GT.1)WRITE(NO,110)NAMISO(MIS)
  110 FORMAT(1H+,60X,'ISOTOPE',3X,A8)
      CALL XTEXT(4HST  ,NAM1(1))
      CALL XTEXT(4H    ,NAM1(2))
      CALL A8FORM(NO,NAM8)
      CALL LUCY(3,BK)
  102 CALL XTEXT(4HST  ,NAME1)
      CALL WDA2(MATK,MMIS, NAME1,NAME2)
C
C     REORDERING OF DATA IN STGF: E,L,J
C
      IF(ISTGF.EQ.0)GOTO 83
      N=NB2-1
      K=1
      DO 251 IE=1,NE
      K2=(IE-1)*11+1
  252 IF((K+11).GT.NBKX) CALL ERROR(204)
      DO 250 IK=1,11
      I=IK-1
  250 BK(K+I)=BK2(K2+I)
      K=K+11
      K2=K2+11*NE
      IF(K2-N)252,252,251
  251 CONTINUE
C
C     INSERTION IN STGF OF ENERGY POINTS ADDED BY CHGINT
C
      IF(LFCHG.EQ.0.OR.ISTGF.EQ.0)GOTO 312
      IF(NCU.EQ.NE)GOTO 310
C      SELECTION IN THE ARRAY ESC OF POINTS ADDED BY CHGINT
      DO 313 IE=1,NE
      DO 314 INCU=1,NCU
      IF(E(IE).GE.ESC(INCU)*0.99999.AND.E(IE).LE.ESC(INCU)*1.00001)
     *     GOTO 315
  314 CONTINUE
      GOTO 313
  315 NCU=NCU-1
      IF(NCU.EQ.0)CALL ERROR(224)
      DO 316 IESC=INCU,NCU
  316 ESC(IESC)=ESC(IESC+1)
  313 CONTINUE
      WRITE(NO,317)(ESC(INCU),INCU=1,NCU)
  317 FORMAT('0ENERGY POINTS ADDED BY CHGINT IN THE STGF DATA TYPE'//
     *     (1X,1P6E15.5))
C     INSERTION OF THE NEW ENERGY POINTS IN BK ARRAY
      NBL=N/NE
      DO 318 INCU=1,NCU
      IF((N+NBL).GT.IDIMBK)CALL ERROR(226)
      DO 319 IK=1,N,NBL
      IF(ESC(INCU).EQ.BK(IK))GOTO 318
      IF(ESC(INCU).LT.BK(IK))GOTO 320
  319 CONTINUE
      CALL ERROR(225)
  320 DO 321 IK1=IK,N
      NCRT=N-IK1+IK
  321 BK(NCRT+NBL)=BK(NCRT)
      N=N+NBL
      KE=NBL/11
      DO 323 NJK=1,KE
      BK(IK)=ESC(INCU)
      DO 322 JK=4,10
      IK2=IK+JK
      CALL TERP1(BK(IK-NBL),BK(IK2-NBL),BK(IK+NBL),BK(IK2+NBL),
     *     BK(IK),BK(IK2),INTS(NJK))
  322 CONTINUE
      IK=IK+11
  323 CONTINUE
  318 CONTINUE
      GOTO 312
  310 WRITE(NO,311)
  311 FORMAT('0NO ENERGY POINTS ADDED BY CHGINT IN STGF')
C
C     PRINTING OF KEDAK DATA TYPE STGF
C
  312 IF(LEDIT2.NE.1)GOTO 103
      WRITE(NO,84)
   84 FORMAT(1H1,'KEDAK DATA TYPE STGF')
      IF(NIS.GT.1)WRITE(NO,110)NAMISO(MIS)
      CALL XTEXT(4HSTGF,NAM1(1))
      CALL A8FORM(NO,NAM8)
      CALL LUCY(2,BK)
  103 CALL XTEXT(4HSTGF,NAME1)
      CALL WDA2(MATK,MMIS, NAME1,NAME2)
   83 N2=NE
      N1=1
      NBT(1)=N2
      JNT(1)=INTS(1)
C     BK2 AND X HAVE THE SAME STARTING ADRESS.
      CALL MAGGY(E,SIGS,NE,BK2,Y)
      CALL WRECS2(MIS,4HSCAT,4HSTAT,EL,EH)
      CALL MAGGY(E,SIGG,NE,BK2,Y)
      CALL WRECS2(MIS,4HCAPT,4HSTAT,EL,EH)
      IF(LFW.NE.1)GOTO 70
      CALL MAGGY(E,SIGF,NE,BK2,Y)
      CALL WRECS2(MIS,4HFISS,4HSTAT,EL,EH)
```

```
 70 CONTINUE                                                          1310
    RETURN                                                            1320
    END                                                               1330
```

```
    SUBROUTINE ERF(EER)                                                10
    COMMON/FLCT/ZYN,ZYF,U,V,RHON,RHOF                                  20
    SW=0.05                                                            30
    SUMME=0.                                                           40
    HU=0.                                                              50
    HO=SW                                                              60
    DO 1 J=1,9999                                                      70
    CALL GAUSS(HU,HO,B)                                                80
    HU=HO                                                              90
    HO=HO+SW                                                          100
    IF(SQRT(B*B)-1.E-6)6,6,3                                          110
  3 SUMME=SUMME+B                                                     120
  1 CONTINUE                                                          130
  6 SUMME=SUMME*(1.+2./ZYN)**RHON                                     140
    EER=SUMME                                                         150
    RETURN                                                            160
    END                                                               170
```

```
    SUBROUTINE GAUSS (HU,HO,B)                                         10
    D=(HO-HU)*0.5                                                      20
    E=(HO+HU)*0.5                                                      30
    B=(BINT(9.061799E-1*D+E)*2.369269E-1+BINT(5.384693E-1*D+E)         40
   1*4.786287E-1+BINT(E)*5.688889E-1+BINT(-5.384693E-1*D+E)            50
   2*4.786287E-1+BINT(-9.061799E-1*D+E)*2.369269E-1)*D                 60
    RETURN                                                             70
    END                                                                80
```

```
    FUNCTION BINT(X)                                                   10
    COMMON/FLCT/ZYN,ZYF,U,V,RHON,RHOF                                  20
    Z=ZYN*0.5+1.+RHON                                                  30
    Y=ZYF*0.5+RHOF                                                     40
    BINT=EXP(-X)/((1.+2.*U*X/ZYN)**Z*(1.+2.*V*X/ZYF)**Y)               50
    RETURN                                                             60
    END                                                                70
```

```
    SUBROUTINE GNRL(GALPHA,GBETA,GAMMA,MU,NU,LAMBDA,S,DF,ID)           10
    COMMON/XBAR/XX(4,10)                                               20
    S=0.0                                                              30
    IF(GALPHA)1000,1000,1001                                           40
1001 IF(GAMMA)1000,1000,1002                                           50
1002 IF(GBETA)1000,1003,1004                                           60
1003 IF(DF)1000,1005,1006                                              70
1005 DO 100 J=1,10                                                     80
    XJ=XX(MU,J)                                                        90
    GO TO(200,201,202),ID                                             100
 200 S=S+((XJ*XJ)/(GALPHA*XJ+GAMMA))                                  110
    GO TO 100                                                         120
 201 S=S+(XJ/(GALPHA*XJ+GAMMA))                                       130
 202 CONTINUE                                                         140
 100 CONTINUE                                                         150
    S=S/10.0                                                          160
    GO TO 1000                                                        170
1006 DO 101 J=1,10                                                    180
    XJ=XX(MU,J)                                                       190
    DO 102 K=1,10                                                     200
    XK=XX(LAMBDA,K)                                                   210
    GO TO(300,301,302),ID                                             220
 300 S=S+((XJ*XJ)/(GALPHA*XJ+GAMMA+DF*XK))                            230
    GO TO 102                                                         240
 301 S=S+(XJ/(GALPHA*XJ+GAMMA+DF*XK))                                 250
 302 CONTINUE                                                         260
 102 CONTINUE                                                         270
 101 CONTINUE                                                         280
    S=S/100.0                                                         290
    GO TO 1000                                                        300
1004 IF(DF)1000,1007,1008                                             310
1007 DO 103 J=1,10                                                    320
    XJ=XX(MU,J)                                                       330
    DO 104 K=1,10                                                     340
    XK=XX(NU,K)                                                       350
    GO TO(400,401,402),ID                                             360
 400 S=S+((XJ*XJ)/(GALPHA*XJ+GBETA*XK+GAMMA))                         370
    GO TO 104                                                         380
 401 S=S+(XJ/(GALPHA*XJ+GBETA*XK+GAMMA))                              390
    GO TO 104                                                         400
 402 S=S+((XJ*XK)/(GALPHA*XJ+GBETA*XK+GAMMA))                         410
 104 CONTINUE                                                         420
 103 CONTINUE                                                         430
    S=S/100.0                                                         440
    GO TO 1000                                                        450
1008 DO 105 J=1,10                                                    460
    XJ=XX(MU,J)                                                       470
    DO 106 K=1,10                                                     480
    XK=XX(NU,K)                                                       490
    DO 107 L=1,10                                                     500
    XL=XX(LAMBDA,L)                                                   510
    GO TO(500,501,502),ID                                             520
 500 S=S+((XJ*XJ)/(GALPHA*XJ+GBETA*XK+GAMMA+DF*XL))                   530
    GO TO 107                                                         540
 501 S=S+(XJ/(GALPHA*XJ+GBETA*XK+GAMMA+DF*XL))                        550
```

```
      GO TO 107                                                          560
  502 S=S+((XJ*XK)/(GALPHA*XJ+GBETA*XK+GAMMA*DF*XL))                     570
  107 CONTINUE                                                           580
  106 CONTINUE                                                           590
  105 CONTINUE                                                           600
      S=S/1000.0                                                         610
 1000 RETURN                                                             620
      END                                                                630
```

```
      FUNCTION ADD(YA,YB,CON)                                             10
      DIMENSION CON(20)                                                   20
      ADD=YA+YB                                                           30
      RETURN                                                              40
      END                                                                 50
```

```
      FUNCTION SUB(YA,YB,CON)                                             10
      SUB=YA-YB                                                           20
      RETURN                                                              30
      END                                                                 40
```

```
      REAL FUNCTION MULT(YA,YB,CON)                                       10
      MULT=YA*YB                                                          20
      RETURN                                                              30
      END                                                                 40
```

```
      SUBROUTINE MARION(LFW,EL,EH,EPS,NAME2)                              10
      COMMON/INOUT/NO                                                     20
      EXTERNAL ADD                                                        30
      DATA EPSMAX/2.E-2/,EPSINC/1.E-3/                                    40
      CALL RRECS(0,4HSGN ,NAME2 ,EL,EH,LNTR)                              50
      IF(LNTR.NE.0) CALL ERROR(777)                                       60
      CALL STORE(3,1000,LOF)                                              70
      IF(LOF.EQ.1)CALL ERROR(408)                                         80
      CALL RRECS(0,4HSGG ,NAME2 ,EL,EH,LNTR)                              90
      IF(LNTR.NE.0) CALL ERROR(777)                                      100
      CALL STORE(3,1001,LOF)                                             110
      IF(LOF.EQ.1)CALL ERROR(408)                                        120
    3 CALL COMB(ADD,CON,1000,1001,EL ,EH ,EPS,LOF,LNT)                   130
      IF(LOF.EQ.0)GOTO 2                                                 140
      EPS=EPS+EPSINC                                                     150
      IF(EPS-EPSMAX)3,3,4                                                160
    4 CALL ERROR(409)                                                    170
    2 IF(LNT.EQ.1)CALLERROR(410)                                         180
      CALL STORE(3,1000,LOF)                                             190
      IF(LOF.EQ.1)CALL ERROR(408)                                        200
      IF(LFW.NE.1)GOTO1                                                  210
      CALL RRECS(0,4HSGF ,NAME2 ,EL,EH,LNTR)                             220
      IF(LNTR.NE.0) CALL ERROR(777)                                      230
      CALL STORE(3,1001,LOF)                                             240
      IF(LOF.EQ.1)CALL ERROR(408)                                        250
    6 CALL COMB(ADD,CON,1000,1001,EL ,EH ,EPS,LOF,LNT)                   260
      IF(LOF.EQ.0)GOTO 5                                                 270
      EPS=EPS+EPSINC                                                     280
      IF(EPS-EPSMAX)6,6,7                                                290
    7 CALL ERROR(409)                                                    300
    5 IF(LNT.EQ.1)CALL ERROR(410)                                        310
      CALL STORE(3,1000,LOF)                                             320
      IF(LOF.EQ.1)CALL ERROR(408)                                        330
    1 CONTINUE                                                           340
      CALL FETCH(1000,LNT)                                               350
      IF(LNT.EQ.1)CALL ERROR(410)                                        360
      WRITE(NO,10)EPS                                                    370
   10 FORMAT(1H0,'EPS=',1PE12.4,' FOR CALCULATING SGT'//)                380
      RETURN                                                             390
      END                                                                400
```

```
      SUBROUTINE MYRIAM(NIS,LISRES,LISUNR,NAME,EL1,EH1,EPS,ITEST,Y)       10
      EXTERNAL ADD                                                        20
      DIMENSION CON(20),Y(1)                                              30
      EL1=1.0E8                                                           40
      EH1=1.0E-6                                                          50
      IFIRST=0                                                            60
      DO 10 MIS=1,NIS                                                     70
    1 IF(ITEST.EQ.1)GOTO 5                                                80
      IF(LISRES.NE.0)RETURN                                               90
      CALL RRECS(MIS,NAME,4H RES,EL,EH,LNTR)                             100
      IF(LNTR.NE.0) CALL ERROR(777)                                      110
      CALL BETSY(MIS,Y)                                                  120
      EL1=AMIN1(EL,EL1)                                                  130
      EH1=AMAX1(EH,EH1)                                                  140
      IF(IFIRST)4,4,3                                                    150
    3 CALL STORE(3,1001,LOF)                                             160
      IF(LOF.EQ.1)CALL ERROR(407)                                        170
      CALL COMB(ADD,CON,1000,1001,EL1,EH1,EPS,LOF,LNT)                   180
      IF(LOF.EQ.1)CALL ERROR(403)                                        190
      IF(LNT.EQ.1)CALL ERROR(404)                                        200
      CALL STORE(3,1000,LOF)                                             210
      IF(LOF.EQ.1)CALL ERROR(407)                                        220
      GOTO 9                                                             230
    4 CALL STORE(3,1000,LOF)                                             240
      IF(LOF.EQ.1)CALL ERROR(407)                                        250
      IFIRST=1                                                           260
      GOTO 9                                                             270
    5 IF(LISUNR.NE.0)RETURN                                              280
      CALL RRECS(MIS,NAME,4HSTAT,EL,EH,LNTR)                             290
```

```
      IF(LNTR.NE.0) CALL ERROR(777)                                     300
      CALL BETSY(MIS,Y)                                                 310
      EL1=AMIN1(EL,EL1)                                                 320
      EH1=AMAX1(EH,EH1)                                                 330
      IF(IFIRST)8,8,7                                                   340
    7 CALL STORE(3,1001,LOF)                                            350
      IF(LOF.EQ.1)CALL ERROR(407)                                       360
      CALL COMB(ADD,CON,1000,1001,EL1,EH1,EPS,LOF,LNT)                  370
      IF(LOF.EQ.1)CALL ERROR(403)                                       380
      IF(LNT.EQ.1)CALL ERROR(404)                                       390
      CALL STORE(3,1000,LOF)                                            400
      IF(LOF.EQ.1)CALL ERROR(407)                                       410
      GOTO 9                                                            420
    8 CALL STORE(3,1000,LOF)                                            430
      IF(LOF.EQ.1)CALL ERROR(407)                                       440
      IFIRST=1                                                          450
    9 CONTINUE                                                          460
   10 CONTINUE                                                          470
      CALL FETCH(1000,LNT)                                              480
      IF(LNT.EQ.1)CALL ERROR(404)                                       490
      RETURN                                                            500
      END                                                               510
```

```
      SUBROUTINE SOPHIE(MATP,LISRES,LISUNR,LFW,LNU,EPS,BK,XIN,YIN,NCMAX,   10
     *NBT,JNT,X,Y,B,TEMP)                                                 20
C----------CHANGED OCTOBRE 1973 BY E.STEIN.                               30
      LOGICAL*4 SALINE,SECOND,LODZMS                                      40
      COMMON/RECS/MAT,MF,MT,C1,C2,L1,L2,N1,N2                             50
      COMMON/INOUT/NO,NI,NT,MODE,LEDIT1,LEDIT2,LEDITS,KEDAVE,NR1(20),     60
     * NOTTRL(50),BIB12,SALINE,NR2(2),MATKCK,NR3(42),NEX,N1X,N2X          70
      COMMON/TAB/ENDTAB(25),KEDTAB(50),KEDR(240),NCNAME(39),NRIF3,NRIF4   80
      EQUIVALENCE (KEDR(1),KDR1(1)),(KEDR(124),KDR2(1))                   90
      INTEGER ENDTAB,BLANK,SGIZ,CZ,BIB12                                 100
      EXTERNAL ADD,SUB,MULT                                              110
      DIMENSION IABSP(8),CON(1),TEMP(1),NAM1(2),XIN(1),YIN(1),BK(1),     120
     *KDR1(123),KDR2(1),NBT(1),JNT(1),X(1),Y(1),B(1)                     130
      DATA BLANK/'    '/,SGIZ/'SGIZ'/,CZ/'C   '/                         140
      REAL*8 NAM8                                                        150
      EQUIVALENCE (NAM1(1),NAM8,NAMEK1),(NAMEK2,NAM1(2))                 160
      WRITE(NO,6)                                                        170
    6 FORMAT(1H1,20X,'FILE 3  -  SMOOTH CROSS SECTIONS'//21X,32('*')///) 180
      WRITE(NO,21)                                                       190
   21 FORMAT(1X,'PRINTING AFTER COMBINATION WITH FILE 2 DATA, BEFORE CHG 200
     *INT'//)                                                            210
      CALL PRIZEI                                                        220
      NF=10                                                              230
      CALL INITSC(NCMAX)                                                 240
      REWIND NF                                                          250
      IFSGIP=0                                                           260
      IFSGIC=0                                                           270
      IFSGA=0                                                            280
      IFSGI=0                                                            290
```

```
      DO 5 I=1,25                                                       300
      MTP=ENDTAB(I)                                                     310
      NAMEK1=KEDTAB(2*I-1)                                              320
      NAMEK2=KEDTAB(2*I)                                                330
      DO 80 INTRL=1,50,2                                                340
      IF(NOTTRL(INTRL).NE.NAMEK1)GOTO 80                                350
      IF(NOTTRL(INTRL+1).NE.NAMEK2)GOTO 80                              360
      WRITE(NO,85)NAMEK1,NAMEK2                                         370
   85 FORMAT('0 DATA TYPE ',2A4,' NOT TRANSLATED. (USER''S REQUEST)')   380
      GOTO 51                                                           390
   80 CONTINUE                                                          400
    1 CALL SEARCH(NT,MODE,MATP,3,MTP,LNT)                               410
      IF(LNT.NE.1)GOTO 3                                                420
      WRITE(NO,2) MTP                                                   430
    2 FORMAT(1HO,'DATA TYPE NUMBER',I5,' NOT INCLUDED IN THE ENDF/B FILE 440
     *')                                                                450
      BACKSPACE NT                                                      460
      MT=MTP                                                            470
      GOTO 51                                                           480
    3 CALL RREC(3,NT,MODE,-1.0,NBT,JNT,X,Y)                             490
      IF(MT.EQ.27)IFSGA=1                                               500
      EL=X(1)                                                           510
      EH=X(N2)                                                          520
      IF(MT.NE.18)GOTO 7                                                530
      IF(LNU.NE.1)GOTO 7                                                540
      DO 18 J=1,6                                                       550
      IF(Y(J).GT.1.E-6)GOTO 8                                           560
   18 CONTINUE                                                          570
    8 IF(J-1)9,9,10                                                     580
    9 ENU=X(1)                                                          590
      GOTO 11                                                           600
   10 ENU=X(J-1)                                                        610
   11 ENUMAX=X(N2)                                                      620
    7 IF(LISRES.EQ.1.AND.LISUNR.EQ.1)GOTO 4                             630
      IF(MT.GT.2.AND.MT.NE.18.AND.MT.NE.102)GOTO 4                      640
      CALL STORE(3,1000,LOF)                                            650
      IF(LOF.EQ.1)CALL ERROR(500)                                       660
      IF(MT.EQ.1)CALL XTEXT(4HSGT ,NAME1)                               670
      IF(MT.EQ.2)CALL XTEXT(4HSGN ,NAME1)                               680
      IF(MT.EQ.18)CALL XTEXT(4HSGF ,NAME1)                              690
      IF(MT.EQ.102)CALL XTEXT(4HSGG ,NAME1)                             700
      IF(LFW.NE.1.AND.MT.EQ.18)GOTO 4                                   710
      DO 20 NA=1,2                                                      720
      IF(NA.EQ.1)CALL XTEXT(4HRES ,NAME2)                               730
      IF(NA.EQ.2)CALL XTEXT(4HSTAT,NAME2)                               740
      CALL RRECS(0,NAME1,NAME2,EL1,EH1,LNTR,NBT,JNT,X,Y)                750
      IF(LNTR.EQ.1)GOTO 20                                              760
      CALL STORE(3,1001,LOF)                                            770
      IF(LOF.EQ.1)CALL ERROR(500)                                       780
      EL=AMIN1(EL,EL1)                                                  790
      EH=AMAX1(EH,EH1)                                                  800
      CALL COMB(ADD,CON,1000,1001,EL,EH,EPS,LOF,LNT,NBT,JNT,X,Y)        810
      IF(LOF.EQ.1)CALL ERROR(501)                                       820
      IF(LNT.EQ.1)CALL ERROR(502)                                       830
      CALL STORE(3,1000,LOF)                                            840
```

```
      IF(LOF.EQ.1)CALL ERROR(500)                                       850
   20 CONTINUE                                                          860
    4 IF(KEDAVE.EQ.0.AND.MT.EQ.4)GOTO 86                                870
      CALL WRECS2(0,MF,MT,0.0,0.0)                                      880
   86 IF(LEDITS.NE.1)GOTO 50                                            890
      WRITE(NO,22)MT,NAMEK1,NAMEK2                                      900
   22 FORMAT(1H1,'ENDF/B REACTION TYPE',I4,20X,'KEDAK REACTION TYPE',   910
     *   3X,2A4/)                                                       920
      CALL PRIZEI                                                       930
      CALL WREC(3,NO,4)                                                 940
C-----INELASTIC CROSS-SECTION                                           950
   50 IF(KEDAVE.NE.0.OR.MT.NE.4)GOTO 51                                 960
C-----ENERGY GRID FROM REACTION 4                                       970
      IFSGI=1                                                           980
      CALL WREC(3,NF,2)                                                 990
      CALL CHGINT(EPS,N1,N2,NBT,JNT,X,Y,N1X,N2X,&70)                   1000
      IF(N2.GT.NCMAX)GOTO 76                                           1010
      CALL GENER(N2,X,TEMP,XIN,ND,NCU)                                 1020
      IF(NCU.GT.NCMAX)CALL ERROR(506)                                  1030
      NKI=NCU                                                          1040
      GOTO 5                                                           1050
C-----INELASTIC EXCITATION CROSS-SECTIONS    MT=51 TO 90                1060
   51 IF(MT.NE.29)GOTO 5                                               1070
      IN=50                                                            1080
      DO 81 INTRL=1,50,2                                               1090
      IF(NOTTRL(INTRL).NE.SGIZ)GOTO 81                                 1100
      IF(NOTTRL(INTRL+1).NE.BLANK)GOTO 81                              1110
      WRITE(NO,85)SGIZ,BLANK                                           1120
      GOTO 65                                                          1130
   81 CONTINUE                                                         1140
      SECOND=.FALSE.                                                   1150
      INANF=51                                                         1160
      NAMEK1=SGIZ                                                      1170
      NAMEK2=BLANK                                                     1180
      DO 53 IN=51,90                                                   1190
      CALL SEARCH(NT,MODE,MATP,3,IN,LNT)                               1200
      IF(LNT.NE.1)GOTO 55                                              1210
      WRITE(NO,2)IN                                                    1220
      BACKSPACE NT                                                     1230
      IF(.NOT.SALINE.OR.SECOND) GOTO 65                                1240
      GOTO 53                                                          1250
   55 IF(SECOND) GOTO 551                                              1260
      SECOND=.TRUE.                                                    1270
      INANF=IN                                                         1280
  551 CALL RREC(3,NT,MODE,-1.0,NBT,JNT,X,Y)                            1290
      IF(KEDAVE.EQ.0) GOTO 552                                         1300
      EXC=ABS(C2)                                                      1310
      NRIF3=NRIF3+1                                                    1320
      CALL WRECS2(0,MF,IN,EXC,0.0)                                     1330
      GOTO 553                                                         1340
  552 IF(N2.GT.NCMAX)CALL ERROR(510)                                   1350
      IFSGIP=1                                                         1360
  553 IF(LEDITS.NE.1) GOTO 400                                         1370
      WRITE(NO,22) IN,NAMEK1,NAMEK2                                    1380
      CALL PRIZEI                                                      1390
      CALL WREC(3,NO,4)                                                1400
  400 IF(KEDAVE.NE.0) GOTO 53                                          1410
      CALL WREC(3,NF,2)                                                1420
      CALL CHGINT(EPS,N1,N2,NBT,JNT,X,Y,N1X,N2X,&70)                   1430
      IF(N2.GT.NCMAX)GOTO 76                                           1440
      CALL GENER(N2,X,TEMP,XIN,ND,NCU)                                 1450
      IF(NCU.GT.NCMAX)CALL ERROR(506)                                  1460
      NKI=NCU                                                          1470
   53 CONTINUE                                                         1480
      IN=91                                                            1490
C-----INELASTIC CROSS-SECTION TO THE CONTINUUM    MT=91                1500
   65 INFN=IN-1                                                        1510
      DO 82 INTRL=1,50,2                                               1520
      IF(NOTTRL(INTRL).NE.SGIZ)GOTO 82                                 1530
      IF(NOTTRL(INTRL+1).NE.CZ)GOTO 82                                 1540
      WRITE(NO,85)SGIZ,CZ                                              1550
      GOTO 5                                                           1560
   82 CONTINUE                                                         1570
      IN=91                                                            1580
      CALL SEARCH(NT,MODE,MATP,3,91,LNT)                               1590
      IF(LNT.NE.1)GOTO 66                                              1600
      WRITE(NO,2)IN                                                    1610
      BACKSPACE NT                                                     1620
      GOTO 5                                                           1630
   66 CALL RREC(3,NT,MODE,-1.0,NBT,JNT,X,Y)                            1640
      IFSGIC=1                                                         1650
      IF(LEDITS.NE.1) GOTO 401                                         1660
      NAMEK2=CZ                                                        1670
      WRITE(NO,22) IN,NAMEK1,NAMEK2                                    1680
      CALL PRIZEI                                                      1690
      CALL WREC(3,NO,4)                                                1700
  401 IF(KEDAVE.EQ.0) GOTO 554                                         1710
      CALL WRECS2(0,MF,IN,0.0,0.0)                                     1720
      GOTO 5                                                           1730
  554 CALL WREC(3,NF,2)                                                1740
      CALL CHGINT(EPS,N1,N2,NBT,JNT,X,Y,N1X,N2X,&70)                   1750
      IF(N2.GT.NCMAX)GOTO 76                                           1760
      CALL GENER(N2,X,TEMP,XIN,ND,NCU)                                 1770
      IF(NCU.GT.NCMAX)CALL ERROR(506)                                  1780
      NKI=NCU                                                          1790
    5 CONTINUE                                                         1800
C                                                                      1810
C                                                                      1820
      IF(KEDAVE.NE.0) GOTO 74                                          1830
      REWIND NF                                                        1840
      IF(IFSGI.EQ.0)GOTO 71                                            1850
C                                                                      1860
C-----REREAD TOTAL INELASTIC CROSS-SECTION FROM TEMP. FILE NF          1870
C                                                                      1880
      CALL RREC(3,NF,2,-1.0,NBT,JNT,X,Y)                               1890
   61 DO 62 KI=1,NKI                                                   1900
   62 CALL ITABI(XIN(KI),YIN(KI))                                      1910
      DO 63 KI=1,NKI                                                   1920
      X(KI)=XIN(KI)                                                    1930
      Y(KI)=YIN(KI)                                                    1940
```

```
   63 CONTINUE                                                          1950
      N2=NKI                                                            1960
      N1=1                                                              1970
      NBT(1)=NKI                                                        1980
      JNT(1)=2                                                          1990
      IF(Y(1).LE.1.0E-20) Y(1)=0.0                                      2000
      IF(MT.NE.4)GOTO 72                                                2010
      CALL WRECS2(0,MF,MT,0.0,0.0)                                      2020
      GOTO 71                                                           2030
   72 IF(IN.EQ.91)GOTO 68                                               2040
      EXC=ABS(C2)                                                       2050
      NRIF3=NRIF3+1                                                     2060
      CALL WRECS2(0,MF,IN,EXC,0.0)                                      2070
      GOTO 73                                                           2080
   68 CALL WRECS2(0,MF,IN,0.0,0.0)                                      2090
      GOTO 74                                                           2100
C                                                                       2110
C-----REREAD PARTIAL INEL. CROSS-SECTIONS FROM TEMP.FILE NF             2120
C                                                                       2130
   71 IF(IFSGIP.EQ.0)GOTO 75                                            2140
      DO 73 IN=INANF,INFN                                               2150
      CALL RREC(3,NF,2,-1.0,NBT,JNT,X,Y)                                2160
      GOTO 61                                                           2170
   73 CONTINUE                                                          2180
C                                                                       2190
C-----REREAD INEL. CROSS-SECTION TO THE CONTINUUM FROM TEMP. FILE NF    2200
C                                                                       2210
   75 IF(IFSGIC.EQ.0)GOTO 74                                            2220
      IN=91                                                             2230
      CALL RREC(3,NF,2,-1.0,NBT,JNT,X,Y)                                2240
      GOTO 61                                                           2250
   74 IF(IFSGA.EQ.1)GOTO 34                                             2260
C-----ABSORPTION CROSS SECTION      MT=27      SGA                      2270
      NAMEK1=KEDTAB(23)                                                 2280
      NAMEK2=KEDTAB(24)                                                 2290
      DO 84 INTRL=1,50,2                                                2300
      IF(NOTTRL(INTRL).NE.NAMEK1)GOTO 84                                2310
      IF(NOTTRL(INTRL+1).NE.NAMEK2)GOTO 84                              2320
      WRITE(NO,85)NAMEK1,NAMEK2                                         2330
      GOTO 34                                                           2340
   84 CONTINUE                                                          2350
      IABSP(1)=15                                                       2360
      IABSP(2)=7                                                        2370
      DO 30 I=3,8                                                       2380
   30 IABSP(I)=I+13                                                     2390
      EL=1.E8                                                           2400
      EH=1.E-10                                                         2410
      WRITE(NO,1000)                                                    2420
 1000 FORMAT(///'0SGA WILL BE CALCULATED BY USING THE FOLLOWING CROSS-SE 2430
     *CTIONS:'/)                                                        2440
      SECOND=.FALSE.                                                    2450
      DO 31 I=1,8                                                       2460
      MTP=ENDTAB(IABSP(I))                                              2470
      RN=0.0                                                            2480
      CALL RRECS(0,3,MTP,RN,RN,LNT,NBT,JNT,X,Y)                         2490
      MT=MTP                                                            2500
      IF(LNT.NE.0)GOTO 31                                               2510
      WRITE(NO,33) KEDTAB(2*IABSP(I)-1),KEDTAB(2*IABSP(I))              2520
   33 FORMAT(1X,2A4)                                                    2530
   32 EL=AMIN1(EL,X(1))                                                 2540
      EH=AMAX1(EH,X(N2))                                                2550
      IF(SECOND)GOTO 45                                                 2560
      CALL STORE(3,1000,LOF)                                            2570
      IF(LOF.EQ.1)CALL ERROR(500)                                       2580
      SECOND=.TRUE.                                                     2590
      GOTO 31                                                           2600
   45 CALL STORE(3,1001,LOF)                                            2610
      IF(LOF.EQ.1)CALL ERROR(500)                                       2620
      CALL COMB(ADD,CON,1000,1001,EL,EH,EPS,LOF,LNT,NBT,JNT,X,Y)        2630
      IF(LOF.EQ.1)CALL ERROR(501)                                       2640
      IF(LNT.EQ.1)CALL ERROR(502)                                       2650
      CALL STORE(3,1000,LOF)                                            2660
      IF(LOF.EQ.1)CALL ERROR(500)                                       2670
   31 CONTINUE                                                          2680
      IF(SECOND) GOTO 23                                                2690
      N1=1                                                              2700
      N2=2                                                              2710
      NBT(1)=2                                                          2720
      JNT(1)=2                                                          2730
      X(1)=1.0E-5                                                       2740
      Y(1)=0.0                                                          2750
      X(2)=20.0E+6                                                      2760
      Y(2)=0.0                                                          2770
      CALL STORE(3,1000,LOF)                                            2780
      IF(LOF.EQ.1) CALL ERROR(500)                                      2790
      WRITE(NO,1001)                                                    2800
 1001 FORMAT('0THE CROSS-SECTION TYPE SGA HAS BEEN SET TO 0.0 IN THE WHO 2810
     *LE ENERGY RANGE ( 1.0E-5 TO 20.0E+6 ) .'/                         2820
     *' THIS HAS BEEN DONE, BECAUSE NO CROSS-SECTIONS NECESSARY TO CALCU 2830
     *LATE SGA HAVE BEEN FOUND.')                                       2840
   23 MT=27                                                             2850
      MF=3                                                              2860
      IF(LEDITS.NE.1)GOTO 35                                            2870
      WRITE(NO,22)MT,NAMEK1,NAMEK2                                      2880
      CALL PRIZEI                                                       2890
      CALL WREC(3,NO,4)                                                 2900
   35 CALL WRECS2(0,MF,MT,0.0,0.0)                                      2910
C-----TRANSPORT CROSS SECTION      MT=201      SGTR                     2920
   34 NAMEK1=KEDTAB(43)                                                 2930
      NAMEK2=KEDTAB(44)                                                 2940
      DO 83 INTRL=1,50,2                                                2950
      IF(NOTTRL(INTRL).NE.NAMEK1)GOTO 83                                2960
      IF(NOTTRL(INTRL+1).NE.NAMEK2)GOTO 83                              2970
      WRITE(NO,85)NAMEK1,NAMEK2                                         2980
      GOTO 38                                                           2990
   83 CONTINUE                                                          3000
      RN=0.0                                                            3010
      CALL RRECS(0,3,2,RN,RN,LNT,NBT,JNT,X,Y)                           3020
      IF(LNT.EQ.0)GOTO 36                                               3030
      WRITE(NO,37)KEDTAB(3),KEDTAB(43)                                  3040
```

```
   37 FORMAT(//1X,A4,2X,' NOT PRESENT.  ',A4,' CANNOT BE CALCULATED'//)  3050
      GOTO 38                                                            3060
   36 EL=X(1)                                                            3070
      EH=X(N2)                                                           3080
      CALL STORE(3,1000,LOF)                                             3090
      IF(LOF.EQ.1)CALL ERROR(500)                                        3100
      RN=0.0                                                             3110
      CALL RRECS(0,3,251,RN,RN,LNT,NBT,JNT,X,Y)                          3120
      IF(LNT.EQ.0)GOTO 39                                                3130
      WRITE(NO,37)KEDTAB(49),KEDTAB(43)                                  3140
      GOTO 38                                                            3150
   39 CALL STORE(3,1001,LOF)                                             3160
      IF(LOF.EQ.1)CALL ERROR(500)                                        3170
      EL=AMIN1(EL,X(1))                                                  3180
      EH=AMAX1(EH,X(N2))                                                 3190
      CALL COMB(MULT,CON,1000,1001,EL,EH,EPS,LOF,LNT,NBT,JNT,X,Y)        3200
      IF(LOF.EQ.1)CALL ERROR(501)                                        3210
      IF(LNT.EQ.1) CALL ERROR(502)                                       3220
      CALL STORE(3,1001,LOF)                                             3230
      IF(LOF.EQ.1)CALL ERROR(500)                                        3240
      RN=0.0                                                             3250
      CALL RRECS(0,3,1,RN,RN,LNT,NBT,JNT,X,Y)                            3260
      IF(LNT.EQ.0)GOTO 40                                                3270
      WRITE(NO,37)KEDTAB(1),KEDTAB(43)                                   3280
      GOTO 38                                                            3290
   40 CALL STORE(3,1000,LOF)                                             3300
      IF(LOF.EQ.1)CALL ERROR(500)                                        3310
      EL=AMIN1(EL,X(1))                                                  3320
      EH=AMAX1(EH,X(N2))                                                 3330
      CALL COMB(SUB,CON,1000,1001,EL,EH,EPS,LOF,LNT,NBT,JNT,X,Y)         3340
      IF(LOF.EQ.1) CALL ERROR(501)                                       3350
      IF(LNT.EQ.1) CALL ERROR(502)                                       3360
      MT=201                                                             3370
      MF=3                                                               3380
      IF(LEDITS.NE.1)GOTO 41                                             3390
      WRITE(NO,22)MT,NAMEK1,NAMEK2                                       3400
      CALL PRIZEI                                                        3410
      CALL WREC(3,NO,4)                                                  3420
   41 CALL WRECS2(0,MF,MT,0.0,0.0)                                       3430
C-----ETA CROSS SECTION      MT=206      ETA                            3440
   38 CONTINUE                                                           3450
C-----NUE DATA TYPE    MT=452                                           3460
      IF(LNU.NE.1)RETURN                                                 3470
      CALL RRECS4(0,4H NUE,4H     ,LNT,8)                                3480
      IF(LNT.EQ.1)RETURN                                                 3490
      NB=N1                                                              3500
      X(1)=ENU                                                           3510
      DE=ALOG(ENUMAX/ENU)/199                                            3520
      DE=EXP(DE)                                                         3530
      DO 12 K=2,200                                                      3540
      X(K)=X(K-1)*DE                                                     3550
   12 CONTINUE                                                           3560
      DO 13 K=1,200                                                      3570
      Y(K)=B(1)                                                          3580
      IF(NB-2)13,14,15                                                   3590
   14 Y(K)=B(1)+B(2)*X(K)                                                3600
      GOTO 13                                                            3610
   15 ET=X(K)                                                            3620
      DO 16 I=2,NB                                                       3630
      Y(K)=Y(K)+B(I)*ET                                                  3640
      ET=ET*X(K)                                                         3650
   16 CONTINUE                                                           3660
   13 CONTINUE                                                           3670
      N1=1                                                               3680
      NBT(1)=200                                                         3690
      JNT(1)=2                                                           3700
      N2=200                                                             3710
      CALL WRECS2(0,1,452,0.0,0.0)                                       3720
      RETURN                                                             3730
   70 CALL ERROR(505)                                                    3740
      STOP                                                               3750
   76 CALL ERROR(506)                                                    3760
      STOP                                                               3770
      END                                                                3780
```

```
      SUBROUTINE ELIMZ(N1,N2,NBT,JNT,X,Y)                                 10
      DIMENSION X(1),Y(1),NBT(1),JNT(1)                                   20
C*********************************************************************   30
C*    THIS SUBROUTINE SUPPRESSES THE POINTS WHERE Y=0.0,                   40
C*    EXCEPT THE POINT PRECEDING THE THRESHOLD OF THE CROSS-SECTION        50
C*********************************************************************   60
      IF(N1.GT.1.OR.JNT(1).GT.2.OR.NBT(1).NE.N2) CALL ERROR(777)          70
C                                                                         80
C=====ELIMINATE ZERO POINTS AT THE END OF ARRAYS X AND Y                   90
C                                                                        100
      DO 1 II=1,N2                                                       110
      I=II                                                               120
      IF(Y(N2-I+1).NE.0.0)GOTO 2                                         130
    1 CONTINUE                                                           140
C-----ALL POINTS ARE ZERO. SAVE ONLY FIRST AND LAST ONES.                150
    7 X(2)=X(N2)                                                         160
      N2=2                                                               170
      GOTO 3                                                             180
C-----CROP END OF ARRAYS                                                 190
C-----THE FIRST POINT EQUAL ZERO AT THE END OF THE ARRAY AND THE LAST    200
C-----POINT EQUAL ZERO AT THE END OF THE ARRAY WILL BE SAVED.            210
    2 IF(I.LE.3)GOTO 4                                                   220
      N2=N2-I+3                                                          230
      X(N2)=X(N2+I-3)                                                    240
C                                                                        250
C=====ELIMINATE ZERO POINTS AT THE BEGINNIG OF ARRAYS X AND Y.           260
C                                                                        270
    4 DO 5 II=1,N2                                                       280
      I=II                                                               290
      IF(Y(I).NE.0.0)GOTO 6                                              300
    5 CONTINUE                                                           310
      GOTO 7                                                             320
```

```
C-----CROP LEADER ZEROS.                                                  330
    6 IF(I.LE.2) GOTO 3                                                   340
      I=I-2                                                               350
      N2=N2-I                                                             360
      DO 8 I2=1,N2                                                        370
      X(I2)=X(I2+I)                                                       380
    8 Y(I2)=Y(I2+I)                                                       390
C                                                                         400
C=====FINISHED                                                            410
C                                                                         420
    3 NBT(1)=N2                                                           430
      RETURN                                                              440
      END                                                                 450
```

```
      SUBROUTINE SUZY(LNU,ITAPE,BK,A,TAB,ITAB,NBT,JNT,X,Y,B)               10
      LOGICAL*4 SALINE,SECOND                                              20
      COMMON/INOUT/NO,NI,NT,MODE,NR1(3),KEDAVE,NR2(71),SALINE,NR3(2),      30
     *MATKDK,NR4(42),NBX,N1X,N2X,N,NBKX,NR5(10),EPS,SCREPS                 40
      COMMON/RECS/MAT,MF,MT,C1,C2,L1,L2,N1,N2,NS                           50
      COMMON/TAB/ENDTAB(25),KEDTAB(50),KEDR(240),NCNAME(39),NRIF3,NRIF4    60
      EQUIVALENCE (KEDR(1),KDR1(1)),(KEDR(124),KDR2(1))                    70
      INTEGER ENDTAB                                                       80
      REAL*8 AM241,NAM8                                                    90
      EQUIVALENCE (NAM1(1),NAME1,NAM8)                                    100
      DIMENSION A(1),TAB(1),ITAB(1),NAM1(2),BK(1),NBT(1),JNT(1),X(1),     110
     *Y(1),KDR1(123),KDR2(1),B(1),SCRCLA(10),NUMSCR(10)                   120
      CALL IRRECS(NBT,JNT,X,Y,B)                                          130
      NTP=15                                                              140
      NS=1                                                                150
      DATA EPSMAX/.02/,EPSINC/.001/                                       160
      MF=3                                                                170
      IADIM=NBKX                                                          180
C-----REACTION TYPES FROM FILE 3                                          190
      DO 1 I=1,26                                                         200
      PS1=EPS                                                             210
      IF(ITAPE.EQ.1)CALL WCONT(NTP,MODE,MAT,MF,NS)                        220
      IF(I.LE.25) MT=ENDTAB(I)                                            230
      IF(I.EQ.26) MT=91                                                   240
   21 RN=0.0                                                              250
      CALL RRECS(0,MF,MT,RN,RN,LNT,NBT,JNT,X,Y)                           260
      IF(LNT.EQ.1)GOTO 1                                                  270
      IF(KEDAVE.EQ.0.AND.(MT.EQ.4.OR.MT.EQ.91)) GOTO 132                  280
      CALL CHGINT(PS1,N1,N2,NBT,JNT,X,Y,N1X,N2X,&20)                      290
      CALL ELIMZ(N1,N2,NBT,JNT,X,Y)                                       300
  132 CALL ADDPNT(X,Y,NBT,JNT,MT)                                         310
      IF(KEDAVE.EQ.0.AND.(MT.EQ.4.OR.MT.EQ.91)) GOTO 14                   320
      XL=X(1)                                                             330
      XH=X(N2)                                                            340
      CALL CROP(XL,XH,SCREPS,.TRUE.,EPS,SCRCLA,NUMSCR,LOF,NBT,JNT,X,Y)    350
      IF(LOF.NE.0) CALL ERROR(777)                                        360
      GOTO 14                                                             370
   20 PS1=PS1+EPSINC                                                      380
      IF(PS1-EPSMAX)21,21,22                                              390
   22 CALL ERROR(850)                                                     400
   14 IF(I.EQ.26) GOTO 40                                                 410
      NAME1=KEDTAB(2*I-1)                                                 420
      NAME2=KEDTAB(2*I)                                                   430
      GOTO 41                                                             440
   40 CALL XTEXT(4HSGIZ,NAME1)                                            450
      CALL XTEXT(4HC   ,NAME2)                                            460
   41 CONTINUE                                                            470
      WRITE(NO,2)MT,PS1,SCREPS,EPS                                        480
    2 FORMAT(1H1,'REACTION TYPE',I4,' AFTER CHGINT',8X,                   490
     *'PRECISION EPS=',F5.4,5X,'EPSMIN=',F5.4,'  EPSMAX=',F5.4)           500
      IF(KEDAVE.EQ.0.AND.(MT.EQ.4.OR.MT.EQ.91)) GOTO 134                  510
      WRITE(NO,133) (SCRCLA(J),J=1,10),(NUMSCR(J),J=1,10)                 520
  133 FORMAT(' SCRATCHCLASS (IN %) 0.00',10('<  <=',2PF4.2)/               530
     *' NUMBER OF POINTS   ',10I9)                                        540
      IF(KEDAVE.EQ.0.AND.(MT.EQ.4.OR.MT.EQ.91)) GOTO 134                  550
      CALL DZMSCH(X,Y,NBT,JNT,NDZ,EPS,SCRCLA,NUMSCR)                      560
      IF(NDZ.GT.0) WRITE(NO,30) NDZ,(SCRCLA(J),J=1,9),(NUMSCR(J),J=1,10)  570
   30 FORMAT(' TOTALLY',I7,' POINTS HAVE HAD X-VALUES (COMPARED WITH OTH  580
     *ER POINTS) CLOSER THAN (RELATIVLY) 5.0E-6.'/                        590
     *' THESE POINTS WERE SCRATCHED AND THE NEIGHBOURING X- AND Y-VALUES   600
     * HAVE BEEN REPLACED BY AVERAGE VALUES.'/                            610
     *' THESE POINTS COULD HAVE BEEN PREDICTED WITH THE FOLLOWING PRECIS   620
     *IONS:'/                                                            630
     *' CLASSES (IN %)    0.0',9('<  <=',2PF4.1),'<  <=UNENDLICH'/        640
     *' NUMBER OF POINTS',10I9)                                          650
  134 IF(MT.EQ.1.OR.MT.EQ.2) CALL SUPNEG(MT,N1,N2,NBT,JNT,X,Y)            660
      CALL PRIZEI                                                         670
      NAM1(2)=NAME2                                                       680
      CALL A8FORM(NO,NAM8)                                                690
      C1=0.0                                                              700
      C2=0.0                                                              710
      L1=0                                                                720
      L1=0                                                                730
      CALL WREC(3,NO,4)                                                   740
      IF(ITAPE.EQ.1)CALL WREC(3,NTP,3)                                    750
      K=1                                                                 760
      DO 3 J=1,N2                                                         770
      BK(K)=X(J)                                                          780
      BK(K+1)=Y(J)                                                        790
      K=K+2                                                               800
    3 CONTINUE                                                            810
      N=N2*2                                                              820
      CALL WDA2(MATKDK,1,NAME1,NAME2)                                     830
    1 CONTINUE                                                            840
C-----INELASTIC EXCITATION CROSS SECTIONS                                 850
      NW=NRIF3*4+1                                                        860
      IA=NW                                                               870
      NINIT=1                                                             880
      NWS=NW                                                              890
      NRS=1                                                               900
      SECOND=.FALSE.                                                      910
      CALL XTEXT(4HSGIZ,NAME1)                                            920
      CALL XTEXT(4H    ,NAME2)                                            930
```

```
      DO 7 MT=51,90                                                          940
      PS1=EPS                                                                950
  131 RN=0.0                                                                 960
      CALL RRECS(0,MF,MT,EXC,RN,LNT,NBT,JNT,X,Y)                             970
      IF(LNT.EQ.0)GOTO 12                                                    980
      IF(SECOND) GOTO 13                                                     990
      IF(SALINE) GOTO 7                                                     1000
      GOTO 8                                                                1010
  130 PS1=PS1+EPSINC                                                        1020
      IF(PS1.LE.EPSMAX) GOTO 131                                            1030
      CALL ERROR(850)                                                       1040
   12 IF(SECOND) GOTO 121                                                   1050
      SECOND=.TRUE.                                                         1060
  121 IF(KEDAVE.EQ.0) GOTO 135                                              1070
      CALL CHGINT(PS1,N1,N2,NBT,JNT,X,Y,N1X,N2X,&130)                       1080
      CALL ELIMZ(N1,N2,NBT,JNT,X,Y)                                         1090
  135 CALL ADDPNT(X,Y,NBT,JNT,MT)                                           1100
      IF(KEDAVE.EQ.0) GOTO 123                                              1110
      XL=X(1)                                                               1120
      XH=X(N2)                                                              1130
      CALL CROP(XL,XH,SCREPS,.TRUE.,EPS,SCRCLA,NUMSCR,LOF,NBT,JNT,X,Y)      1140
      IF(LOF.NE.0) CALL ERROR(777)                                          1150
  123 WRITE(NO,2) MT,PS1,SCREPS,EPS                                         1160
      IF(KEDAVE.EQ.0) GOTO 136                                              1170
      WRITE(NO,133) (SCRCLA(J),J=1,10),(NUMSCR(J),J=1,10)                   1180
      CALL DZMSCH(X,Y,NBT,JNT,NDZ,EPS,SCRCLA,NUMSCR)                        1190
      IF(NDZ.GT.0) WRITE(NO,30) NDZ,(SCRCLA(J),J=1,9),(NUMSCR(J),J=1,10)    1200
  136 CALL PRIZEI                                                           1210
      CALL XTEXT(4H 00 ,NAM1(2))                                            1220
      IH1=MT/10                                                             1230
      NAM1(2)=NAM1(2)+IH1*65536+(MT-IH1*10)*256                             1240
      CALL A8FORM(NO,NAM8)                                                  1250
      C1=0.0                                                                1260
      C2=-EXC                                                               1270
      L1=0                                                                  1280
      L2=0                                                                  1290
      CALL WREC(3,NO,4)                                                     1300
      DO 9 J=1,N2                                                           1310
      A(IA)=X(J)                                                            1320
      A(IA+1)=Y(J)                                                          1330
      IA=IA+2                                                               1340
      IF(IA.GE.IADIM)CALL ERRCR(515)                                        1350
    9 CONTINUE                                                              1360
      TAB(NINIT)=EXC                                                        1370
      ITAB(NINIT+1)=N2                                                      1380
      ITAB(NINIT+2)=NRS                                                     1390
      ITAB(NINIT+3)=NWS                                                     1400
      NINIT=NINIT+4                                                         1410
      NRS=(IA-1)/880+1                                                      1420
      NWS=MOD(IA-1,880)+1                                                   1430
    7 CONTINUE                                                              1440
   13 IA=IA-1                                                               1450
      CALL XTEXT(4H     ,NAME2)                                             1460
      N=IA                                                                  1470
      CALL WDA2(MATKDK,1,NAME1,NAME2)                                       1480
      WRITE(NO,10)NAME1,NAME2                                               1490
   10 FORMAT(1H1,2A4,' REACTION TYPE TABLE'/)                               1500
      NWM1=NW-1                                                             1510
      WRITE(NO,11)(A(I),I=1,NWM1)                                           1520
   11 FORMAT(1X,1PE15.5,3I8)                                                1530
C-----NUE DATA TYPE                                                         1540
    8 PS1=EPS                                                               1550
   28 RN=0.0                                                                1560
      CALL RRECS(0,1,452,RN,RN,LNT,NBT,JNT,X,Y)                             1570
      IF(LNT.EQ.0)GOTO 5                                                    1580
      IF(ITAPE.EQ.1)CALL WCONT(NTP,MODE,MAT,0,NS)                           1590
      GOTO 6                                                                1600
    5 CALL CHGINT(PS1,N1,N2,NBT,JNT,X,Y,N1X,N2X,&23)                        1610
      XL=X(1)                                                               1620
      XH=X(N2)                                                              1630
      CALL CROP(XL,XH,SCREPS,.TRUE.,EPS,SCRCLA,NUMSCR,LOF,NBT,JNT,X,Y)      1640
      IF(LOF.NE.0) CALL ERROR(777)                                          1650
      GOTO 24                                                               1660
   23 PS1=PS1+EPSINC                                                        1670
      IF(PS1-EPSMAX)28,28,25                                                1680
   25 CALL ERROR(851)                                                       1690
   24 CALL XTEXT(4HNUE ,NAME1)                                              1700
      CALL XTEXT(4H     ,NAME2)                                             1710
      MF=3                                                                  1720
      MT=452                                                                1730
      WRITE(NO,2)MT,PS1,SCREPS,EPS                                          1740
      WRITE(NO,133) (SCRCLA(J),J=1,10),(NUMSCR(J),J=1,10)                   1750
      CALL DZMSCH(X,Y,NBT,JNT,NDZ,EPS,SCRCLA,NUMSCR)                        1760
      IF(NDZ.GT.0) WRITE(NO,30) NDZ,(SCRCLA(J),J=1,9),(NUMSCR(J),J=1,10)    1770
      CALL PRIZEI                                                           1780
      NAM1(2)=NAME2                                                         1790
      CALL A8FORM(NO,NAM8)                                                  1800
      C1=0.0                                                                1810
      C2=0.0                                                                1820
      L1=0                                                                  1830
      L2=0                                                                  1840
      CALL WREC(3,NO,4)                                                     1850
      K=1                                                                   1860
      DO 4 J=1,N2                                                           1870
      BK(K)=X(J)                                                            1880
      BK(K+1)=Y(J)                                                          1890
      K=K+2                                                                 1900
    4 CONTINUE                                                              1910
      N=N2*2                                                                1920
      CALL WDA2(MATKDK,1,NAME1,NAME2)                                       1930
      IF(ITAPE.EQ.1)CALL WREC(3,NTP,3)                                      1940
    6 IF(ITAPE.EQ.1)CALL WCONT(NTP,MODE,0,0,NS)                             1950
      RETURN                                                                1960
      END                                                                   1970
```

```
      SUBROUTINE DZMSCH(X,Y,NBT,JNT,NDZ,EPS,SCRCLA,NUMSCR)                10
      DIMENSION X(1),Y(1),NBT(1),JNT(1),SCRCLA(10),NUMSCR(10)            20
      COMMON/RECS/ NR1(7),N1,N2                                          30
C     N PUNKTE, DEREN X-WERTE AUF 6 STELLEN UEBEREINSTIMMEN,             40
C     WERDEN FOLGENDERMASSEN ZUSAMMENGEFASST:                            50
C     XNEU=1/N*(X1+X2+...XN); YNEU=1/N*(Y1+Y2+...+YN) .                  60
C                                                                        70
      IF(N1.GT.1.OR.JNT(1).GT.2.OR.NBT(1).NE.N2) CALL ERROR(777)         80
C     BILDUNG DER SCRATCHCLASSEN.                                        90
      SCRCLA(1)=EPS                                                     100
      HI1=(1.0-EPS)/9.0                                                 110
      DO 1 I=2,9                                                        120
    1 SCRCLA(I)=SCRCLA(1)+(I-1)*HI1                                     130
      DO 2 I=1,10                                                       140
    2 NUMSCR(I)=0                                                       150
      NDZ=0                                                             160
      I=1                                                               170
  305 IF((I+1).GT.N2) GOTO 320                                          180
      J=I+1                                                             190
  307 IF(J.GT.N2) GOTO 306                                              200
      IF((X(J)-X(I))/X(J).GE.5.0E-6) GOTO 306                           210
      J=J+1                                                             220
      GOTO 307                                                          230
  306 IF(J.EQ.(I+1)) GOTO 308                                           240
C     MIT WELCHER GENAUIGKEIT WAREN DIE PUNKTE I+1,...,J-1 ZWISCHEN      250
C     DEN PUNKTEN I UND MIN(J,N2) ZU INTERPOLIEREN. (SIEHE CROP).        260
      IP1=I+1                                                           270
      JM1=J-1                                                           280
      MINJN2=MIN(J,N2)                                                  290
      DO 10 K=IP1,JM1                                                   300
      CALL TERP1(X(I),Y(I),X(MINJN2),Y(MINJN2),X(K),YP,2)               310
      EPSAKT=ABS((Y(K)-YP)/Y(K))                                        320
      DO 11 KK=1,10                                                     330
      IF(KK.EQ.10) GOTO 12                                              340
      IF(EPSAKT.LE.SCRCLA(10-KK)) GOTO 11                               350
   12 NUMSCR(11-KK)=NUMSCR(11-KK)+1                                     360
      GOTO 10                                                           370
   11 CONTINUE                                                          380
   10 CONTINUE                                                          390
C     MITTELWERT DER PUNKTE Y(I), Y(I+1), ..... Y(J-1) BILDEN           400
      M=J-I                                                             410
      NDZ=NDZ+(M-1)                                                     420
      X1=0.0                                                            430
      Y1=0.0                                                            440
      DO 309 K=1,M                                                      450
      X1=X1+X(I+K-1)                                                    460
  309 Y1=Y1+Y(I+K-1)                                                    470
      X(I)=1.0/M*X1                                                     480
      Y(I)=1.0/M*Y1                                                     490
C     DELETEN DER PUNKTE X(I+1),Y(I+1), X(I+2),Y(I+2), ... X(J-1),Y(J-1) 500
      IF((J-1).NE.N2) GOTO 310                                          510
C     J-1=N2                                                            520
      N2=I                                                              530
      NBT(1)=N2                                                         540
      GOTO 320                                                          550
  310 N=I+1                                                             560
      M=M-1                                                             570
      L=N2-M                                                            580
      DO 311 K=N,L                                                      590
      X(K)=X(K+M)                                                       600
  311 Y(K)=Y(K+M)                                                       610
      N2=N2-M                                                           620
      NBT(1)=N2                                                         630
  308 I=I+1                                                             640
      GOTO 305                                                          650
  320 CONTINUE                                                          660
      RETURN                                                            670
      END                                                               680
```

```
      SUBROUTINE ADDPNT(X,Y,NBT,JNT,MT)                                  10
C     BEI REAKTIONEN MIT EINER SCHWELLE WERDEN DAS VORHANDENSEIN         20
C     VON ZWEI NULLEN, EINE BEI 1.0E-5 UND EINE BEI DER SCHWELLE,        30
C     NOETIGENFALLS DURCH ERZEUGUNG SICHERGESTELLT.                      40
      LOGICAL*4 LT1                                                      50
      DIMENSION X(1),Y(1),NBT(1),JNT(1)                                  60
      COMMON/RECS/ NR1(7),N1,N2                                          70
      COMMON/INOUT/ NO,NR2(126),N2X                                      80
C     NUR LINEARE ODER KONSTANTE INTERPOLATION UND EIN ENERGIEBEREICH    90
C     ERLAUBT.                                                          100
      IF(N1.NE.1.OR.NBT(1).NE.N2.OR.JNT(1).GT.2) CALL ERROR(777)        110
C     SCHWELLREAKTION?                                                  120
      IF((ABS(X(1)-1.0E-5)/X(1).LE.1.0E-6).OR.(ABS(Y(1)).GT.1.0E-20))   130
     *RETURN                                                            140
C     Y(1).GT.0?                                                        150
      LT1=.FALSE.                                                       160
      IF(Y(1).EQ.0.0) GOTO 2                                            170
      LT1=.TRUE.                                                        180
      Y(1)=0.0                                                          190
      WRITE(NO,100) MT                                                  200
  100 FORMAT('0$$$ FOR ENDF/B TYPE MF=3 MT=',I3)                        210
      WRITE(NO,101)                                                     220
  101 FORMAT('+',31X,' Y(1) HAS BEEN SET TO ZERO.')                     230
    2 N2=N2+1                                                           240
      IF(N2.GT.N2X) CALL ERROR(105)                                     250
      NBT(1)=N2                                                         260
      DO 1 I=2,N2                                                       270
      X(N2+2-I)=X(N2+1-I)                                               280
    1 Y(N2+2-I)=Y(N2+1-I)                                               290
      X(1)=1.0E-5                                                       300
      Y(1)=0.0                                                          310
      IF(.NOT.LT1) WRITE(NO,100) MT                                     320
      IF(LT1)      WRITE(NO,102)                                        330
  102 FORMAT(' $$$                  ADDITIONALLY')                      340
      WRITE(NO,103)                                                     350
  103 FORMAT('+',31X,' A POINT X=1.0E-5EV, Y=0.0BARN HAS BEEN ADDED.')  360
      RETURN                                                            370
      END                                                               380
```

```
      SUBROUTINE SUPNEG(MT,N1,N2,NBT,JNT,X,Y)                          10
C     FUER MT=1 UND MT=2 WERDEN NEGATIVE Y-BEREICHE                    20
C     DIAGNOSTIZIERT UND BESEITIGT.                                    30
      INTEGER*4 ELINEG                                                 40
      DIMENSION X(1),Y(1),NBT(1),JNT(1),TEXT(2)                        50
      LOGICAL*4 NEG,PREVAL                                             60
      COMMON/INOUT/ NO,NR1(152),ELINEG                                 70
      DATA TEXT/'SGT ','SGN '/                                         80
      IF(MT.NE.1.AND.MT.NE.2) CALL ERROR(777)                          90
      NEG=.FALSE.                                                     100
      PREVAL=.FALSE.                                                  110
      I=1                                                             120
C     LOOP UEBER ALLE ENERGIEPUNKTE.                                  130
    4 IF(Y(I).GE.0.0) GOTO 2                                          140
      IF(.NOT.PREVAL) INEGAN=I                                        150
      IF(.NOT.NEG) WRITE(NO,100) TEXT(MT)                             160
  100 FORMAT('ONEGATIVE VALUES FOUND FOR CROSS-SECTION-TYPE ',A4,'.')  170
      IF(.NOT.PREVAL) WRITE(NO,101)                                   180
  101 FORMAT('ONUMBER  ENERGY ( EV )        CROSS SECTION ( BARN )')  190
      WRITE(NO,102) I,X(I),Y(I)                                       200
  102 FORMAT(' ',I6,2X,1PE15.7,4X,1PE15.7)                            210
      NEG=.TRUE.                                                      220
      PREVAL=.TRUE.                                                   230
      GOTO 1                                                          240
C     BESEITIGEN DER NEGATIVEN BEREICHE VON INEGAN BIS INEGEN         250
    2 CONTINUE                                                        260
      IF(.NOT.PREVAL) GOTO 1                                          270
      INEGEN=I-1                                                      280
      IF(INEGEN.LT.INEGAN) CALL ERROR(777)                            290
C     BESEITIGUNG ERFOLGT NUR AUF WUNSCH.                             300
      IF(ELINEG.NE.1) GOTO 1                                          310
      IF(INEGAN.LE.1) CALL ERROR(777)                                 320
      WRITE(NO,103) X(INEGAN),X(INEGEN),X(INEGAN-1),Y(INEGAN-1),      330
     *X(INEGEN+1),Y(INEGEN+1)                                         340
  103 FORMAT(' THE PREVIOUS VALUES FROM ',1PE15.7,' EV  TO ',         350
     *1PE15.7,' EV  WILL BE SCRATCHED.'/                              360
     *' POINT BEFORE INTERVAL:',1PE15.7,' EV, ',1PE15.7,' BARN.'/     370
     *' POINT  AFTER INTERVAL:',1PE15.7,' EV, ',1PE15.7,' BARN.')     380
C     DIE PUNKTE INEGAN+1 BIS N2 WERDEN AUF INEGAN BIS                390
C     N2-(INEGEN-INEGAN+1) GELEGT.                                    400
      IANZ=N2-INEGEN                                                  410
      DO 3 II=1,IANZ                                                  420
      X(INEGAN+II-1)=X(INEGEN+II)                                     430
    3 Y(INEGAN+II-1)=Y(INEGEN+II)                                     440
      IF(N1.NE.1.OR.JNT(1).GT.2.OR.NBT(1).NE.N2) CALL ERROR(777)      450
      N2=N2-(INEGEN-INEGAN+1)                                         460
      NBT(1)=N2                                                       470
      PREVAL=.FALSE.                                                  480
    1 I=I+1                                                           490
      IF(I.LE.N2) GOTO 4                                              500
  333 RETURN                                                          510
      END                                                             520
```

```
      SUBROUTINE ISABEL(MATP,IF4,ITAB,TAB,CLEG,NBT,JNT,X,Y,B,          10
     *A,ITAB2,TAB2)                                                    20
      COMMON/INOUT/NO,NI,NT,MODE,NR1(78),MATKOK,NR2(45),N,NBKX,JX,IADIM 30
      COMMON/RECS/MAT,MF,MT,C1,C2,L1,L2,N1,N2                          40
      COMMON/TAB/ENDTAB(25),KEDTAB(50),KEDR(240),NCNAME(39),NRIF3,NRIF4 50
      EQUIVALENCE (KEDR(1),KDR1(1)),(KEDR(124),KDR2(1))                60
      INTEGER ENDTAB                                                   70
      DIMENSION CLEG(1),ITAB(1),TAB(1),ITAB2(1),TAB2(1),               80
     *  NBT(1),JNT(1),X(1),Y(1),B(1),A(1),ISATZ(400),NAUX(2),          90
     *KDR1(123),KDR2(1)                                               100
      REAL*4 LEGP                                                     110
      REAL*8 NAM8                                                     120
      EQUIVALENCE (NAUX(1),NAM8)                                      130
      IDLEG=NBKX                                                      140
      NXMAX=NBKX                                                      150
      NF=10                                                           160
      NF1=11                                                          170
      NINIT2=1                                                        180
      REWIND NF                                                       190
      WRITE(NO,1)                                                     200
    1 FORMAT(1H1,20X,'FILE 4  -  ANGULAR DISTRIBUTIONS OF SECONDARY NEUT 210
     *RONS'//21X,54('*')///)                                          220
      CALL PRIZEI                                                     230
      CALL SEARCH(NT,MODE,MATP,4,2,LNT)                               240
      IF(LNT.EQ.0)GOTO 40                                             250
      WRITE(NO,2)                                                     260
    2 FORMAT(1HO,'ELASTIC ANGULAR DISTRIBUTIONS NOT INCLUDED IN THE ENDF 270
     */B FILE')                                                       280
      BACKSPACE NT                                                    290
      GOTO 3                                                          300
   40 LVT=L1                                                          310
      LTT=L2                                                          320
      IF(LTT.EQ.2)GOTO 4                                              330
C                                                                     340
C=====LEGENDRE POLYNOMIAL COEFFICIENTS.    LTT=1                      350
C                                                                     360
      IF(LVT.EQ.0)GOTO 5                                              370
C     TRANSFORMATION MATRIX GIVEN LVT=1                               380
      CALL RREC(2,NT,MODE,-1.0,B)                                     390
      GOTO 6                                                          400
C     TRANSFORMATION MATRIX NOT GIVEN LVT=0                           410
    5 CALL RREC(1,NT,MODE,-1.0)                                       420
    6 LCT=L2                                                          430
      CALL RREC(4,NT,MODE,-1.0,NBT,JNT)                               440
      NE=N2                                                           450
      IF(LCT-1)9,9,10                                                 460
C     LCT=1   LAB SYSTEM                                              470
    9 IKL=25                                                          480
      INCL=9                                                          490
      IKS=1                                                           500
      INCS=1                                                          510
```

```
      GOTO 11                                                           520
C     LCT=2  C.M. SYSTEM                                                530
   10 IKL=28                                                            540
      INCL=10                                                           550
      IKS=4                                                             560
      INCS=2                                                            570
   11 NCNAME(INCL)=NE                                                   580
      NCNAME(INCS)=NE                                                   590
      IKL2=IKL+2                                                        600
      NAUX(1)=KDR2(IKL)                                                 610
      NAUX(2)=KDR2(IKL+1)                                               620
      CALL A8FORM(NC,NAM8)                                              630
      NAUX(1)=KDR2(IKS)                                                 640
      NAUX(2)=KDR2(IKS+1)                                               650
      CALL A8FORM(NO,NAM8)                                              660
      WRITE(NO,13)(KDR2(I),I=IKL,IKL2),NE                               670
   13 FORMAT(1HO,'THE LEGENDRE POLYNOMIAL COEFFICIENTS FOR REACTION TYPE 680
     *',2X,2A4,2X,I6,2X,'ARE GIVEN AT',I6,' ENERGY POINTS'/)            690
      CALL A8FORM(NO,'LEGENDRE')                                        700
      CALL A8FORM(NO,'COEFFIC.')                                        710
      NINIT=1                                                           720
      NWORD=NE*5+1                                                      730
      NWORD2=NE*4+1                                                     740
      NRIF4=NRIF4+2                                                     750
      NLEGI=NWORD                                                       760
      IA=NWORD2                                                         770
      DO 7 I=1,NE                                                       780
      CALL RREC(2,NT,MODE,-1.0,B)                                       790
      E=C2                                                              800
   61 IF(B(N1).GT.0.)GOTO 60                                            810
      IF(N1.EQ.1)GOTO 60                                                820
      N1=N1-1                                                           830
      GOTO 61                                                           840
   60 NWL=MOD(NLEGI-1,880)+1                                            850
      NRL=(NLEGI-1)/880+1                                               860
      DO 8 N=1,N1                                                       870
      IF(NLEGI.GE.IDLEG)CALL ERROR(700)                                 880
      CLEG(NLEGI)=N                                                     890
      CLEG(NLEGI+1)=B(N)                                                900
    8 NLEGI=NLEGI+2                                                     910
      TAB(NINIT)=E                                                      920
      TAB(NINIT+1)=N1                                                   930
      ITAB(NINIT+2)=N1                                                  940
      ITAB(NINIT+3)=NRL                                                 950
      ITAB(NINIT+4)=NWL                                                 960
      NINIT=NINIT+5                                                     970
      WRITE(NO,15)                                                      980
   15 FORMAT(1HO//5X,' ENERGY', 7X,'LEGENDRE COEFFICIENTS'/)            990
      WRITE(NO,14)E,(B(N),N=1,N1)                                      1000
   14 FORMAT(1X,1P7E15.5/(16X,1P6E15.5))                               1010
      WRITE(NO,52)                                                     1020
   52 FORMAT(1HO)                                                      1030
C CALCUL DES SIGMA A PARTIR DES POLYNOMES                              1040
      X(1)=-1.0                                                        1050
      NPMAX=81                                                         1060
      X(NPMAX/2+1)=0.0                                                 1070
      DX=1.0/(NPMAX/2)                                                 1080
      DO 45 K=2,NPMAX                                                  1090
      IF(K.EQ.(NPMAX/2+1))GOTO 45                                      1100
      X(K)=X(K-1)+DX                                                   1110
   45 CONTINUE                                                         1120
      X(NPMAX)=1.                                                      1130
      NRS=(IA-1)/880+1                                                 1140
      NWS=MOD(IA-1,880)+1                                              1150
      DO 46 K=1,NPMAX                                                  1160
      Y(K)=LEGP(X(K),N1,B)/6.283184                                    1170
      A(IA)=X(K)                                                       1180
      A(IA+1)=Y(K)                                                     1190
      IA=IA+2                                                          1200
      IF(IA.GE.IADIM) CALL ERROR(799)                                  1210
   46 CONTINUE                                                         1220
      N2=NPMAX                                                         1230
      N1=1                                                             1240
      NBT(1)=NPMAX                                                     1250
      JNT(1)=2                                                         1260
      CALL WREC(3,NC,4)                                                1270
      GOTO 22                                                          1280
    7 CONTINUE                                                         1290
      GOTO 36                                                          1300
C                                                                      1310
C                                                                      1320
C=====TABULATED DISTRIBUTION PROBABILITIES  LTT=2                      1330
C                                                                      1340
C                                                                      1350
    4 IF(LVT.EQ.0)GOTO 16                                              1360
C     TRANSFORMATION MATRIX GIVEN  LVT=1                               1370
      CALL RREC(2,NT,MODE,-1.0,B)                                      1380
      GOTO 17                                                          1390
C     TRANSFORMATION MATRIX NOT GIVEN  LVT=0                           1400
   16 CALL RREC(1,NT,MODE,-1.0)                                        1410
   17 LCT=L2                                                           1420
      CALL RREC(4,NT,MODE,-1.0,NBT,JNT)                                1430
      NE=N2                                                            1440
      IF(LCT-1)18,18,19                                                1450
   18 IKS=1                                                            1460
      INCS=1                                                           1470
      GOTO 20                                                          1480
   19 IKS=4                                                            1490
      INCS=2                                                           1500
   20 NCNAME(INCS)=NE                                                  1510
      IKS2=IKS+2                                                       1520
      NAUX(1)=KDR2(IKS)                                                1530
      NAUX(2)=KDR2(IKS+1)                                              1540
      CALL A8FORM(NC,NAM8)                                             1550
      WRITE(NO,21)(KDR2(I),I=IKS,IKS2),NE                              1560
   21 FORMAT(1HO,'THE ANGULAR DISTRIBUTION PROBABILITIES FOR REACTION TY 1570
     *PE',2X,2A4,2X,I6,2X,' ARE GIVEN AT',I6,' ENERGY POINTS'/)        1580
      CALL A8FORM(NO,'  POINT ')                                       1590
      CALL A8FORM(NO,'  WISE  ')                                      1600
      NWORD2=NE*4+1                                                    1610
```

```
      NRIF4=NRIF4+1                                                     1620
      NPMAX=0                                                           1630
      DO 41 I=1,NE                                                      1640
      CALL RREC(3,NT,MODE,-1.0,NBT,JNT,X,Y)                            1650
      IF(NPMAX.GE.N2)GOTO 42                                            1660
      NPMAX=N2                                                          1670
      IF(N2.GT.NXMAX/2) CALL ERROR(700)                                 1680
      DO 25 KI=1,N2                                                     1690
   25 TAB(KI)=X(KI)                                                     1700
   42 CONTINUE                                                          1710
      CALL WREC(3,NF,2)                                                 1720
   41 CONTINUE                                                          1730
      REWIND NF                                                         1740
      IA=NWORD2                                                         1750
      WRITE(NO,28)                                                      1760
   28 FORMAT('1')                                                       1770
      DO 24 I=1,NE                                                      1780
      CALL RREC(3,NF,2,-1.0,NBT,JNT,X,Y)                               1790
      E=C2                                                              1800
      DO 26 J=1,NPMAX                                                   1810
      CALL ITAB1(TAB(J),TAB(J+NXMAX/2))                                 1820
   26 CONTINUE                                                          1830
      DO 27 J=1,NPMAX                                                   1840
      X(J)=TAB(J)                                                       1850
      Y(J)=TAB(J+NXMAX/2)/6.283184                                      1860
   27 CONTINUE                                                          1870
      N2=NPMAX                                                          1880
      N1=1                                                              1890
      NBT(1)=NPMAX                                                      1900
      JNT(1)=2                                                          1910
      CALL WREC(3,NC,4)                                                 1920
      NRS=(IA-1)/880+1                                                  1930
      NWS=MOD(IA-1,880)+1                                               1940
      DO 29 J=1,NPMAX                                                   1950
      A(IA)=X(J)                                                        1960
      A(IA+1)=Y(J)                                                      1970
      IA=IA+2                                                           1980
      IF(IA.GE.IADIM)CALL ERROR(703)                                    1990
   29 CONTINUE                                                          2000
      GOTO 22                                                           2010
   24 CONTINUE                                                          2020
   36 NLEGI=NLEGI-1                                                     2030
      IA=IA-1                                                           2040
      IF(IF4.EQ.1)GOTO 47                                               2050
      READ(NF1'1)(ISATZ(I),I=1,400)                                     2060
      IR=ISATZ(399)                                                     2070
      IW=ISATZ(400)                                                     2080
      GOTO 48                                                           2090
   47 IR=2                                                              2100
      IW=1                                                              2110
   48 IF(LTT.EQ.2)GOTO 37                                               2120
      ISATZ(IW)=MATKDK                                                  2130
      ISATZ(IW+1)=1                                                     2140
      ISATZ(IW+2)=KDR2(IKL)                                             2150
      ISATZ(IW+3)=KDR2(IKL+1)                                           2160
      ISATZ(IW+4)=IR                                                    2170
      ISATZ(IW+5)=NLEGI                                                 2180
      IW=IW+6                                                           2190
      IF(IW.GE.390)CALL ERROR(701)                                      2200
      WRITE(NF1'IR)(CLEG(I),I=1,NLEGI)                                  2210
      IR=IR+NLEGI/400+1                                                 2220
      WRITE(NO,51)KDR2(IKL),KDR2(IKL+1)                                 2230
   51 FORMAT(1H1,2A4,'REACTION TYPE TABLE'/)                            2240
      NWM1=NWORD-1                                                      2250
      WRITE(NO,53)(CLEG(I),I=1,NWM1)                                    2260
   53 FORMAT(1X,1P2E15.5,3I8)                                           2270
   37 ISATZ(IW+2)=KDR2(IKS)                                             2280
      ISATZ(IW+3)=KDR2(IKS+1)                                           2290
      ISATZ(IW+4)=IR                                                    2300
      ISATZ(IW+5)=IA                                                    2310
      ISATZ(IW)=MATKDK                                                  2320
      ISATZ(IW+1)=1                                                     2330
      IW=IW+6                                                           2340
      IF(IW.GE.390)CALL ERROR(701)                                      2350
      WRITE(NF1'IR)(A(I),I=1,IA)                                        2360
      IR=IR+IA/400+1                                                    2370
      ISATZ(399)=IR                                                     2380
      ISATZ(400)=IW                                                     2390
      WRITE(NF1'1)(ISATZ(I),I=1,400)                                    2400
      NM1=NWORD2-1                                                      2410
      WRITE(NO,51)KDR2(IKS),KDR2(IKS+1)                                 2420
      WRITE(NO,43)(A(I),I=1,NM1)                                        2430
   43 FORMAT(1X,1PE15.5,3I8)                                            2440
      GOTO 3                                                            2450
C     EXTENDED DO LOOP OF 7 AND 24                                      2460
   22 TAB2(NINIT2)=E                                                    2470
      ITAB2(NINIT2+1)=N2                                                2480
      ITAB2(NINIT2+2)=NRS                                               2490
      ITAB2(NINIT2+3)=NWS                                               2500
      NINIT2=NINIT2+4                                                   2510
      IF(LTT-1)7,7,24                                                   2520
C                                                                       2530
C=====INELASTIC ANGULAR DISTRIBUTIONS                                   2540
C                                                                       2550
    3 CONTINUE                                                          2560
      RETURN                                                            2570
      END                                                               2580
```

```
      SUBROUTINE ITAB1(XP,YP)                                           10
C----------CHANGED OCTOBRE 1973 BY E.STEIN.                            20
      COMMON/RECS/MAT,MF,MT,C1,C2,L1,L2,N1,N2                           30
      DIMENSION NBT(1),JNT(1),X(1),Y(1)                                 40
      YP=0.0                                                            50
      IF(XP.LT.X(1))GOTO 50                                             60
      DO 10 N=2,N2                                                      70
      IF(XP.LT.X(N))GOTO 20                                             80
   10 CONTINUE                                                          90
```

```
      IF(XP.GT.X(N2))GOTO 50                                              100
      IF(X(N2-1).NE.X(N2)) GOTO 1                                         110
      YP=Y(N2)                                                            120
      GOTO 50                                                             130
    1 N=N2                                                                140
   20 DO 30 M=1,N1                                                        150
      IF(N.LE.NBT(M))GOTO 40                                              160
   30 CONTINUE                                                            170
      CALL ERROR(900)                                                     180
   40 I=JNT(M)                                                            190
      CALL TERP1(X(N-1),Y(N-1),X(N),Y(N),XP,YP,I)                         200
   50 RETURN                                                              210
      ENTRY EITAB1(NBT,JNT,X,Y)                                           220
      GOTO 50                                                             230
      END                                                                 240
```

```
      SUBROUTINE WCONT(NT,MODE,MAT,MF,NS)                                  1C
C=====WRITE END RECORD ON TAPE NT                                          2C
      IF(NT.LE.0)GOTO 20                                                   3C
      MT=0                                                                 4C
      C1=0.0                                                               5C
      C2=0.0                                                               6C
      L1=0                                                                 7C
      L2=0                                                                 8C
      N1=0                                                                 9C
      N2=0                                                                10C
      GOTO(1,1,2),MODE                                                    11C
C-----MODE=1                                                              12C
    1 WRITE(NT)MAT,MF,MT,C1,C2,L1,L2,N1,N2                                13C
      GOTO 20                                                             14C
C-----MODE=3                                                              15C
    2 WRITE(NT,3)MAT,MF,MT,NS                                             16C
    3 FORMAT(66X,I4,I2,I3,I5)                                             17C
   20 RETURN                                                              18C
      END                                                                 19C
```

```
      SUBROUTINE BETSY(MIS,Y)                                              1C
      COMMON/RECS/NR2(8),N2                                                2C
      DIMENSION Y(1)                                                       3C
      COMMON/INOUT/NR1(8),ABIS(20)                                         4C
      DO 1 I=1,N2                                                          5C
      Y(I)=Y(I)*ABIS(MIS)                                                  6C
    1 CONTINUE                                                             7C
      RETURN                                                               8C
      END                                                                  9C
```

```
      REAL FUNCTION LEGP(X,NL,CON)                                         1C
      DIMENSION CON(1)                                                     2C
      LEGP=0.5+1.5*CON(1)*X                                                3C
      IF(NL.LE.1)GOTO 30                                                   4C
      C1=1.0                                                               5C
      C2=X                                                                 6C
      DO 20 N=2,NL                                                         7C
      AL=N                                                                 8C
      C3=((2.)*AL-1.0)*X*C2-(AL-1.0)*C1)/AL                                9C
      LEGP=LEGP+(AL+0.5)*CON(N)*C3                                        10C
      C1=C2                                                               11C
   20 C2=C3                                                               12C
   30 RETURN                                                              13C
      END                                                                 14C
```

```
      SUBROUTINE RDAB                                                      1C
      DIMENSION ISATZ(400),XSATZ(400),BK(1)                                2C
      EQUIVALENCE (ISATZ(1),XSATZ(1))                                      3C
      COMMON/TAB/ENDTAB(25),KEDTAB(50),KEDR(177),NCNAME(20),NRIF3,NRIF4    4C
      DIMENSION KDR1(117),KDR2(60)                                         5C
      INTEGER ENDTAB                                                       6C
      EQUIVALENCE (KEDR(1),KDR1(1)),(KEDR(118),KDR2(1))                    7C
      COMMON/INOUT/NR1(83),NTYP,NR2(44),N                                  8C
      ENTRY RDA1(BK)                                                       9C
      NO=6                                                                10C
      NF=11                                                               11C
      L1=1                                                                12C
      N4=400                                                              13C
      READ(NF'L1)(ISATZ(I),I=1,N4)                                        14C
      RETURN                                                              15C
      ENTRY RDA4(MATKDK,MIS,NAME1,NAME2,LNT,IR)                           16C
      L4=1                                                                17C
      GOTO 20                                                             18C
      ENTRY RDA2(MATKDK,MIS,NAME1,NAME2,LNT)                              19C
      L4=0                                                                20C
   20 LNT=0                                                               21C
      IW=1                                                                22C
    3 IF(ISATZ(IW).NE.MATKDK)GOTO 1                                       23C
      IF(ISATZ(IW+1).NE.MIS)GOTO 1                                        24C
      IF(ISATZ(IW+2).NE.NAME1)GOTO 1                                      25C
      IF(ISATZ(IW+3).NE.NAME2)GOTO 1                                      26C
      GOTO 5                                                              27C
    1 IW=IW+6                                                             28C
      IF(IW-360)3,2,2                                                     29C
    5 IR=ISATZ(IW+4)                                                      30C
      N=ISATZ(IW+5)                                                       31C
      IF(L4.EQ.1)RETURN                                                   32C
   30 READ(NF'IR)(BK(I),I=1,N)                                            33C
      RETURN                                                              34C
    2 LNT=1                                                               35C
      RETURN                                                              36C
      ENTRY RDA3(MATKDK)                                                  37C
```

```
      NK=41                                                             380
      IK=1                                                              390
      NTYP=0                                                            400
   13 IF(ISATZ(IW).NE.MATKDK)GOTO 10                                    410
      IK=1                                                              420
      DO 11 ITK=1,NK                                                    430
      IF(ISATZ(IW+2).NE.KEDR(IK))GOTO 12                                440
      IF(ISATZ(IW+3).NE.KEDR(IK+1))GOTO 12                              450
      NTYP=NTYP+1                                                       460
      GOTO 10                                                           470
   12 IK=IK+3                                                           480
   11 CONTINUE                                                          490
   10 IW=IW+6                                                           500
      IF(IW-360)13,13,14                                                510
   14 RETURN                                                            520
      END                                                               530
```

```
      SUBROUTINE FILE5(MATP,IF5,BK,IBK,NBT,JNT,X,Y,B,TEMP)               10
      COMMON/INOUT/NO,NI,NT,MODE,NR1(78),MATKDK,NR2(42),NBX,N1X,N2X,N,    20
     *NRKX,NR3(10),EPS                                                   30
      COMMON/RECS/MAT,MF,MT,C1,C2,L1,L2,N1,N2                            40
      COMMON/TAB/ENDTAB(25),KEDTAB(50),KEDR(240),NCNAME(39),NRIF3,NRIF4   50
      EQUIVALENCE (KEDR(1),KDR1(1)),(KEDR(124),KDR2(1))                  60
      INTEGER ENDTAB                                                     70
      REAL*8 NAM8                                                        80
      EQUIVALENCE (NAUX(1),NAM8)                                         90
      DIMENSION NUMTYP(4),KEDTYP(4),ENERG(300),TEMP(1),IBK(1),BK(1),    100
     *NBT(1),JNT(1),X(1),NAUX(2),KDR1(123),KDR2(1),Y(1),B(1)            110
      DATA NUMTYP/16,17,18,91/                                          120
      DATA KEDTYP/61,67,70,55/                                          130
      NF=10                                                             140
C-----MAXIMUM NUMBER OF POINTS ALLOWED FOR THE PRIMARY ENERGY GRID      150
      NCMAX=300                                                         160
      WRITE(NO,1)                                                       170
    1 FORMAT('1',20X,'FILE 5 - ENERGY DISTRIBUTIONS OF SECONDARY NEUTRON 180
     *S'//21X,51('*')///)                                               190
      CALL PRIZEI                                                       200
      IPAGE=0                                                           210
      DO 2 NUM=1,4                                                      220
      CALL INITSC(NCMAX)                                                230
      NEN=0                                                             240
      REWIND NF                                                         250
      CALL SEARCH(NT,MODE,MATP,5,NUMTYP(NUM),LNT)                       260
      IF(LNT.EQ.0)GOTO 3                                                270
      WRITE(NO,4)NUMTYP(NUM)                                            280
    4 FORMAT('0ENERGY DISTRIBUTIONS FOR REACTION TYPE',I3,' NOT INCLUDED 290
     *IN THE ENDF/B FILE')                                              300
      BACKSPACE NT                                                      310
      GOTO 2                                                            320
    3 NK=N1                                                             330
      CALL WREC(1,NF,2)                                                 340
C=====READ PROBABILITY AND TEMPERATURE ARRAYS FROM ENDF/B FILE          350
C-----GENERATE AN ENERGY GRID DESCRIBING EVERY DISTRIBUTION WITH A PRECI 360
C-----STORE DATA ON FILE NF IN ENDF/B FORMAT                            370
      DO 5 NKN=1,NK                                                     380
      DO 8 NKN2=1,2                                                     390
      CALL RREC(3,NT,MODE,-1.0,NBT,JNT,X,Y)                             400
      CALL WREC(3,NF,2)                                                 410
      IF(NKN2.EQ.2)GOTO 6                                               420
      LF=L2                                                             430
      IF(LF.EQ.3.OR.LF.EQ.7.OR.LF.EQ.9.OR.LF.EQ.10)GOTO 6               440
      WRITE(NO,7)LF,NUMTYP(NUM)                                         450
    7 FORMAT('0LF EQUALS',I4,'. THIS CASE IS NOT FORSEEN IN THE CODE.'/ 460
     *1X,'REACTION TYPE',I4,' NOT TRANSLATED')                          470
      GOTO 2                                                            480
C-----LF=3,7,9,10                                                       490
    6 CALL CHGINT(EPS,N1,N2,NBT,JNT,X,Y,N1X,N2X,&30)                    500
            IF(N2.GT.NCMAX) CALL ERROR(751)                             510
      CALL GENER(N2,X,TEMP,ENERG,NO,NCU)                                520
      IF(NCU.GT.NCMAX)CALL ERROR(751)                                   530
      IF(LF.EQ.3)GOTO 5                                                 540
      IF(LF.EQ.10)GOTO 11                                               550
    8 CONTINUE                                                          560
      GOTO 5                                                            570
   11 CALL RREC(2,NT,MODE,-1.0,B)                                       580
      CALL WREC(2,NF,2)                                                 590
    5 CONTINUE                                                          600
      ENDFILE NF                                                        610
      REWIND NF                                                         620
C=====REREAD ENDF DATA FROM FILE NF AND CONSTRUCT ARRAY BK (DATA IN      630
C-----KEDAK FORMAT)                                                     640
      IF(IPAGE.EQ.1)WRITE(NO,31)                                        650
   31 FORMAT('1')                                                       660
      K2=KEDTYP(NUM)                                                    670
      NAUX(1)=KDR2(K2)                                                  680
      NAUX(2)=KDR2(K2+1)                                                690
      CALL A8FORM(NO,NAM8)                                              700
      IPAGE=1                                                           710
      WRITE(NO,2000)(ENERG(I),I=1,NCU)                                  720
 2000 FORMAT(//'0ENERGY GRID GENERATED'/(1P8E15.5))                      730
C-----LENGTH OF TYPE TABLE: NLT                                         740
      NLT=4*NCU                                                         750
C-----INDEX FOR WRITING IN RELOCATABLE PART: ILT                        760
C-----INDEX FOR WRITING DATA: IA                                        770
      ILT=1                                                             780
      CALL RREC(1,NF,2,-1.0)                                            790
      NK=NI                                                             800
      NKM4=4*NK                                                         810
C-----LOOP ON SPECTRUM TYPES                                            820
      DO 12 NKN=1,NK                                                    830
      CALL RREC(3,NF,2,-1.0,NBT,JNT,X,Y)                                840
      LF=L2                                                             850
      IA=NLT+1+4*(NKN-1)                                                860
C-----LOOP ON INCIDENT ENERGIES                                         870
      DO 14 I=1,NCU                                                     880
      IF((IA+3).GT.NBKX) CALL ERROR(752)                                890
      IF(NKN.GT.1) GOTO 36                                              900
```

```
      BK(ILT)=ENERG(I)                                                  910
      IBK(ILT+1)=NK                                                     920
      IBK(ILT+2)=(IA-1)/880+1                                           930
      IBK(ILT+3)=MOD(IA-1,880)+1                                        940
      ILT=ILT+4                                                         950
C=====SET KEDAK DATA TYPE NUMBER OF THE DISTRIBUTION                    960
C-----LF=3                                                              970
   36 IF(LF.NE.3)GOTO 32                                                980
      BK(IA)=4                                                          990
      GOTO 35                                                          1000
C-----LF=7                                                             1010
   32 IF(LF.NE.7)GOTO 33                                               1020
      BK(IA)=2                                                         1030
      GOTO 35                                                          1040
C-----LF=9                                                             1050
   33 IF(LF.NE.9)GOTO 34                                               1060
      BK(IA)=1                                                         1070
      GOTO 35                                                          1080
C-----LF=10                                                            1090
   34 BK(IA)=3                                                         1100
C-----INTERPOLATION IN THE PROBABILITY ARRAY                           1110
   35 CALL ITAB1(ENERG(I),BK(IA+1))                                    1120
      IF(LF.NE.3)GOTO 15                                               1130
      BK(IA+2)=C2                                                      1140
      BK(IA+3)=0.0                                                     1150
      GOTO 14                                                          1160
   15 IF(LF.NE.7.AND.LF.NE.9)GOTO 14                                   1170
C-----LF=7 OR 9: U                                                     1180
      BK(IA+3)=C1                                                      1190
   14 IA=IA+NKM4                                                       1200
      IF(LF.EQ.3)GOTO 12                                               1210
      IF(LF.EQ.10)GOTO 16                                              1220
C-----READ THE TEMPERATURE ARRAY                                       1230
      CALL RREC(3,NF,2,-1.0,NBT,JNT,X,Y)                               1240
      GOTO 17                                                          1250
C-----READ THE WATT FISSION SPECTRUM PARAMETERS                        1260
   16 CALL RREC(2,NF,2,-1.0,B)                                         1270
C-----LOOP ON INCIDENT ENERGIES                                        1280
   17 IA=IA-4*NCU*NK                                                   1290
      DO 18 I=1,NCU                                                    1300
      IF(LF.EQ.10)GOTO 19                                              1310
C-----LF=7 OR 9: INTERPOLATE TEMPERATURE                               1320
      CALL ITAB1(ENERG(I),BK(IA+2))                                    1330
      GOTO 18                                                          1340
C-----LF=10                                                            1350
C     DIE BEIDEN FOLGENDEN STATEMENTS SIND VERMUTLICH  F A L S C H .   1360
   19 BK(IA+2)=B(1)                                                    1370
      BK(IA+3)=B(2)                                                    1380
   18 IA=IA+NKM4                                                       1390
   12 CONTINUE                                                         1400
      NCNAME(KEDTYP(NUM)/3+1)=NCU                                      1410
      WRITE(NO,20)KDR2(K2),KDR2(K2+1)                                  1420
   20 FORMAT('0',2A4,' REACTION TYPE AND PARAMETER TABLE')             1430
      IA=IA-1-4*(NK-1)                                                 1440
      ILT=ILT-1                                                        1450
      DO 40 I=1,ILT,4                                                  1460
      IP3=I+3                                                          1470
      WRITE(NO,21)(BK(II),II=I,IP3)                                    1480
   21 FORMAT(1X,1PE15.5,3I8)                                           1490
      IPA=IBK(I+3)                                                     1500
      DO 41 J=1,NK                                                     1510
      IPE=IPA+3                                                        1520
      WRITE(NO,22)(BK(II),II=IPA,IPE)                                  1530
   22 FORMAT(10X,1P4E15.5)                                             1540
   41 IPA=IPE+1                                                        1550
   40 CONTINUE                                                         1560
      N=IA                                                             1570
      CALL WDA2(MATKDK,1,KDR2(K2),KDR2(K2+1))                          1580
      NRIF4=NRIF4+1                                                    1590
    2 CONTINUE                                                         1600
      RETURN                                                           1610
   30 CALL ERROR (750)                                                 1620
      STOP                                                             1630
      END                                                              1640
```

```
      SUBROUTINE CHGINT(EPS,N1,N2,NBT,JNT,X,Y,N1X,N2X,*)                10
C     SUBROUTINE CHGINT  J.DANIELS NOV. 1970                            20
C                        MODIFIED MAY 1972                               30
      DIMENSION NBT(1),JNT(1),X(1),Y(1)                                 40
      COMMON/RF/XR(300),YR(300)                                         50
      DO 1  I=1,N1                                                      60
      IF(I-1)95,96,95                                                   70
   96 J=1                                                               80
      GO TO 97                                                          90
   95 J=NBT(I-1)                                                       100
   97 LLL=NBT(I)-1                                                     110
      JNTX=JNT(I)                                                      120
      IF(JNTX.LE.0.OR.JNTX.GT.5)CALL ERROR(621)                        130
      IF(JNTX.EQ.1)GO TO 3                                             140
      IF(JNTX.EQ.2) GOTO 1                                             150
   28 N=2                                                              160
      X1=X(J)                                                          170
      Y1=Y(J)                                                          180
      X2=X(J+1)                                                        190
      Y2=Y(J+1)                                                        200
      IF(ABS(X1-X2)/X1.LE.1.0E-06)GO TO 170                            210
      CALL DANI(X1,Y1,X2,Y2,JNTX,N,EPS)                                220
      IF(N.EQ.2) GOTO 170                                              230
      KKK=J+N-1                                                        240
      N2OLD=N2                                                         250
      N2=N2+N-2                                                        260
      IF(N2.GT.N2X)RETURN 1                                            270
   25 DO 7 KK=KKK,N2                                                   280
      IX1=N2-KK+KKK                                                    290
      IX2=N2OLD-KK+KKK                                                 300
      X(IX1)=X(IX2)                                                    310
    7 Y(IX1)=Y(IX2)                                                    320
```

```
      DO 8 K=1,N                                                        330
      X(J+K-1)=XR(K)                                                    340
      Y(J+K-1)=YR(K)                                                    350
    8 CONTINUE                                                          360
      DO 6 K=I,N1                                                       370
    6 NBT(K)=NBT(K)+N-2                                                 380
  170 J=J+N-1                                                           390
      LLL=LLL+N-2                                                       400
      IF(J.LE.LLL)GO TO 28                                              410
    3 JNT(I)=2                                                          420
    1 CONTINUE                                                          430
C                                                                       440
C=====ELIMINATE DISCONTINUITIES.                                        450
C                                                                       460
      L=N2-1                                                            470
      I=1                                                               480
  303 K=1                                                               490
      IF(X(I).NE.X(I+1))GO TO 302                                       500
      X(I)=X(I)*0.99999                                                 510
      IF(I.GT.(L-1)) GOTO 304                                           520
      K=2                                                               530
      IF(X(I+1).NE.X(I+2))GO TO 302                                     540
      K=3                                                               550
      X(I+2)=X(I+2)+1.0E-5*X(I+2)                                       560
      Y(I+1)=Y(I)+(X(I+1)-X(I))*(Y(I+2)-Y(I))/(X(I+2)-X(I))             570
  302 I=I+K                                                             580
      IF(I.GT.L)GOTO 304                                                590
      GO TO 303                                                         600
  304 CONTINUE                                                          610
C     INTERPOLATIONSTABELLE VEREINFACHEN.                               620
      DO 2 M=1,N1                                                       630
      IF(JNT(M).NE.2) CALL ERROR(777)                                   640
    2 CONTINUE                                                          650
      NBT(1)=NBT(N1)                                                    660
      N1=1                                                              670
C     FALLS Y(1)<=1.0E-20 IST, SO WIRD Y(1)=0.0 GESETZT.                680
      IF(Y(1).LE.1.0E-20) Y(1)=0.0                                      690
      RETURN                                                            700
      END                                                               710
```

```
      SUBROUTINE DANI(X1,Y1,X2,Y2,JNTX,N,EPS)                           10
C     SUBROUTINE DANI    J.DANIELS    NOVEMBER 1970                     20
C     CHANGED BY E. STEIN IN SEPTEMBRE 1973.                            30
C     INTERPOLATION OF VALUES FROM ENDF  LIBRARY                        40
      COMMON/RE/XR(300),YR(300)                                         50
      XR(1)=X1                                                          60
      YR(1)=Y1                                                          70
      XR(2)=X2                                                          80
      YR(2)=Y2                                                          90
      N=2                                                              100
      J=1                                                              110
      IF(X1.EQ.X2)GOTO 1010                                            120
      IF(Y1.EQ.Y2)GOTO 1010                                            130
      DO 100 M=1,300                                                   140
  600 XP=(XR(J)+XR(J+1))*0.5                                           150
      GOTO(1010,1010,501,502,503),JNTX                                 160
  500 GOTO 1010                                                        170
  501 IF(XR(J))50,50,70                                                180
   70 IF(XR(J+1))50,50,80                                              190
   80 IF(XP)90,90,101                                                  200
   90 CALL ERROR(134)                                                  210
  101 XVER=XR(J+1)/XR(J)                                               220
      XVERM1=XVER-1.0                                                  230
      IF(XVERM1.GT.1.0E-3.OR.XVERM1.LT.0) GOTO 2101                    240
      AKOR=0.125*XVERM1*XVER/(1.0-XVERM1*(0.5-0.33333333*XVERM1))      250
      YP=0.5*(YR(J)+YR(J+1))-AKOR*(YR(J+1)-YR(J))                      260
      GOTO 700                                                         270
 2101 YP=YR(J)+ALOG(XP/XR(J))*(YR(J+1)-YR(J))/ALOG(XVER)               280
      GO TO 700                                                        290
  502 IF(YR(J))50,50,120                                               300
  120 IF(YR(J+1))50,50,130                                             310
  130 YP=SQRT(YR(J+1)*YR(J))                                           320
      GO TO 700                                                        330
  503 IF(YR(J))501,501,150                                             340
  150 IF(YR(J+1))501,501,160                                           350
  160 IF(XR(J))130,130,170                                             360
  170 IF(XR(J+1))130,130,180                                           370
  180 IF(XP)90,90,190                                                  380
  190 XVER=XR(J+1)/XR(J)                                               390
      XVERM1=XVER-1.0                                                  400
      IF(XVERM1.GT.1.0E-3.OR.XVERM1.LT.0) GOTO 2190                    410
      AKOR=0.125*XVERM1*XVER/(1.0-XVERM1*(0.5-0.33333333*XVERM1))      420
      YP=SQRT(YR(J+1)*YR(J))*EXP(-AKOR*ALOG(YR(J+1)/YR(J)))            430
      GOTO 700                                                         440
 2190 YP=YR(J)*EXP(ALOG(XP/XR(J))*ALOG(YR(J+1)/YR(J))/ALOG(XVER))      450
  700 YLIN=0.5*(YR(J)+YR(J+1))                                         460
      IF(YP.NE.0.)GOTO 800                                             470
      TEST=0.0                                                         480
      GOTO 801                                                         490
  800 TEST=ABS((YLIN-YP)/YP)                                           500
  801 IF (TEST-EPS)1001,1001,1002                                      510
 1002 N=N+1                                                            520
      KKK=N                                                            530
      L=J+2                                                            540
   45 XR(KKK)=XR(KKK-1)                                                550
      YR(KKK)=YR(KKK-1)                                                560
      KKK=KKK-1                                                        570
      IF(KKK-L)66,65,65                                                580
   66 XR(KKK)=XP                                                       590
      YR(KKK)=YP                                                       600
      GO TO 600                                                        610
 1001 IF(XR(M+1)-X2)1009,1010,1010                                     620
 1009 J=J+1                                                            630
  100 CONTINUE                                                         640
      CALL ERROR(620)                                                  650
   50 YP=0.5*(YR(J)+YR(J+1))                                           660
      GO TO 700                                                        670
```

```
 1010 RETURN                                                            680
      END                                                               690
```

```
      SUBROUTINE COMB(OPER,CON,MA,MB,XL,XH,EPS,LOF,LNT)                  10
C=====COMBINE TWO TAB1 FUNCTIONS=====================================   20
C     THE SUBROUTINE WILL PRODUCE A TAB1 FUNCTION IN THE COMMON         30
C     RECS AND IN THE ARRAYS NBT,JNT,X, AND Y.                          40
C     ACCURATE TO EPS FOR X IN THE RANGE (XL,XH). IF FA(X) IS THE       50
C     TAB1 RECORD IDENTIFIED BY MA, AND FB(X) IS THE TAB1 RECORD        60
C     IDENTIFIED BY MB, THE ANSWER FC(X)=FA(X) * FB(X), WHERE THE       70
C     OPERATION * IS DEFINED IN THE FUNCTION S.R. OPER WHICH MAY        80
C     ALSO USE CONSTANTS CON(N).                                        90
C     MA = IDENT OF TAB1 RECORD IN DENSE STORAGE                       100
C     MB = IDENT OF TAB1 RECORD IN DENSE STORAGE                       110
C     XL = LOWER LIMIT OF X REQUIRED                                   120
C     XH = UPPER LIMIT OF X REQUIRED                                   130
C     LOF= OVERFLOW INDICATOR NORMALLY 0. IF LOF=1, ANSWER WILL        140
C          NOT FIT IN EXPANDED STORAGE                                 150
C     EPS= RELATIVE ERROR CRITERION                                    160
C     LNT= 0, NORMAL EXIT.                                             170
C        = 1, RECORD MA OR MB NOT IN STORAGE                           180
C-----ERROR STOP 310, XL.GE.XH                                         190
C     ERROR STOP 311, MA OR MB IS ZERO                                 200
      DIMENSION CON(20),NBT(1),JNT(1),X(1),Y(1)                        210
      LOGICAL*4 LODZMS                                                 220
      COMMON/RECS/NR1(7),N1,N2                                         230
      COMMON/INOUT/ NR2(141),SCREPS                                    240
      EXTERNAL OPER                                                    250
C=====INITIALIZE                                                       260
  100 IF(XL.GE.XH)CALL ERROR(310)                                      270
      IF((MA.EQ.0).OR.(MB.EQ.0))CALL ERROR(311)                        280
      N1=0                                                             290
      N2=0                                                             300
      CALL LRIDS(MA,JA,LNT)                                            310
      IF(LNT.GT.0)GOTO400                                              320
      CALL LRIDS(MB,JB,LNT)                                            330
      IF(LNT.GT.0)GOTO400                                              340
C-----GET STARTING POINTS                                              350
      NA=0                                                             360
      CALL IPDS(JA,NA,XL,YA1,IA1)                                      370
      NA=NA+1                                                          380
      CALL FPDS(JA,NA,XA,YA2,IA2)                                      390
      NB=0                                                             400
      CALL IPDS(JB,NB,XL,YB1,IB1)                                      410
      NB=NB+1                                                          420
      CALL FPDS(JB,NB,XB,YB2,IB2)                                      430
      X1=XL                                                            440
C=====TEST FOR SMALLEST OF XA, XB, XH                                  450
  200 IF(XA.GT.XB)GOTO220                                              460
      IF(XA.EQ.XB)GOTO230                                              470
C-----XA.LT.XB                                                         480
  210 IF(XA.GE.XH)GOTO300                                              490
      X2=XA                                                            500
      CALL IPDS(JB,NB-1,X2,YB3,IB3)                                    510
      CALL COMBP(X1,YA1,YB1,X2,YA2,YB3,IA1,IB3,OPER,CON,EPS,LOF,       520
     *NBT,JNT,X,Y)                                                     530
      IF(LOF.GT.0)GOTO400                                              540
      X1=X2                                                            550
      YA1=YA2                                                          560
      IA1=IA2                                                          570
      NA=NA+1                                                          580
      YB1=YB3                                                          590
      IB1=IB3                                                          600
      CALL FPDS(JA,NA,XA,YA2,IA2)                                      610
      GOTO200                                                          620
C-----XA.GT.XB                                                         630
  220 IF(XB.GE.XH)GOTO300                                              640
      X2=XB                                                            650
      CALL IPDS(JA,NA-1,X2,YA3,IA3)                                    660
      CALL COMBP(X1,YA1,YB1,X2,YA3,YB2,IA3,IB1,OPER,CON,EPS,LOF,       670
     *NBT,JNT,X,Y)                                                     680
      IF(LOF.GT.0)GOTO400                                              690
      X1=X2                                                            700
      YB1=YB2                                                          710
      IB1=IB2                                                          720
      NB=NB+1                                                          730
      YA1=YA3                                                          740
      IA1=IA3                                                          750
      CALL FPDS(JB,NB,XB,YB2,IB2)                                      760
      GOTO200                                                          770
C-----XA.EQ.XB                                                         780
  230 IF(XA.GE.XH)GOTO300                                              790
      X2=XA                                                            800
      CALL COMBP(X1,YA1,YB1,X2,YA2,YB2,IA1,IB1,OPER,CON,EPS,LOF,       810
     *NBT,JNT,X,Y)                                                     820
      IF(LOF.GT.0)GOTO400                                              830
      X1=X2                                                            840
      YA1=YA2                                                          850
      IA1=IA2                                                          860
      NA=NA+1                                                          870
      CALL FPDS(JA,NA,XA,YA2,IA2)                                      880
      YB1=YB2                                                          890
      IB1=IB2                                                          900
      NB=NB+1                                                          910
      CALL FPDS(JB,NB,XB,YB2,IB2)                                      920
      GOTO200                                                          930
C=====X2.GT.XH                                                         940
  300 CALL IPDS(JA,NA-1,XH,YA2,IA2)                                    950
      CALL IPDS(JB,NB-1,XH,YB2,IB2)                                    960
      CALL COMBP(X1,YA1,YB1,XH,YA2,YB2,IA1,IB1,OPER,CON,EPS,LOF,       970
     *NBT,JNT,X,Y)                                                     980
      XLC=X(1)                                                         990
      XHC=X(N2)                                                       1000
      CALL CROP(XLC,XHC,SCREPS,.FALSE.,SCREPS,DUMMY,DUMMY,LOF,NBT,JNT,X, 1010
     *Y)                                                              1020
C=====FINISHED                                                        1030
  400 RETURN                                                          1040
```

```
      ENTRY IICOMB(NBT,JNT,X,Y)                                          1050
      GOTO 400                                                           1060
      END                                                                1070
```

```
      SUBROUTINE ERROR(N)                                                  10
C=====PRINT ERROR MESSAGE=============================================     20
      COMMON/RECS/MAT,MF,MT                                                30
      COMMON/INOUT/NOUT                                                    40
   10 WRITE(NOUT,20)N                                                      50
   20 FORMAT(11HOERROR STOP,I6)                                            60
      WRITE(NOUT,30) MAT,MF,MT                                             70
   30 FORMAT(5HOMAT=,I5,4H MF=,I3,4H MT=,I4)                               80
      WRITE(NOUT,50)                                                       90
   50 FORMAT(///'          ***DUMP OF THE ARRAYS NBT,JNT,X, AND Y.***'//)  100
      CALL WREC(3,NOUT,4)                                                 110
      STOP 1                                                              120
      END                                                                 130
```

```
      SUBROUTINE MAGGY(XD,YD,N,X,Y)                                        10
      DIMENSION XD(1),YD(1),X(1),Y(1)                                      20
      DO 1 I=1,N                                                           30
      X(I)=XD(I)                                                           40
      Y(I)=YD(I)                                                           50
    1 CONTINUE                                                             60
      RETURN                                                               70
      END                                                                  80
```

```
      SUBROUTINE SEARCH(NT,MODE,MATP,MFP,MTP,LNT)                          10
C=====SEARCH TAPE NT FOR A PARTICULAR SECTION=========================     20
C     NT   - LOGICAL TAPE NUMBER                                           30
C     MODE - TAPE MODE(1,2,3)                                              40
C     MATP - DESIRED MATERIAL NUMBER(SEE NOTE 2)                           50
C     MFP  - DESIRED FILE NUMBER(SEE NOTE 2)                               60
C     MTP  - DESIRED SECTION NUMBER(SEE NOTE 2)                            70
C     LNT  = 0, SECTION LOCATED, HEAD RECORD READ AND IN /RECS/            80
C          = 1, SECTION NOT LOCATED BECAUSE EOT ENCOUNTERED, OR            90
C               BECAUSE DESIRED RECORD WOULD OCCUR PRIOR TO               100
C               CURRENT POSITION OF TAPE(SEE NOTE 1).                     110
C     NOTE 1 - SEARCH IS IN FORWARD DIRECTION ONLY                        120
C     NOTE 2 - ZERO VALUES OF MATP, MFP, MTP ARE READ AS (ANY).           130
C              THUS, MATP,MFP,0 MEANS ANY SECTION IN MATERIAL MATP,       140
C              FILE MFP.  0,MFP,0 MEANS ANY SECTION, FILE MFP             150
C              FOR ANY MATERIAL.                                          160
C-----ERROR STOP 317, MODE NOT IN RANGE 1-3                              170
      COMMON/RECS/MAT,MF,MT                                              180
      LNT=0                                                              190
C-----TEST MODE                                                          200
      IF((MODE.LT.1).OR.(MODE.GT.3))CALL ERROR(317)                      210
      GOTO(100,200,100),MODE                                             220
C-----MODES 1 AND 3                                                      230
  100 CALL RREC(1,NT,MODE,-1.0)                                          240
      IF(MAT.LT.0)GOTO300                                                250
      IF(MAT.EQ.0)GOTO100                                                260
      IF(MF.EQ.0)GOTO100                                                 270
      IF(MT.EQ.0)GOTO100                                                 280
      IF(MATP.EQ.0)GOTO110                                               290
      IF(MATP-MAT)300,110,130                                            300
  110 IF(MFP.EQ.0)GOTO120                                                310
      IF(MFP-MF) 300,120,130                                             320
  120 IF(MTP.EQ.0)GOTO400                                                330
      IF(MTP-MT) 300,400,130                                             340
  130 GOTO(140,140,150),MODE                                             350
  140 READ(NT)MAT,MF,MT                                                  360
      GOTO160                                                            370
  150 READ(NT,155)MAT,MF,MT                                              380
  155 FORMAT(66X,I4,I2,I3)                                               390
  160 IF(MT.NE.0)GOTO130                                                 400
      GOTO100                                                            410
C-----MODE 2                                                             420
  200 CALL RREC(1,NT,2,-1.0)                                             430
      IF(MF.LT.0)GOTO300                                                 440
      IF(MF.EQ.0)GOTO200                                                 450
      IF(MAT.EQ.0)GOTO200                                                460
      IF(MT.EQ.0)GOTO200                                                 470
      IF(MFP.EQ.0)GOTO210                                                480
      IF(MFP-MF)300,210,230                                              490
  210 IF(MATP.EQ.0)GOTO220                                               500
      IF(MATP-MAT)300,220,230                                            510
  220 IF(MTP.EQ.0)GOTO400                                                520
      IF(MTP-MT)300,400,230                                              530
  230 READ(NT)MF,MAT,MT                                                  540
      IF(MT.NE.0)GOTO230                                                 550
      GOTO200                                                            560
C-----SECTION NOT THERE                                                  570
  300 LNT=1                                                              580
C-----FINISHED                                                           590
  400 RETURN                                                             600
      END                                                                610
```

```
      SUBROUTINE TERP1(X1,Y1,X2,Y2,X,Y,I)                                  10
C----------CHANGED OCTOBRE 1973 BY E.STEIN.                               20
C=====INTERPOLATE ONE POINT===========================================     30
C     (X1,Y1) AND (X2,Y2) ARE THE END POINTS OF THE LINE                   40
C     (X,Y) IS THE INTERPOLATED POINT                                      50
C     I IS THE INTERPOLATION CODE                                          60
C           IF XA.LE.0 OR XB.LE.0 OR XP/XA.EQ.0 AND XB/XA.EQ.0             70
C           DETECTED, THE INTERPOLATION CODE IS AUTOMATICALLY              80
C           CHANGED FROM LOG TO LINEAR                                     90
```

```
C-----ERROR STOP 133, INTERPOLATION CODE OUT OF RANGE
C     ERROR STOP 134, ZERO OR NEG VALUE CANNOT BE
C                     INTERPOLATED BY LOGS
C-----ERROR STOP 135, X1=X2 (DISCONTINUITY)
   10 XA=X1
      YA=Y1
      XB=X2
      YB=Y2
      XP=X
      II=I
      IF(XA.EQ.XB) CALL ERROR(135)
      IF(II.LE.0.OR.II.GT.5) CALL ERROR(133)
      IF(YA.EQ.YB.OR.XP.EQ.XA)GOTO 40
      IF(XP.EQ.XB) GOTO 45
      GO TO (40,50,60,110,140),II
   40 YP=YA
      GO TO 200
   45 YP=YB
      GOTO 200
   50 YP=YA+(XP-XA)*(YB-YA)/(XB-XA)
      GO TO 200
   60 IF(XA) 50,50,70
   70 IF(XB) 50,50,80
   80 IF(XP) 90,90,100
   90 CALL ERROR(134)
  100 ASSIGN 104 TO LAB01
  105 Q1=XP/XA
      Q2=XB/XA
      Q11=Q1-1.0
      Q21=Q2-1.0
      IF(Q21.EQ.0.AND.Q11.EQ.0) GOTO 50
      IF(ABS(Q11).GT.1.0E-4) GOTO 101
      T1=Q11*(1.0-0.5*Q11)
      IF(ABS(Q21).GT.1.0E-4) GOTO 102
      T2=Q21*(1.0-0.5*Q21)
      GOTO 103
  101 T1=ALOG(Q1)
  102 T2=ALOG(Q2)
  103 GOTO LAB01,(104,194)
  104 YP=YA+(YB-YA)*T1/T2
      GO TO 200
  110 IF(YA) 50,50,120
  120 IF(YB) 50,50,130
  130 YP=YA*EXP((XP-XA)*ALOG(YB/YA)/(XB-XA))
      GO TO 200
  140 IF(YA) 60,60,150
  150 IF(YB) 60,60,160
  160 IF(XA) 130,130,170
  170 IF(XB) 130,130,180
  180 IF(XP) 90,90,190
  190 ASSIGN 194 TO LAB01
      GOTO 105
  194 YP=YA*EXP(ALOG(YB/YA)*T1/T2)
  200 Y=YP
      RETURN
      END
```

```
      SUBROUTINE WDA
      COMMON/INOUT/NO,NR1(127),N,NBKX
      DIMENSION ISATZ(400),XSATZ(400),BK(1)
      EQUIVALENCE (ISATZ(1),XSATZ(1)),(ISATZ(399),IR),(ISATZ(400),IW)
C
C
C
      ENTRY WDA1(BK)
      NE=11
      DO 1 I=1,400
      ISATZ(I)=0
    1 CONTINUE
      IR=2
      IW=1
      WRITE(NE'1)(ISATZ(I),I=1,400)
      RETURN
C
C
C
      ENTRY WDA2(MATKDK,MIS,NAME1,NAME2)
      IF(N.GT.NBKX )CALL ERROR(516)
      READ(NE'1)(ISATZ(I),I=1,400)
      ISATZ(IW)=MATKDK
      ISATZ(IW+1)=MIS
      ISATZ(IW+2)=NAME1
      ISATZ(IW+3)=NAME2
      ISATZ(IW+4)=IR
      ISATZ(IW+5)=N
      IW=IW+6
      IF(IW.GT.390)CALL ERROR(550)
      WRITE(NE'IR)(BK(I),I=1,N)
      IR=IR+N/400+1
      WRITE(NE'1)(ISATZ(I),I=1,400)
      RETURN
C
C
C
      ENTRY WDA3(IT)
      READ(NE'1)(ISATZ(I),I=1,400)
      WRITE(NO,4)
    4 FORMAT(1H1,'CONTENTS OF FILE 11'///)
      IW1=IW-1
      DO 2 I=1,IW1,6
      WRITE(NO,3)(ISATZ(I),ISATZ(I+1),ISATZ(I+2),ISATZ(I+3),ISATZ(I+4),
     *   ISATZ(I+5))
    3 FORMAT(1X,216,4X,2A4,2I6)
    2 CONTINUE
      WRITE(NE'1)(ISATZ(I),I=1,400)
      IT=IW
      RETURN
      END
```

```
      SUBROUTINE WREC(JT,NT,MODE)                                       10
C=====WRITE ONE RECORD ON ENDF/B TAPE=================================  20
C     JT  = RECORD TYPE, 1-CONT, 2-LIST, 3-TAB1, 4-TAB2, 5-HOL LIST     30
C                        6-TPID                                         40
C     NT  = OUTPUT TAPE NUMBER. IF NT.LE.0, WRITE IS IGNORED.           50
C     MODE=1 BINARY TAPE, STANDARD ARRANGEMENT.                         60
C         =2 BINARY TAPE, ALTERNATE ARRANGEMENT.                        70
C         =3 BCD CARD IMAGE TAPE FOR PUNCHING.                          80
C         =4 EXPANDED AND INTERPRETED PRINT TAPE.                       90
C-----ERROR STOP 108, JT OUT OF RANGE 1-6                              100
C     ERROR STOP 109, MODE OUT OF RANGE 1-4                            110
      DIMENSION NBT(1),JNT(1),X(1),Y(1),B(1)                           120
      COMMON/RECS/MAT,MF,MT,C1,C2,L1,L2,N1,N2                          130
C-----PRELIMINARY TESTS                                                140
  100 IF(NT.LE.0)GOTO900                                               150
      IF(JT.LT.1.OR.JT.GT.6) CALL ERROR(108)                           160
      IF(MODE.LT.1.OR.MODE.GT.4) CALL ERROR(109)                       170
      GOTO(200,300,400,500),MODE                                       180
C-----BINARY TAPE, STANDARD ARRANGEMENT                                190
  200 GOTO(210,220,230,240,220,245), JT                                200
  210 WRITE(NT)MAT,MF,MT,C1,C2,L1,L2,N1,N2                             210
      GOTO250                                                          220
  220 WRITE(NT)MAT,MF,MT,C1,C2,L1,L2,N1,N2,(B(N),N=1,N1)               230
      GOTO250                                                          240
  230 WRITE(NT)MAT,MF,MT,C1,C2,L1,L2,N1,N2,(NBT(N),JNT(N),N=1,N1),     250
     1(X(N),Y(N),N=1,N2)                                               260
      GOTO250                                                          270
  240 WRITE(NT)MAT,MF,MT,C1,C2,L1,L2,N1,N2,(NBT(N),JNT(N),N=1,N1)      280
      GO TO 250                                                        290
  245 WRITE (NT) MAT,MF,MT,(B(N),N=1,17)                               300
  250 GOTO900                                                          310
C-----BINARY TAPE, ALTERNATE ARRANGEMENT                               320
  300 GOTO(310,320,330,340,320,345), JT                                330
  310 WRITE(NT)MF,MAT,MT,C1,C2,L1,L2,N1,N2                             340
      GOTO350                                                          350
  320 WRITE(NT)MF,MAT,MT,C1,C2,L1,L2,N1,N2,(B(N),N=1,N1)               360
      GOTO350                                                          370
  330 WRITE(NT)MF,MAT,MT,C1,C2,L1,L2,N1,N2,(NBT(N),JNT(N),N=1,N1),     380
     1(X(N),Y(N),N=1,N2)                                               390
      GOTO350                                                          400
  340 WRITE(NT)MF,MAT,MT,C1,C2,L1,L2,N1,N2,(NBT(N),JNT(N),N=1,N1)      410
      GO TO 350                                                        420
  345 WRITE (NT) MF,MAT,MT,(B(N),N=1,17)                               430
  350 GOTO900                                                          440
C-----BCD CARD IMAGE TAPE                                              450
  400 GOTO(420,430,440,450,460,470), JT                                460
  420 CALL PUCONT(NT)                                                  470
      GOTO900                                                          480
  430 CALL PULIST(NT,B)                                                490
      GOTO900                                                          500
  440 CALL PUTAB1(NT,NBT,JNT,X,Y)                                      510
      GOTO900                                                          520
  450 CALL PUTAB2(NT,NBT,JNT)                                          530
      GOTO900                                                          540
  460 CALL PUHOL(NT,B)                                                 550
      GOTO900                                                          560
  470 CALL PUTPID(NT,B)                                                570
      GO TO 900                                                        580
C-----EXPANDED AND INTERPRETED PRINT                                   590
  500 GOTO(520,530,540,550,560,570), JT                                600
  520 CALL PRCONT(NT)                                                  610
      GOTO900                                                          620
  530 CALL PRLIST(NT,B)                                                630
      GOTO900                                                          640
  540 CALL PRTAB1(NT,NBT,JNT,X,Y)                                      650
      GOTO900                                                          660
  550 CALL PRTAB2(NT,NBT,JNT)                                          670
      GOTO900                                                          680
  560 CALL PRHOL(NT,B)                                                 690
      GO TO 900                                                        700
  570 CALL PRTPID(NT,B)                                                710
C-----FINISHED                                                         720
  900 RETURN                                                           730
      ENTRY IWREC(NBT,JNT,X,Y,B)                                       740
      GOTO 900                                                         750
      END                                                              760
```

```
      SUBROUTINE WRECS                                                  10
      INTEGER BIB12                                                     20
      COMMON/INOUT/NO,NR1(77),BIB12                                     30
      COMMON/RECS/MAT,MF,MT,C1,C2,L1,L2,N1,N2                           40
      DIMENSION ISATZ(400),XSATZ(400),NBT(1),JNT(1),X(1),Y(1),B(1)      50
      EQUIVALENCE (ISATZ(1),XSATZ(1))                                   60
      ENTRY WRECS1(NBT,JNT,X,Y,B)                                       70
      K12=1                                                             80
      NF=12                                                             90
      IBIB12=1                                                         100
      IR=BIB12+1                                                       110
      IW=1                                                             120
      DO 1 I=1,400                                                     130
      ISATZ(I)=0                                                       140
    1 CONTINUE                                                         150
      DO 4 IB=1,BIB12                                                  160
    4 WRITE(NF'IB)(ISATZ(I),I=1,400)                                   170
      GOTO 2000                                                        180
      ENTRY WRECS2(MIS,NAME1,NAME2,EL,EH)                              190
      ASSIGN 1000 TO LABEN                                             200
      IF(IW.GE.400) GOTO 3                                             210
 1000 READ(NF'IBIB12)(ISATZ(I),I=1,400)                                220
      ISATZ(IW)=MIS                                                    230
      ISATZ(IW+1)=NAME1                                                240
      ISATZ(IW+2)=NAME2                                                250
      XSATZ(IW+3)=EL                                                   260
```

```
      XSATZ(IW+4)=EH                                                    270
      ISATZ(IW+5)=IR                                                    280
      ISATZ(IW+6)=N1                                                    290
      ISATZ(IW+7)=N2                                                    300
      WRITE(NF'IBIB12)(ISATZ(I),I=1,400)                                310
      IW=IW+8                                                           320
      WRITE(NF'IR)(NBT(J),JNT(J),J=1,N1),(X(N),Y(N),N=1,N2)             330
      IR=IR+(2*N1+2*N2)/400+1                                           340
      GOTO 2000                                                         350
      ENTRY WRECS4(MIS,NAME1,NAME2)                                     360
      ASSIGN 104 TO LABEN                                               370
      IF(IW.GE.400) GOTO 3                                              380
  104 READ(NF'IBIB12)(ISATZ(I),I=1,400)                                 390
      ISATZ(IW)=MIS                                                     400
      ISATZ(IW+1)=NAME1                                                 410
      ISATZ(IW+2)=NAME2                                                 420
      ISATZ(IW+3)=0                                                     430
      ISATZ(IW+4)=0                                                     440
      ISATZ(IW+5)=IR                                                    450
      ISATZ(IW+6)=N1                                                    460
      ISATZ(IW+7)=0                                                     470
      WRITE(NF'IBIB12)(ISATZ(I),I=1,400)                                480
      IW=IW+8                                                           490
      WRITE(NF'IR)(B(N),N=1,N1)                                         500
       IR=IR+N2/400+1                                                   510
 2000 RETURN                                                            520
    3 IBIB12=IBIB12+1                                                   530
      IW=1                                                              540
      IF(IBIB12.LE.BIB12) GOTO LABEN,(1000,104)                         550
      WRITE(NO,2)                                                       560
    2 FORMAT(1X,'TABLE OF CONTENTS OF FILE 12 IS FULL.'/                570
     1' REDUCE THE NUMBER OF REACTION TYPES TO BE TRANSLATED OR INCREASE 580
     2 BIB12. THEN START THE JOB AGAIN.')                               590
      DO 11 IB=1,BIB12                                                  600
      WRITE(NO,10) IB                                                   610
   10 FORMAT('1TABLE OF CONTENTS OF FILE 12 PART ',I2,'.'/'+',38('_')// 620
     14X,'MIS  NAME',6X,'MF    MT',14X,'EL',18X,'EH',12X,'IR    N1    N2 630
     2'/)                                                               640
      READ(NF'IB) (ISATZ(I),I=1,400)                                    650
      DO 11 I=1,50                                                      660
      WRITE(NO,12) (ISATZ(J+8*(I-1)),J=1,3),(ISATZ(J+8*(I-1)),J=2,3),   670
     1             (ISATZ(J+8*(I-1)),J=4,8)                             680
   12 FORMAT(' ',I6,2X,2A4,I4,I6,4X,2E20.7,3I6)                         690
   11 CONTINUE                                                          700
      STOP                                                              710
      END                                                               720
```

```
      SUBROUTINE CROP(XL,XH,EPSMIN,LOSSTE,EPSMAX,SCRCLA,NUMSCR,LOF,NBT,  10
     *JNT,X,Y)                                                          20
C=====CROP A TAB1 RECORD=============================================== 30
C     A TAB1 RECORD IN /RECS/ AND ARRAYS NBT,JNT,X, AND Y IS CROPPED    40
C     ONLY THE PORTION BETWEEN X=XL AND X=XH IS RETAINED. PORTIONS OF   50
C     THE RANGE NOT DEFINED BY THE ORIGINAL TAB1 RECORD ARE SET TO ZERO. 60
C     UNNECESSARY POINTS ARE ELIMINATED IF THEY CAN BE                  70
C     PREDICTED BY INTERPOLATION BETWEEN ADJACENT POINTS TO             80
C     RELATIVE ACCURACY OF EPS.                                         90
C     THIS EPS IS EQUAL EPSMIN IF LOSSTE IS .FALSE. IN THIS CASE       100
C     THE VARIABLES EPSMAX,SCRCLA AND NUMSCR HAVE NO MEANING.          110
C     IF LOSSTE IS .TRUE., EPS HAS ALSO NORMALLY THE VALUE EPSMIN.     120
C     IT HAS THE VALUE EPSMAX IF TWO ADJACENT POINTS FULFILL THE       130
C     CONDITION (X2-X1)/X1<5.0E-6.                                     140
C     IN THE CASE OF LOSSTE EQUAL .TRUE. THE NUMBER OF POINTS WHICH    150
C     HAVE BEEN ELIMINATED, WILL BE COUNTED. FIRST SCRATCH-CLASSES     160
C     WERE BUILT IN THAT WAY: IF EPSMIN IS LESS EPSMAX FIVE INTERVALS  170
C     ARE MADE FROM 0.0 UP TO EPSMIN AND ANOTHER FIVE FROM EPSMIN TO   180
C     EPSMAX. IF EPSMIN IS EQUAL EPSMAX TEN INTERVALS ARE MADE FROM    190
C     0.0 TO EPSMIN (ORE EPSMAX). LATER THE INTERPOLATION ACCURACY     200
C     OF THE CROPPED POINTS WILL BE COMPUTED AND THE NUMBER OF THIS    210
C     POINTS ADDED UP IN THE CORRECT ARRAY ELEMENT NUMSCR. THE ARRAY   220
C     ELEMENT NUMSCR(I) GIVES THE NUMBER OF CROPPED POINTS THAT COULD  230
C     HAVE BEEN PREDICTED WITH AN RELATIVE ERROR IN THE RANGE          240
C     SCRCLA(I-1) < EPS <= SCRLA(I) . (SCRCLA(0) IS EQUAL 0.0)         250
C     LOF=0, NORMAL RETURN                                             260
C        =1, RECORD TOO LARGE, REPEAT WITH LARGER EPSMIN.              270
C-----ERROR STOP 312, TAB1 RECORD INCORRECT                           280
C     ERROR STOP 313, NBT, JNT TABLES TOO LARGE                        290
      DIMENSION NBT(1),JNT(1),X(1),Y(1),SCRCLA(10),NUMSCR(10)          300
      LOGICAL*4 LOSSTE                                                 310
      COMMON/RECS/MAT,MF,MT,C1,C2,L1,L2,N1,N2                          320
      COMMON/INOUT/NR1(125),NBX,N1X,N2X                                330
      LOF=0                                                            340
      IF(.NOT.LOSSTE) GOTO 100                                         350
      IF(EPSMIN.GT.EPSMAX) CALL ERROR(777)                             360
C     BILDUNG DER SCRATCHCLASSEN.                                      370
      IF(EPSMIN.EQ.EPSMAX) GOTO 310                                    380
      SCRCLA(5)=EPSMIN                                                 390
      H1=SCRCLA(5)/5.0                                                 400
      DO 311 I=1,4                                                     410
  311 SCRCLA(I)=I*H1                                                   420
      SCRCLA(10)=EPSMAX                                                430
      H1=(SCRCLA(10)-SCRCLA(5))/5.0                                    440
      DO 312 I=6,9                                                     450
  312 SCRCLA(I)=SCRCLA(5)+(I-5)*H1                                     460
      GOTO 315                                                         470
  310 H1=EPSMIN/10.0                                                   480
      SCRCLA(10)=EPSMIN                                                490
      DO 313 I=1,9                                                     500
  313 SCRCLA(I)=I*H1                                                   510
C     NULLSETZEN DER ZAEHLER.                                          520
  315 DO 314 I=1,10                                                    530
  314 NUMSCR(I)=0                                                      540
C     ENDE DER BERECHNUNG DER SCRATCHKLASSEN.                          550
C-----CROP UPPER END                                                  560
  100 IF(X(N2).LE.XH)GOTO200                                           570
      DO110N=1,N2                                                      580
      NP=N                                                             590
      IF(X(N).GE.XH)GOTO120                                            600
```

```
  110 CONTINUE
  120 IF(NP.LE.1)GOTO300
      DO130M=1,N1
      MP=M
      IF(NP.LE.NBT(M))GOTO140
  130 CONTINUE
      CALL ERROR(312)
  140 I=JNT(MP)
      CALL TERP1(X(NP-1),Y(NP-1),X(NP),Y(NP),XH,YP,I)
      X(NP)=XH
      Y(NP)=YP
      N2=NP
      N1=MP
C-----CROP LOWER END
  200 IF(X(1).GE.XL)GOTO400
      DO210N=1,N2
      NP=N
      IF(X(N).GT.XL)GOTO220
  210 CONTINUE
      GOTO300
  220 DO230M=1,N1
      MP=M
      IF(NP.LE.NBT(M))GOTO240
  230 CONTINUE
      CALL ERROR(312)
  240 I=JNT(MP)
      CALL TERP1(X(NP-1),Y(NP-1),X(NP),Y(NP),XL,YP,I)
      X(NP-1)=XL
      Y(NP-1)=YP
      ND=NP-2
      IF(ND.LE.0)GOTO400
      N2=N2-ND
      DO250N=1,N2
      K=N+ND
      X(N)=X(K)
  250 Y(N)=Y(K)
      DO260M=1,N1
  260 NBT(M)=NBT(M)-ND
      ND=MP-1
      IF(ND.LE.0)GOTO280
      N1=N1-ND
      DO270M=1,N1
      K=M+ND
      NBT(M)=NBT(K)
  270 JNT(M)=JNT(K)
  280 GOTO400
C-----ENTIRE RANGE ZERO
  300 X(1)=XL
      Y(1)=0.0
      X(2)=XH
      Y(2)=0.0
      N2=2
      NBT(1)=2
      JNT(1)=2
      N1=1
      GOTO900
C-----ELIMINATE UNNECESSARY POINTS
C-----CHANGED JULY 1974 BY E.STEIN.
C-----POINTS BETWEEN TWO POINTS ARE ONLY ELIMINATED IF ALL POINTS
C-----BETWEEN CAN BE OBTAINED BY INTERPOLATION WITH AN ERROR LESS EPS.
  400 IF(EPSMIN.LE.0.0.OR.N2.LE.2) GOTO 500
      K1=1
  401 K2=K1+2
C     LETZTES INTERVALL ERREICHT?
      IF(K2.GT.N2) GOTO 480
C     INTERVALLANFANG GLEICH INTERVALLENDE?
      IF(X(K1).EQ.X(K2)) GOTO 420
C     INTERPOLATIONSART UND BEREICH FUER INTERVALLENDPUNKT.
  402 N=K2
      ASSIGN 403 TO LAB490
      GOTO 490
  403 IK2=I
      MK2=M
      KI=K1+1
C     LAUFENDER PUNKT GLEICH INTERVALLANFANG ODER INTERVALLENDE?
C     ( UND Y-WERT DES LAUFENDEN PUNKTES UNGLEICH DEM Y-WERT
C       VON INTERVALLANFANG ODER INTERVALLENDE )
      IF((X(KI).EQ.X(K1).AND.Y(KI).NE.Y(K1)).OR.(X(KI).EQ.X(K2).AND.
     *Y(KI).NE.Y(K2))) GOTO 420
C     INTERPOLATIONSART UND BEREICH FUER LAUFENDEN PUNKT.
  404 N=KI
      ASSIGN 405 TO LAB490
      GOTO 490
  405 IKI=I
      MKI=M
C     BEI AENDERN VON INTERPOLATIONSART KANN KEIN PUNKT ELIMINIERT WERD.
      IF(IKI.NE.IK2.OR.Y(KI).LE.1.0E-10) GOTO 420
      IF(MKI.GT.MK2) GOTO 416
      CALL TERP1(X(K1),Y(K1),X(K2),Y(K2),X(KI),YP,IKI)
      EPS=EPSMIN
      IF(LOSSTE.AND.(
     *(((X(KI)-X(KI-1))/X(KI)).LT.5.0E-6).OR.
     *(((X(KI+1)-X(KI))/X(KI+1)).LT.5.0E-6))) EPS=EPSMAX
      IF(ABS((Y(KI)-YP)/Y(KI)).GT.EPS) GOTO 420
C     NAECHSTER ZU ELIMINIERENDER PUNKT IM INTERVALL K1...K2.
      KI=KI+1
      IF(KI.LT.K2) GOTO 404
C     VERGROESSERN DES INTERVALLS.
      K2=K2+1
      IF(K2.LE.N2) GOTO 402
C     INTERVALL WAR ZU GROSS. LETZTER PUNKT ERREICHT.
C     GENAUIGKEIT NICHT ERFUELLT. INTERVALL VERKLEINERN UND
C     EVENTUELL PUNKTE ELIMINIEREN.
  420 K2=K2-1
      IF(K2.GT.(K1+1)) GOTO 406
C     INTERVALL NACH OBEN VERSCHIEBEN.
      K1=K1+1
      GOTO 401
C     ELIMINIEREN DER PUNKTE K1+1 BIS K2-1
  406 K1P1=K1+1
```

```
      K2M1=K2-1                                                          1710
      KANZ=K2-K1-1                                                       1720
      IF(.NOT.LOSSTE) GOTO 309                                           1730
C     EINORDNEN DER PUNKTE IN DIE ZUGEHOERIGE SCRATCHCLASSE.             1740
      KI=K1P1                                                            1750
  301 IF(KI.GT.K2M1) GOTO 309                                            1760
      N=KI                                                               1770
      ASSIGN 302 TO LAB490                                               1780
      GOTO 490                                                           1790
  302 IKI=I                                                              1800
      CALL TERP1(X(K1),Y(K1),X(K2),Y(K2),X(KI),YP,IKI)                   1810
      EPS=ABS((Y(KI)-YP)/Y(KI))                                          1820
      IF(EPS.GT.SCRCLA(10)) CALL ERROR(777)                              1830
      DO 303 K=1,10                                                      1840
      IF(K.EQ.10) GOTO 304                                               1850
      IF(EPS.LE.SCRCLA(10-K)) GOTO 303                                   1860
  304 NUMSCR(11-K)=NUMSCR(11-K)+1                                        1870
      GOTO 305                                                           1880
  303 CONTINUE                                                           1890
  305 KI=KI+1                                                            1900
      GOTO 301                                                           1910
  309 DO 407 K=K2,N2                                                     1920
      X(K-KANZ)=X(K)                                                     1930
  407 Y(K-KANZ)=Y(K)                                                     1940
      N2=N2-KANZ                                                         1950
      K2=K2-KANZ                                                         1960
C     VERAENDERN DER INTERPOLATIONSTABELLE.                              1970
      N=K1P1                                                             1980
      ASSIGN 409 TO LAB490                                               1990
      GOTO 490                                                           2000
  409 MK1P1=M                                                            2010
      N=K2M1                                                             2020
      ASSIGN 421 TO LAB490                                               2030
      GOTO 490                                                           2040
  421 MK2M1=M                                                            2050
      IF(MK1P1.NE.MK2M1.OR.MK1P1.GT.N1) GOTO 416                         2060
      DO 410 MM=MK1P1,N1                                                 2070
  410 NBT(MM)=NBT(MM)-KANZ                                               2080
C     LOESCHEN VON IDENTISCHEN EINTRAEGEN IN DER INTERPOLATIONSTABELLE.  2090
      IF(N1.LE.1) GOTO 412                                               2100
      K=2                                                                2110
  414 IF(NBT(K-1).NE.NBT(K)) GOTO 411                                    2120
      IF(JNT(K-1).NE.JNT(K)) GOTO 416                                    2130
      DO 413 KK=K,N1                                                     2140
      NBT(KK-1)=NBT(KK)                                                  2150
  413 JNT(KK-1)=JNT(KK)                                                  2160
      N1=N1-1                                                            2170
      GOTO 415                                                           2180
  411 K=K+1                                                              2190
  415 IF(K.LE.N1) GOTO 414                                               2200
  412 K1=K2                                                              2210
      GOTO 401                                                           2220
  490 CONTINUE                                                           2230
C     DER ZU N GEHOERENDE INTERPOLATIONSBEREICH M UND DIE ZU N           2240
C     GEHOERENDE INTERPOLATIONSART I WIRD GESUCHT.                       2250
      M=1                                                                2260
  491 IF(N.LE.NBT(M)) GOTO 492                                           2270
      M=M+1                                                              2280
      IF(M.LE.N1) GOTO 491                                               2290
      CALL ERROR(312)                                                    2300
  492 I=JNT(M)                                                           2310
      GOTO LAB490,(403,405,409,421,302)                                  2320
  416 CALL ERROR(777)                                                    2330
  480 CONTINUE                                                           2340
C-----ADD ZEROS TO UPPER END                                             2350
  500 IF(X(N2).GE.XH)GOTO600                                             2360
      IF(N2+2.GT.N2X)GOTO700                                             2370
      X(N2+1)=X(N2)                                                      2380
      Y(N2+1)=0.0                                                        2390
      X(N2+2)=XH                                                         2400
      Y(N2+2)=0.0                                                        2410
      N2=N2+2                                                            2420
      IF(JNT(N1).NE.2)GOTO510                                            2430
      NBT(N1)=N2                                                         2440
      GOTO520                                                            2450
  510 IF(N1+1.GT.N1X)CALL ERROR(313)                                     2460
      N1=N1+1                                                            2470
      NBT(N1)=N2                                                         2480
      JNT(N1)=2                                                          2490
  520 CONTINUE                                                           2500
C-----ADD ZEROS TO LOWER END                                             2510
  600 IF(X(1).LE.XL)GOTO800                                              2520
      IF(N2+2.GT.N2X)GOTO700                                             2530
      DO610N=1,N2                                                        2540
      K=N2-N+1                                                           2550
      X(K+2)=X(K)                                                        2560
  610 Y(K+2)=Y(K)                                                        2570
      X(1)=XL                                                            2580
      Y(1)=0.0                                                           2590
      X(2)=X(3)                                                          2600
      Y(2)=0.0                                                           2610
      N2=N2+2                                                            2620
      DO620M=1,N1                                                        2630
  620 NBT(M)=NBT(M)+2                                                    2640
      IF(JNT(1).EQ.2)GOTO640                                             2650
      IF(N1+1.GT.N1X)CALL ERROR(313)                                     2660
      DO630M=1,N1                                                        2670
      K=N1-M+1                                                           2680
      NBT(K+1)=NBT(K)                                                    2690
  630 JNT(K+1)=JNT(K)                                                    2700
      NBT(1)=3                                                           2710
      JNT(1)=2                                                           2720
      N1=N1+1                                                            2730
  640 GOTO800                                                            2740
C-----OVERFLOW                                                           2750
  700 LOF=1                                                              2760
C-----FINISHED                                                           2770
  800 RETURN                                                             2780
      END                                                                2790
```

```
      SUBROUTINE LUCY(I,BK)                                              10
      COMMON/INOUT/NO,NR1(127),N                                         20
      DIMENSION BK(1)                                                    30
      IF(I.EQ.2) GOTO 2                                                  40
      IF(I.EQ.3) GOTO 3                                                  50
      CALL ERROR(203)                                                    60
C RESOLVED RESONANCE FORMAT                                              70
    2 WRITE(NO,5)(BK(J),J=1,N)                                           80
    5 FORMAT(6E15.6/5E15.6/)                                             90
      GOTO 7                                                            100
C STATISTICAL DATA FORMAT                                               110
    3 WRITE(NO,6)(BK(J),J=1,N)                                          120
    6 FORMAT(8E15.6)                                                    130
    7 RETURN                                                            140
      END                                                               150
```

```
      SUBROUTINE RRECS(MIS,NAME1,NAME2,EL,EH,LNT)                        10
      INTEGER BIB12                                                      20
      COMMON/INOUT/NR1(78),BIB12,NR2(46),NBX,N1X,N2X                     30
      COMMON/RECS/MAT,MF,MT,C1,C2,L1,L2,N1,N2                            40
      DIMENSION NBT(1),JNT(1),X(1),Y(1),B(1)                             50
      DIMENSION ISATZ(400),XSATZ(400)                                    60
      EQUIVALENCE (ISATZ(1),XSATZ(1))                                    70
      IT=0                                                               80
      GOTO 6                                                             90
C                                                                       100
C                                                                       110
      ENTRY RRECS4(MIS,NAME1,NAME2,LNT)                                 120
C                                                                       130
C                                                                       140
      IT=1                                                              150
    6 NF=12                                                             160
      LNT=0                                                             170
      IBIB12=1                                                          180
    8 IW=1                                                              190
      READ(NF'IBIB12)(ISATZ(I),I=1,400)                                 200
    3 IF(ISATZ(IW).NE.MIS)GOTO 1                                        210
      IF(ISATZ(IW+1).NE.NAME1)GOTO1                                     220
      IF(ISATZ(IW+2).NE.NAME2)GOTO1                                     230
      GOTO 5                                                            240
    1 IW=IW+8                                                           250
      IF(IW.LT.400) GOTO 3                                              260
      IBIB12=IBIB12+1                                                   270
      IF(IBIB12.LE.BIB12) GOTO 8                                        280
      GOTO 2                                                            290
    5 EL=XSATZ(IW+3)                                                    300
      EH=XSATZ(IW+4)                                                    310
      IR=ISATZ(IW+5)                                                    320
      N1=ISATZ(IW+6)                                                    330
      N2=ISATZ(IW+7)                                                    340
      IF(IT.EQ.1)GOTO 7                                                 350
      IF(N1.GT.N1X) CALL ERROR(103)                                     360
      IF(N2.GT.N2X) CALL ERROR(105)                                     370
      READ(NF'IR)       (NBT(J),JNT(J),J=1,N1),(X(N),Y(N),N=1,N2)       380
      RETURN                                                            390
    7 IF(N1.GT.NBX) CALL ERROR(104)                                     400
      READ(NF'IR)(B(N),N=1,N1)                                          410
      RETURN                                                            420
    2 LNT=1                                                             430
      RETURN                                                            440
      ENTRY IRRECS(NBT,JNT,X,Y,B)                                       450
      RETURN                                                            460
      END                                                               470
```

```
      SUBROUTINE PRTPID(NT,B)                                            10
C=====PRINT TPID RECORD ==============================================   20
      COMMON/RECS/MAT,NR1(8),NS                                          30
      DIMENSION B(1)                                                     40
      WRITE (NT,20) (B(N),N=1,17), MAT                                   50
   20 FORMAT(5HOTPID,82X,5HLABEL/10X,16A4,A2,9X,I7)                      60
      NS=NS+1                                                            70
      RETURN                                                             80
      END                                                                90
```

```
      SUBROUTINE PUTPID(NT,B)                                            10
C=====PUNCH TPID RECORD ==============================================   20
      COMMON/RECS/MAT,NR1(8),NS                                          30
      DIMENSION B(1)                                                     40
   10 NZ=0                                                               50
      WRITE (NT,20) (B(N),N=1,17), MAT,NZ,NZ,NZ                          60
   20 FORMAT(16A4,A2,I4,I2,I3,I5)                                        70
      NS=NS+1                                                            80
      RETURN                                                             90
      END                                                               100
```

```
      SUBROUTINE RREC(JT,NT,MODE,T)                                      10
C     READ ONE RECORD FROM ENDF/B TAPE                                   20
C     JT  = RECORD TYPE, 1-CONT, 2-LIST, 3-TAB1, 4-TAB2, 5-HOL LIST      30
C                        6-TPID                                          40
C     NT  = INPUT TAPE NUMBER.                                           50
C     MODE=1 BINARY TAPE, STANDARD ARRANGEMENT                           60
C         =2 BINARY TAPE, ALTERNATE ARRANGEMENT                          70
C         =3 BCD CARD IMAGE TAPE                                         80
C     T   = TEMPERATURE(KELVIN). IF T.GE.0 AND RECORD HAS A TEMPERATURE  90
C           DEPENDENCE, DATA WILL BE EVALUATED AT T.  IF T.LT.0, ONLY    100
C           FIRST RECORD WILL BE READ.                                   110
C-----ERROR STOP  99, NT NOT DEFINED                                     120
C     ERROR STOP 100, JT OUT OF RANGE 1-6                                130
```

```
C      ERROR STOP 101, MODE OUT OF RANGE 1-3                                    140
C      ERROR STOP 102, T NOT IN RANGE GIVEN IN DATA                             150
C      ERROR STOP 103, INTERPOLATION TABLE TOO LONG OR 0                        160
C      ERROR STOP 104, LIST TOO LONG OR 0                                       170
C      ERROR STOP 105, TABULATION TOO LONG OR .LT.2                             180
C      ERROR STOP 106, IMPROPER TEMPERATURE DEPENDENCE                          190
C-----ERROR STOP 107, MAT, MF, MT INCORRECT FOR JT=2, 3, 4, 5                   200
       COMMON/RECS/MAT,MF,MT,C1,C2,L1,L2,N1,N2                                  210
       DIMENSION NBT(1),JNT(1),X(1),Y(1),B(1)                                   220
       COMMON/INOUT/NRL(125),NBX,N1X,N2X                                        230
C-----PRELIMINARY TESTS                                                         240
  100  IF(NT.LE.0)CALL ERROR(99)                                                250
       IF(JT.LT.1.OR.JT.GT.6) CALL ERROR(100)                                   260
       IF(MODE.LT.1.OR.MODE.GT.3) CALL ERROR(101)                               270
  140  GOTO(200,300,400,500,570,590),JT                                         280
C-----JT=1, CONT RECORD                                                         290
  200  GOTO(210,220,230),MODE                                                   300
  210  READ(NT)MAT,MF,MT,C1,C2,L1,L2,N1,N2                                      310
       GOTO240                                                                  320
  220  READ(NT)MF,MAT,MT,C1,C2,L1,L2,N1,N2                                      330
       GOTO240                                                                  340
  230  READ(NT,235)C1,C2,L1,L2,N1,N2,MAT,MF,MT                                  350
  235  FORMAT(2E11.0,4I11,I4,I2,I3)                                             360
  240  GOTO900                                                                  370
C-----JT=2, LIST RECORD                                                         380
  300  GOTO(310,320,330),MODE                                                   390
  310  READ(NT)MAT,MF,MT,C1,C2,L1,L2,N1,N2,(B(N),N=1,N1)                        400
       GOTO340                                                                  410
  320  READ(NT)MF,MAT,MT,C1,C2,L1,L2,N1,N2,(B(N),N=1,N1)                        420
       GOTO340                                                                  430
  330  READ(NT,335)C1,C2,L1,L2,N1,N2,MAT,MF,MT,(B(N),N=1,N1)                    440
  335  FORMAT(2E11.0,4I11,I4,I2,I3/(6E11.0))                                    450
  340  IF(N1.GT.NBX.OR.N1.LT.1) CALL ERROR(104)                                 460
       IF(L1.LE.0.OR.T.LT.0.0.OR.MF.EQ.2) GOTO 800                              470
       NP=N1                                                                    480
       DO370N=1,NP                                                              490
  370  Y(N)=B(N)                                                                500
       GOTO600                                                                  510
C-----JT=3, TAB1 RECORD                                                         520
  400  GOTO(410,420,430),MODE                                                   530
  410  READ(NT)MAT,MF,MT,C1,C2,L1,L2,N1,N2,(NBT(N),JNT(N),N=1,N1),(X(N),Y       540
      1(N),N=1,N2)                                                              550
       GOTO450                                                                  560
  420  READ(NT)MF,MAT,MT,C1,C2,L1,L2,N1,N2,(NBT(N),JNT(N),N=1,N1),(X(N),Y       570
      1(N),N=1,N2)                                                              580
       GOTO450                                                                  590
  430  READ(NT,435)C1,C2,L1,L2,N1,N2,MAT,MF,MT,(NBT(N),JNT(N),N=1,N1)           600
  435  FORMAT(2E11.0,4I11,I4,I2,I3/(6I11))                                      610
       READ(NT,445)(X(N),Y(N),N=1,N2)                                           620
  445  FORMAT(6E11.0)                                                           630
  450  IF(N1.GT.N1X.OR.N1.LT.1) CALL ERROR(103)                                 640
       IF(N2.GT.N2X.OR.N2.LT.2) CALL ERROR(105)                                 650
       IF(L1.EQ.0.OR.T.LT.0.0.OR.MF.EQ.2) GOTO 800                              660
       NP=N2                                                                    670
       GOTO600                                                                  680
C-----JT=4, TAB2 RECORD                                                         690
  500  GOTO(510,520,530),MODE                                                   700
  510  READ(NT)MAT,MF,MT,C1,C2,L1,L2,N1,N2,(NBT(N),JNT(N),N=1,N1)               710
       GOTO540                                                                  720
  520  READ(NT)MF,MAT,MT,C1,C2,L1,L2,N1,N2,(NBT(N),JNT(N),N=1,N1)               730
       GOTO540                                                                  740
  530  READ(NT,535)C1,C2,L1,L2,N1,N2,MAT,MF,MT,(NBT(N),JNT(N),N=1,N1)           750
  535  FORMAT(2E11.0,4I11,I4,I2,I3/(6I11))                                      760
  540  IF(N1.GT.N1X.OR.N1.LT.1) CALL ERROR(103)                                 770
  560  IF(N2.GT.N2X.OR.N2.LT.2) CALL ERROR(105)                                 780
       GO TO 800                                                                790
C-----JT=5, HOL LIST                                                            800
  570  GOTO(300,300,580),MODE                                                   810
  580  READ(NT,585)C1,C2,L1,L2,NCD,N2,MAT,MF,MT                                 820
  585  FORMAT(2E11.0,4I11,I4,I2,I3)                                             830
       N1=17*NCD                                                                840
       IF(N1.LE.0.OR.N1.GT.NBX) CALL ERROR(104)                                 850
       READ(NT,588)(B(N),N=1,N1)                                                860
  588  FORMAT(16A4,A2)                                                          870
       GOTO800                                                                  880
C-----JT=6, TPID RECORD                                                         890
  590  GO TO (591,592,593), MODE                                                900
  591  READ (NT) MAT,MF,MT,(B(N),N=1,17)                                        910
       GO TO 595                                                                920
  592  READ (NT) MF,MAT,MT, (B(N),N=1,17)                                       930
       GO TO 595                                                                940
  593  READ (NT,594) (B(N),N=1,17), MAT,MF,MT                                   950
  594  FORMAT(16A4,A2,I4,I2,I3)                                                 960
  595  GO TO 900                                                                970
C-----TEMPERATURE DEPENDENCE                                                    980
  600  IF(L1.LT.1) CALL ERROR(106)                                              990
       LT=L1                                                                   1000
       TA=C1                                                                   1010
       IF(T.LT.TA) CALL ERROR(102)                                             1020
       IF(T.EQ.TA)GOTO720                                                      1030
  650  GOTO(660,660,670),MODE                                                  1040
  660  READ(NT)Z,Z,Z,TB,Z,IC,Z,NP1,Z,(B(N),N=1,NP1)                           1050
       GOTO680                                                                 1060
  670  READ(NT,675)TB,IC,NP1,(B(N),N=1,NP1)                                    1070
  675  FORMAT(E11.0,11X,I11,11X,I11/(6E11.0))                                  1080
  680  LT=LT-1                                                                 1090
       IF(NP1.NE.NP.OR.TB.LT.TA.OR.IC.LT.1.OR.IC.GT.5) CALL ERROR(106)         1100
       IF(T.LE.TB)GOTO700                                                      1110
       TA=TB                                                                   1120
       DO690N=1,NP                                                             1130
  690  Y(N)=B(N)                                                               1140
       IF(LT.LE.0) CALL ERROR(106)                                             1150
       GOTO650                                                                 1160
  700  DO710N=1,NP                                                             1170
       CALL TERP1(TA,Y(N),TB,B(N),T,Z,IC)                                      1180
  710  Y(N)=Z                                                                  1190
  720  GOTO(750,730,750,750),JT                                                1200
  730  DO740N=1,NP                                                             1210
  740  B(N)=Y(N)                                                               1220
  750  IF(LT.LE.0)GOTO790                                                      1230
```

```
      DO780L=1,LT                                                    1240
      GOTO(760,760,770),MODE                                         1250
  760 READ(NT)Z,Z,Z,TB,Z,IC,Z,NP1,Z,(B(N),N=1,NP1)                   1260
      GOTO780                                                        1270
  770 READ(NT,675)TB,IC,NP1,(B(N),N=1,NP1)                           1280
  780 CONTINUE                                                       1290
  790 GOTO800                                                        1300
C-----CHECK MAT, MF, AND MT FOR JT=2, 3, 4, 5                        1310
  800 IF(MAT.LE.0.OR.MF.LE.0.OR.MT.LE.0) CALL ERROR(107)             1320
C-----FINISHED                                                       1330
  900 RETURN                                                         1340
      ENTRY IRREC(NBT,JNT,X,Y,B)                                     1350
      GOTO 900                                                       1360
      END                                                            1370
```

```
      SUBROUTINE PRHOL(NT,B)                                          10
C=====PRINT HOLLERITH LIST RECORD=======================================  20
      COMMON/RECS/MAT,MF,MT,C1,C2,L1,L2,N1,N2,NS                      30
      DIMENSION B(1)                                                  40
      NCD=N1/17                                                       50
   10 WRITE(NT,20) C1,C2,L1,L2,NCD,N2,MAT,MF,MT,NS                    60
   20 FORMAT('0COMM',9X,'C1  ',9X,'C2',11X,'L1',11X,'L2 ',9X,'NCD',11X,  70
     1'N2',8X,'MAT  MF   MT    SEQ'/E20.4,E13.4,I9,3I13,I11,I4,I5,I7)    80
      NS=NS+1                                                         90
      NI=1                                                           100
      DO 40 NC=1,NCD                                                 110
      NF=NI+16                                                       120
      WRITE(NT,30) (B(N),N=NI,NF),MAT,MF,MT,NS                       130
   30 FORMAT(1H /(10X,16A4,A2,9X,I7,I4,I5,I7))                       140
      NS=NS+1                                                        150
   40 NI=NI+17                                                       160
      RETURN                                                         170
      END                                                            180
```

```
      SUBROUTINE PUHOL(NT,B)                                          10
C=====PUNCH HOLLERITH LIST RECORD=======================================  20
      COMMON/RECS/MAT,MF,MT,C1,C2,L1,L2,N1,N2,NS                      30
      DIMENSION F(6),S(6),J(6),B(1)                                   40
   10 NCD=N1/17                                                       50
      CALL CXFP(C1,F(1),S(1),J(1))                                    60
      CALL CXFP(C2,F(2),S(2),J(2))                                    70
      WRITE(NT,20) F(1),S(1),J(1),F(2),S(2),J(2),L1,L2,NCD,N2,MAT,MF,MT,  80
     1NS                                                              90
   20 FORMAT(2(F8.5,A1,I2),4I11,I4,I2,I3,I5)                         100
      NS=NS+1                                                        110
      NI=1                                                           120
      DO 40 NC=1,NCD                                                 130
      NF=NI+16                                                       140
      WRITE(NT,30) (B(N),N=NI,NF),MAT,MF,MT,NS                       150
```

```
   30 FORMAT(16A4,A2,I4,I2,I3,I5)                                    160
      NS=NS+1                                                        170
   40 NI=NI+17                                                       180
      RETURN                                                         190
      END                                                            200
```

```
      SUBROUTINE PRCONT(NT)                                           10
C=====PRINT CONT TYPE RECORD ============================================  20
      COMMON/RECS/MAT,MF,MT,C1,C2,L1,L2,N1,N2,NS                      30
   10 IF(MAT.LT.0) GO TO 40                                           40
      IF(MAT.EQ.0) GO TO 100                                          50
      IF(MF.EQ.0) GO TO 80                                            60
      IF(MT.EQ.0) GO TO 60                                            70
   20 WRITE(NT,30)C1,C2,L1,L2,N1,N2,MAT,MF,MT,NS                      80
   30 FORMAT('0CONT',9X,'C1',11X,'C2',11X,'L1',11X,'L2',11X,'N1',11X,  90
     1'N2',9X,'MAT  MF   MT    SEQ'/E20.4,E13.4,I9,3I13,I11,I4,I5,I7)  100
      GO TO 120                                                      110
   40 WRITE(NT,50) MAT,MF,MT,NS                                      120
   50 FORMAT(5HOTEND,19X,11HEND OF TAPE,53X,I4,I4,I5,I7)             130
      GO TO 120                                                      140
   60 WRITE(NT,70) MAT,MF,MT,NS                                      150
   70 FORMAT(5HOSEND,4X,14HEND OF SECTION,65X,I4,I4,I5,I7)           160
      GO TO 120                                                      170
   80 WRITE(NT,90) MAT,MF,MT,NS                                      180
   90 FORMAT(5HOFEND,9X,11HEND OF FILE,63X,I4,I4,I5,I7)              190
      GO TO 120                                                      200
  100 WRITE(NT,110) MAT,MF,MT,NS                                     210
  110 FORMAT(5HOMEND,14X,15HEND OF MATERIAL,54X,I4,I4,I5,I7)         220
  120 NS=NS+1                                                        230
      RETURN                                                         240
      END                                                            250
```

```
      SUBROUTINE PUCONT(NT)                                           10
C=====PUNCH CONT TYPE RECORD ============================================  20
      COMMON/RECS/MAT,MF,MT,C1,C2,L1,L2,N1,N2,NS                      30
   10 IF(MT.LE.0) GO TO 40                                            40
   20 CALL CXFP(C1,F1,S1,J1)                                          50
      CALL CXFP(C2,F2,S2,J2)                                          60
      WRITE(NT,30)F1,S1,J1,F2,S2,J2,L1,L2,N1,N2,MAT,MF,MT,NS          70
   30 FORMAT(2(F8.5,A1,I2),4I11,I4,I2,I3,I5)                          80
      GO TO 80                                                        90
   40 WRITE(NT,50)MAT,MF,MT,NS                                       100
   50 FORMAT(66X,I4,I2,I3,I5)                                        110
   80 NS=NS+1                                                        120
      RETURN                                                         130
      END                                                            140
```

```
      SUBROUTINE PRLIST(NT,B)                                           10
C=====PRINT LIST TYPE RECORD=============================================  20
      COMMON/RECS/MAT,MF,MT,C1,C2,L1,L2,N1,N2,NS                        30
      DIMENSION B(1)                                                    40
   10 WRITE(NT,20) C1,C2,L1,L2,N1,N2,MAT,MF,MT,NS                       50
   20 FORMAT('0LIST',9X,'C1',11X,'C2',11X,'L1',11X,'L2',11X,'N1',11X,   60
     1'N2',8X,'MAT  MF   MT    SEQ'/E20.4,E13.4,I9,3I13,I11,I4,I5,I7)   70
      NS=NS+1                                                           80
      WRITE(NT,30)                                                      90
   30 FORMAT('  (N)',7X,'B(N)',9X,'B(N+1)',7X,'B(N+2)',7X,'B(N+3)',7X,  100
     1'B(N+4)',7X,'B(N+5)')                                             110
      NI=1                                                              120
      NE=6                                                              130
   40 IF(NE-N1) 50,50,80                                                140
   50 WRITE(NT,60)NI,(B(N),N=NI,NE),MAT,MF,MT,NS                        150
   60 FORMAT(2H (,I3,2H) ,6E13.4,I7,I4,I5,I7)                           160
      NS=NS+1                                                           170
      IF(NE-N1) 70,200,200                                              180
   70 NI=NI+6                                                           190
      NE=NI+5                                                           200
      GO TO 40                                                          210
   80 NE=N1-NI+1                                                        220
      GO TO(90,110,130,150,170),NE                                      230
   90 WRITE(NT,100)NI,(B(N),N=NI,N1),MAT,MF,MT,NS                       240
  100 FORMAT(2H (,I3,2H) , E13.4,65X,I7,I4,I5,I7)                       250
      GO TO 190                                                         260
  110 WRITE(NT,120)NI,(B(N),N=NI,N1),MAT,MF,MT,NS                       270
  120 FORMAT(2H (,I3,2H) ,2E13.4,52X,I7,I4,I5,I7)                       280
      GO TO 190                                                         290
  130 WRITE(NT,140)NI,(B(N),N=NI,N1),MAT,MF,MT,NS                       300
  140 FORMAT(2H (,I3,2H) ,3E13.4,39X,I7,I4,I5,I7)                       310
      GO TO 190                                                         320
  150 WRITE(NT,160)NI,(B(N),N=NI,N1),MAT,MF,MT,NS                       330
  160 FORMAT(2H (,I3,2H) ,4E13.4,26X,I7,I4,I5,I7)                       340
      GO TO 190                                                         350
  170 WRITE(NT,180)NI,(B(N),N=NI,N1),MAT,MF,MT,NS                       360
  180 FORMAT(2H (,I3,2H) ,5E13.4,13X,I7,I4,I5,I7)                       370
  190 NS=NS+1                                                           380
  200 RETURN                                                            390
      END                                                               400
```

```
      SUBROUTINE PULIST(NT,B)                                           10
C=====PUNCH LIST TYPE RECORD=============================================  20
      COMMON/RECS/MAT,MF,MT,C1,C2,L1,L2,N1,N2,NS                        30
      DIMENSION F(6),S(6),J(6),B(1)                                     40
   10 CALL CXFP(C1,F(1),S(1),J(1))                                      50
      CALL CXFP(C2,F(2),S(2),J(2))                                      60
      WRITE(NT,20)F(1),S(1),J(1),F(2),S(2),J(2),L1,L2,N1,N2,MAT,MF,MT,NS 70
   20 FORMAT(2(F8.5,A1,I2),4I11,I4,I2,I3,I5)                            80
      NS=NS+1                                                           90
      NI=1                                                              100
   30 NE=N1-NI+1                                                        110
      KX=6                                                              120
      IF(NE-6) 40,50,50                                                 130
   40 KX=NE                                                             140
   50 DO 60 K=1,KX                                                      150
      N=NI+K-1                                                          160
   60 CALL CXFP(B(N),F(K),S(K),J(K))                                    170
      IF(NE-6) 100,70,70                                                180
   70 WRITE(NT,80)(F(K),S(K),J(K),K=1,6),MAT,MF,MT,NS                   190
   80 FORMAT(6(F8.5,A1,I2),I4,I2,I3,I5)                                 200
      NS=NS+1                                                           210
      IF(NE-6) 220,220,90                                               220
   90 NI=NI+6                                                           230
      GO TO 30                                                          240
  100 GO TO (110,130,150,170,190),NE                                    250
  110 WRITE(NT,120)(F(K),S(K),J(K),K=1,1),MAT,MF,MT,NS                  260
  120 FORMAT(1(F8.5,A1,I2),55X,I4,I2,I3,I5)                             270
      GO TO 210                                                         280
  130 WRITE(NT,140)(F(K),S(K),J(K),K=1,2),MAT,MF,MT,NS                  290
  140 FORMAT(2(F8.5,A1,I2),44X,I4,I2,I3,I5)                             300
      GO TO 210                                                         310
  150 WRITE(NT,160)(F(K),S(K),J(K),K=1,3),MAT,MF,MT,NS                  320
  160 FORMAT(3(F8.5,A1,I2),33X,I4,I2,I3,I5)                             330
      GO TO 210                                                         340
  170 WRITE(NT,180)(F(K),S(K),J(K),K=1,4),MAT,MF,MT,NS                  350
  180 FORMAT(4(F8.5,A1,I2),22X,I4,I2,I3,I5)                             360
      GO TO 210                                                         370
  190 WRITE(NT,200)(F(K),S(K),J(K),K=1,5),MAT,MF,MT,NS                  380
  200 FORMAT(5(F8.5,A1,I2),11X,I4,I2,I3,I5)                             390
  210 NS=NS+1                                                           400
  220 RETURN                                                            410
      END                                                               420
```

```
      SUBROUTINE PRTAB1(NT,NBT,JNT,X,Y)                                 10
C=====PRINT TAB1 TYPE RECORD=============================================  20
      COMMON/RECS/MAT,MF,MT,C1,C2,L1,L2,N1,N2,NS                        30
      DIMENSION NBT(1),JNT(1),X(1),Y(1)                                 40
      NR=N1                                                             50
      NP=N2                                                             60
   10 WRITE(NT,20)C1,C2,L1,L2,NR,NP,MAT,MF,MT,NS                        70
   20 FORMAT('0TAB1',9X,'C1',11X,'C2',11X,'L1',11X,'L2',11X,'NR',11X,   80
     1'NP',9X,'MAT  MF   MT    SEQ'/E20.4,E13.4,I9,3I13,I12,I4,I5,I7/)  90
      NS=NS+1                                                           100
      WRITE(NT,30)                                                      110
   30 FORMAT('  (N)',8X,'NBT(N)',7X,'INT(N)',7X,'NBT(N+1)',5X,         120
     1'INT(N+1)',5X,'NBT(N+2)',5X,'INT(N+2)')                           130
      NI=1                                                              140
      NE=3                                                              150
   40 IF(NE-NR)50,50,80                                                 160
   50 WRITE(NT,60)NI,(NBT(N),JNT(N),N=NI,NE),MAT,MF,MT,NS               170
   60 FORMAT(' (',I4,')',I10,5I13,I11,I4,I5,I7)                         180
      NS=NS+1                                                           190
      IF(NE-NR)70,140,140                                               200
```

```
   70 NI=NI+3                                                          210
      NF=NI+2                                                          220
      GOTO40                                                           230
   80 IF(NF-NR-1)140,110,90                                            240
   90 WRITE(NT,100)NI,(NBT(N),JNT(N),N=NI,NR),MAT,MF,MT,NS             250
  100 FORMAT(' (',I4,')',I10,I13,52X,I11,I4,I5,I7)                     260
      GOTO130                                                          270
  110 WRITE(NT,120)NI,(NBT(N),JNT(N),N=NI,NR),MAT,MF,MT,NS             280
  120 FORMAT(' (',I4,')',I10,3I13,26X,I11,I4,I5,I7)                    290
  130 NS=NS+1                                                          300
  140 WRITE(NT,150)                                                    310
  150 FORMAT('0  (N)',7X,'X(N)',9X,'Y(N)',9X,'X(N+1)',7X,'Y(N+1)',     320
     17X,'X(N+2)',7X,'Y(N+2)')                                         330
      NI=1                                                             340
      NF=3                                                             350
  160 IF(NF-NP)170,170,200                                             360
  170 WRITE(NT,180)NI,(X(N),Y(N),N=NI,NF),MAT,MF,MT,NS                 370
  180 FORMAT(2H (,I5,1H),1P6E13.5,I7,I4,I5,I7)                         380
      NS=NS+1                                                          390
      IF(NF-NP)190,260,260                                             400
  190 NI=NI+3                                                          410
      NF=NI+2                                                          420
      GOTO160                                                          430
  200 IF(NF-NP-1)260,230,210                                           440
  210 WRITE(NT,220)NI,(X(N),Y(N),N=NI,NP),MAT,MF,MT,NS                 450
  220 FORMAT(2H (,I5,1H),1P2E13.5,52X,I7,I4,I5,I7)                     460
      GOTO250                                                          470
  230 WRITE(NT,240)NI,(X(N),Y(N),N=NI,NP),MAT,MF,MT,NS                 480
  240 FORMAT(2H (,I5,1H),1P4E13.5,26X,I7,I4,I5,I7)                     490
  250 NS=NS+1                                                          500
  260 RETURN                                                           510
      END                                                              520
```

```
      SUBROUTINE PUTAB1(NT,NBT,JNT,X,Y)                                 10
C=====PUNCH TAB1 TYPE RECORD ==========================================  20
      COMMON/RECS/MAT,MF,MT,C1,C2,L1,L2,N1,N2,NS                        30
      DIMENSION F(6),S(6),J(6),NBT(1),JNT(1),X(1),Y(1)                  40
      NR=N1                                                             50
      NP=N2                                                             60
   10 CALL CXFP(C1,F(1),S(1),J(1))                                      70
      CALL CXFP(C2,F(2),S(2),J(2))                                      80
      WRITE(NT,20)F(1),S(1),J(1),F(2),S(2),J(2),L1,L2,NR,NP,MAT,MF,MT,NS  90
   20 FORMAT(F8.5,A1,I2,F8.5,A1,I2,4I11,I4,I2,I3,I5)                   100
      NS=NS+1                                                          110
      NI=1                                                             120
      NF=3                                                             130
   30 IF(NF-NR) 40,40,70                                               140
   40 WRITE(NT,50)(NBT(N),JNT(N),N=NI,NF),MAT,MF,MT,NS                 150
   50 FORMAT(6I11,I4,I2,I3,I5)                                         160
      NS=NS+1                                                          170
      IF(NF-NR) 60,130,130                                             180
   60 NI=NI+3                                                          190
      NF=NI+2                                                          200
      GO TO 30                                                         210
   70 IF(NF-NR-1) 130,100,80                                           220
   80 WRITE(NT,90)(NBT(N),JNT(N),N=NI,NR),MAT,MF,MT,NS                 230
   90 FORMAT(2I11,44X,I4,I2,I3,I5)                                     240
      GO TO 120                                                        250
  100 WRITE(NT,110) (NBT(N),JNT(N),N=NI,NR),MAT,MF,MT,NS               260
  110 FORMAT(4I11,22X,I4,I2,I3,I5)                                     270
  120 NS=NS+1                                                          280
  130 NI=1                                                             290
      NF=3                                                             300
  140 IF(NF-NP) 150,150,190                                            310
  150 DO 160 K=1,6,2                                                   320
      N=NI+K/2                                                         330
      CALL CXFP(X(N),F(K),S(K),J(K))                                   340
  160 CALL CXFP(Y(N),F(K+1),S(K+1),J(K+1))                             350
      WRITE(NT,170)(F(K),S(K),J(K),K=1,6),MAT,MF,MT,NS                 360
  170 FORMAT(6(F8.5,A1,I2),I4,I2,I3,I5)                                370
      NS=NS+1                                                          380
      IF(NF-NP) 180,260,260                                            390
  180 NI=NI+3                                                          400
      NF=NI+2                                                          410
      GO TO 140                                                        420
  190 IF(NF-NP-1) 260,220,200                                          430
  200 CALL CXFP(X(NI),F(1),S(1),J(1))                                  440
      CALL CXFP(Y(NI),F(2),S(2),J(2))                                  450
      WRITE(NT,210)(F(K),S(K),J(K),K=1,2),MAT,MF,MT,NS                 460
  210 FORMAT(2(F8.5,A1,I2),44X,I4,I2,I3,I5)                            470
      GO TO 250                                                        480
  220 DO 230 K=1,4,2                                                   490
      N=NI+K/2                                                         500
      CALL CXFP(X(N),F(K),S(K),J(K))                                   510
  230 CALL CXFP(Y(N),F(K+1),S(K+1),J(K+1))                             520
      WRITE(NT,240)(F(K),S(K),J(K),K=1,4),MAT,MF,MT,NS                 530
  240 FORMAT(4(F8.5,A1,I2),22X,I4,I2,I3,I5)                            540
  250 NS=NS+1                                                          550
  260 CONTINUE                                                         560
      RETURN                                                           570
      END                                                              580
```

```
      SUBROUTINE PRTAB2(NT,NBT,JNT)                                     10
C=====PRINT TAB2 TYPE RECORD===========================================  20
      COMMON/RECS/MAT,MF,MT,C1,C2,L1,L2,N1,N2,NS                        30
      DIMENSION NBT(1),JNT(1)                                           40
      NR=N1                                                             50
      NP=N2                                                             60
   10 WRITE(NT,20)C1,C2,L1,L2,NR,NP,MAT,MF,MT,NS                        70
   20 FORMAT('0TAB2',9X,'C1',11X,'C2',11X,'L1',11X,'L2',11X,'NR',11X,   80
     1'NP',9X,'MAT  MF   MT    SEQ'/E20.4,E13.4,I9,3I13,I12,I4,I5,I7)   90
      NS=NS+1                                                          100
      WRITE(NT,30)                                                     110
   30 FORMAT('  (N)',8X,'NBT(N)',7X,'INT(N)',7X,'NBT(N+1)',            120
```

```
     15X,'INT(N+1)',5X,'NBT(N+2)',5X,'INT(N+2)')                        130
       NI=1                                                             140
       NE=3                                                             150
    40 IF(NE-NR)50,50,80                                                160
    50 WRITE(NT,60)NI,(NBT(N),JNT(N),N=NI,NE),MAT,MF,MT,NS              170
    60 FORMAT(' (',I4,')',I10,5I13,I11,I4,I5,I7)                        180
       NS=NS+1                                                          190
       IF(NE-NR)70,140,140                                              200
    70 NI=NI+3                                                          210
       NE=NI+2                                                          220
       GOTO40                                                           230
    80 IF(NE-NR-1)140,110,90                                            240
    90 WRITE(NT,100)NI,(NBT(N),JNT(N),N=NI,NR),MAT,MF,MT,NS             250
   100 FORMAT(' (',I4,')',I10,I13,52X,I11,I4,I5,I7)                     260
       GOTO130                                                          270
   110 WRITE(NT,120)NI,(NBT(N),JNT(N),N=NI,NR),MAT,MF,MT,NS             280
   120 FORMAT(' (',I4,')',I10,3I13,26X,I11,I4,I5,I7)                    290
   130 NS=NS+1                                                          300
   140 RETURN                                                           310
       END                                                              320
```

```
       SUBROUTINE PUTAB2(NT,NBT,JNT)                                     10
C=====PUNCH TAB2 TYPE RECORD ============================================  20
       COMMON/RECS/MAT,MF,MT,C1,C2,L1,L2,N1,N2,NS                        30
       DIMENSION F(6),S(6),J(6),NBT(1),JNT(1)                            40
       NR=N1                                                             50
       NP=N2                                                             60
    10 CALL CXFP(C1,F(1),S(1),J(1))                                      70
       CALL CXFP(C2,F(2),S(2),J(2))                                      80
       WRITE(NT,20)F(1),S(1),J(1),F(2),S(2),J(2),L1,L2,NR,NP,MAT,MF,MT,NS  90
    20 FORMAT(F8.5,A1,I2,F8.5,A1,I2,4I11,I4,I2,I3,I5)                   100
       NS=NS+1                                                          110
       NI=1                                                             120
       NE=3                                                             130
    30 IF(NE-NR) 40,40,70                                               140
    40 WRITE(NT,50)(NBT(N),JNT(N),N=NI,NE),MAT,MF,MT,NS                 150
    50 FORMAT(6I11,I4,I2,I3,I5)                                         160
       NS=NS+1                                                          170
       IF(NE-NR) 60,130,130                                             180
    60 NI=NI+3                                                          190
       NE=NI+2                                                          200
       GO TO 30                                                         210
    70 IF(NE-NR-1) 130,100,80                                           220
    80 WRITE(NT,90)(NBT(N),JNT(N),N=NI,NR),MAT,MF,MT,NS                 230
    90 FORMAT(2I11,44X,I4,I2,I3,I5)                                     240
       GO TO 120                                                        250
   100 WRITE(NT,110) (NBT(N),JNT(N),N=NI,NR),MAT,MF,MT,NS               260
   110 FORMAT(4I11,22X,I4,I2,I3,I5)                                     270
   120 NS=NS+1                                                          280
   130 RETURN                                                           290
       END                                                              300
```

```
       SUBROUTINE CXFP(X,F,S,N)                                          10
C=====CONVERT X FOR PUNCHING ============================================  20
C      X - FLOATING POINT NUMBER = F*10.0**N                             30
C      F - 0.999995 LE F LT 9.999995                                     40
C      S - SIGN (HOLLERITH + OR -) OF EXPONENT                           50
C      N - EXPONENT                                                      60
       DATA SP/1H+/,SM/1H-/                                              70
       IF(X.NE.0.0) GO TO 10                                             80
       F=0.0                                                             90
       S=SP                                                             100
       N=0                                                              110
       RETURN                                                           120
    10 N=ALOG10(ABS(X))                                                 130
       IF(ABS(X)-1.0) 40,20,20                                          140
    20 F=X/10.0**N                                                      150
       S=SP                                                             160
       IF(ABS(F)-9.999995) 70,30,30                                     170
    30 F=F/10.0                                                         180
       N=N+1                                                            190
       GO TO 70                                                         200
    40 N=1-N                                                            210
       F=X*10.0**N                                                      220
       S=SM                                                             230
       IF(ABS(F)-9.999995) 70,50,50                                     240
    50 F=F/10.0                                                         250
       N=N-1                                                            260
       IF(N) 60,60,70                                                   270
    60 S=SP                                                             280
    70 CONTINUE                                                         290
       RETURN                                                           300
       END                                                              310
```

```
       SUBROUTINE XTEXT(I,J)                                             10
       J=I                                                               20
       RETURN                                                            30
       END                                                               40
```

```
       SUBROUTINE EPT(EMIN,EMAX,NPR,NTP,NR,ER,GTR,ENS,N2X)               10
       COMMON/RECS/NR1(9),NS                                             20
       DIMENSION XP(200),XPT(100),ER(1),GTR(1),ENS(1)                    30
C      CALCULATES ENERGY MESH BASED ON NPR POINTS PER RESONANCE          40
C      NTP IS CALCULATED TOTAL POINTS TO DESCRIBE NR RESONANCES FROM     50
C       EMIN TO EMAX - ENS IS CALCULATED ENERGY MATRIX                   60
C      ER(J) IS RESONANCE ENERGY FOR JTH RESONANCE WHICH HAS TOTAL WIDTH  70
C       EQUAL GTR(J) AND XI(J) = GTR(J)/DELTA(J) WHERE DELTA(J) IS THE   80
C       DOPPLER WIDTH                                                    90
C                                                                       100
C      METHOD BASED ON UNICORN CODE (REF. NAA-SR-11980,VOL.VI)          110
C      OPTIMIZED POINTWISE REPRESENTATION OF (1+X**2)**-1 USING LOG     120
```

```
C       INTERPOLATION. THE CRITERION USED IS THAT BETWEEN ANY TWO POINTS
C       THE DIFFERENCE BETWEEN THE EXACT INTEGRAL AND THE DIFFERENCE
C       ASSUMING LOG Y,LINEAR X INTERPOLATION BE WITHIN 1 PERCENT OF A
C       CONSTANT, Q =(PI/2)(3/NPR)**3
C      IN THIS ROUTINE, X= 2(E-ER)/WIDTH WHERE WIDTH IS THE MAXIMUM
C      OF GTR OR DELTA
C      SET UP ENERGY POINTS XP IN VARIABLE X TO DESCRIBE A RESONANCE
       PNPR=NPR
       CONS=42.4115/PNPR**3
       XPT(1)=0.0
       XPT(2)=1.817121*CONS**(1.0/3.0)
       XT=0.0
       DO 25 N=1,98
       CPX=1.0+XPT(N)**2
    10 Q=CPX/(1.0+XPT(N+1)**2)
       CLN=ALOG(Q)
       XIN=XPT(N+1)-XPT(N)
       ULO=XIN*(Q-1.0)/CLN/CPX
       VXT=ATAN(XPT(N+1))
       UXT=VXT-XT
       CRI=ABS(ULO-UXT)-CONS
       IF(ABS(CRI).LE.0.01*CONS) GO TO 20
    15 DELX=              -CRI/(ULO*ABS((2.0*XPT(N+1)*Q/CPX+CLN/XIN)*(Q/
      1(Q-1.0)-1.0/CLN))+ABS(CRI)/XPT(N+1))
C      CHECK FOR OSCILLATION ABOUT SOLUTION
       DEL=DELX+DELX1
       IF(ABS(DEL)/XPT(N+1)-5.0E-6) 17,17,16
    16 IF((XPT(N+1)+DELX)-XPT(N)) 17,17,19
    17 DELX=DELX/2.0
       GO TO 16
    19 XPT(N+1)=XPT(N+1)+DELX
       DELX1=DELX
       NS=N
       IF(XPT(N+1)-10000.0) 10,10,30
    20 XT=VXT
       IF(N-1) 22,22,24
    22 XPT(N+2)=XPT(N+1)*2.0
       GO TO 25
    24 XPT(N+2)=XPT(N+1)**2/XPT(N)
    25 CONTINUE
    30 NPT=2*NS-1
       DO 35 I=1,NS
       J=NS-I+1
       XP(I)=-XPT(J)
       K=NPT-I+1
    35 XP(K)=XPT(J)
       NPTP1=NPT+1
       DO 40 I=NPTP1,200
    40 XP(I)=0.0
C      FIND RESONANCE NO. MM NEAREST TO EMIN TO START ENERGY CALCULATION
       ENS(1)=EMIN
       NRM=MAXO(NR-1,1)
       DO 45 MM=1,NRM
       IF(2.0*ENS(1).LT. (ER(MM)+ER(MM+1))) GO TO 46
    45 CONTINUE
       MM=NRM
C      FIND POINT N OF XP ARRAY FOR THE MMTH RESONANCE WHICH LIES JUST
C       BELOW EMIN IN ABSOLUTE ENERGY
    46 WIDTH=GTR(MM)
       X1=2.*(ENS(1)-ER(MM))/WIDTH
       DO 47 K=1,NPT
       N=MAXO(K-1,1)
       IF(X1-XP(K)) 48,47,47
    47 CONTINUE
C      CALCULATE ENERGY POINTS
    48 IF(MM.LT.NR)WIDTH1=GTR(MM+1)
       I=1
       GOTO 52
    50 IF(ENS(I).LE.ENS(1))GOTO 51
    52 I=I+1
    51 IF(ENS(I-1).GE.EMAX) GO TO 96
       N=N+1
       IF(I.GT.N2X) CALL RESERR(105,MM,NR,2)
       ENS(I)=XP(N)*WIDTH/2.0+ER(MM)
       IF(MM-NR) 93,87,96
    87 IF(N-NPT) 50,50,96
    93 XMD=2.0*(ER(MM+1)-ER(MM))/(WIDTH+WIDTH1)
       IF(N.GT.NPT) GO TO 89
       IF(XP(N).LT.XMD) GO TO 50
    89 MM=MM+1
       WIDTH=WIDTH1
       IF(MM.LT.NR)WIDTH1=GTR(MM+1)
       DO 91 K=1,NPT
       N=MAXO(K-1,1)
       IF(-XMD-XP(K)) 92,91,91
    91 CONTINUE
    92 IF(I.GT.N2X) CALL RESERR(105,MM,NR,2)
       ENS(I)=-XMD*WIDTH/2.0+ER(MM)
       GO TO 50
    96 NTP=I-1
       IF(NTP.GT.N2X) CALL RESERR(105,MM,NR,2)
       ENS(NTP)=EMAX
       RETURN
       END
```

```
       SUBROUTINE SCALE
C
C          SCALE GENERATES COMPLETE SCALE.IF IT IS SHORTER THAN NCMAX IT
C          IS KEPT IN MAIN STORAGE,OTHERWISE DATASETS NIN,NOUT ARE USED.
C
       DIMENSION E(1),TEMP(1),EC(1)
       DATA NIN,NOUT/3,4/
       ENTRY INITSC(NCMAX)
       NC=0
       NC=0
       NDS=0
       RETURN
```

```
C
      ENTRY GENER(MAX,E,TEMP,EC,ND,NCU)
      IF(NC.NE.0) GOTO 10
C
C          FIRST ENTRY TO GENER.
      IF(MAX.LE.NCMAX) GOTO 4
      REWIND NOUT
      DO 2 M=1,MAX,NCMAX
      ME=MIN0(M+NCMAX-1,MAX)
      LDAT=ME-M+1
    2 WRITE(NOUT)  LDAT,(E(I),I=M,ME)
      REWIND NOUT
      NDS=NOUT
      NOUT=NIN
      NIN=NDS
      GOTO 6
    4 DO 5 M=1,MAX
    5 EC(M)=E(M)
    6 NC=MAX
      ND=NDS
      NCU=NC
      RETURN
C
C          FURTHER ENTRIES.
   10 ICDONE=0
      NEWNC=0
      KC=0
      IF(NC.LE.NCMAX) GOTO 19
      REWIND NIN
      REWIND NOUT
C
C          ENERGIES ) E(1) ARE SIMPLY COPIED.
   12 READ(NIN) LDAT,(EC(I),I=1,LDAT)
      IF(E(1).LT.EC(LDAT)*1.000001) GOTO 20
      IF(LDAT.LT.NCMAX) GOTO 20
      WRITE(NOUT) LDAT,(EC(I),I=1,LDAT)
      ICDONE=ICDONE+LDAT
      NEWNC=ICDONE
      IF(ICDONE.GE.NC) GOTO 60
      GOTO 12
C
   19 LDAT=NC
   20 IC=1
      M=1
C
C          INSERT NEW ENERGIES IF NECESSARY.
   26 IF(EC(IC).GT.E(M)*1.000010) GOTO 40
      IF(EC(IC).LT.E(M)*0.99999 ) GOTO 30
C
C          EC=E
      M=M+1
      IF(M.GT.MAX) GOTO 50
C
C          EC<E
   30 KC=KC+1
      IF(KC.LE.NCMAX) GOTO 36
      WRITE(NOUT) NCMAX,(TEMP(I),I=1,NCMAX)
      KC=1
      NEWNC=NEWNC+NCMAX
   36 TEMP(KC)=EC(IC)
      IC=IC+1
      IF(IC.LE.LDAT) GOTO 26
      ICDONE=ICDONE+LDAT
      IF(ICDONE.GE.NC) GOTO 60
      READ(NIN) LDAT,(EC(I),I=1,LDAT)
      IC=1
      GOTO 26
C
C          EC>E
C     CHECK IF E(M) IS NEAR BY TEMP(KC).
   40 IF(KC.GT.0.AND.E(M).LE.TEMP(KC)*1.000010) GOTO 43
      KC=KC+1
      IF(KC.LE.NCMAX) GOTO 42
      WRITE(NOUT) NCMAX,(TEMP(I),I=1,NCMAX)
      KC=1
      NEWNC=NEWNC+NCMAX
   42 TEMP(KC)=E(M)
   43 M=M+1
      IF(M.LE.MAX) GOTO 26
C
C          END OF DATA FOR E.
   50 DO 52 J=IC,LDAT
      KC=KC+1
      IF(KC.LE.NCMAX) GOTO 52
      WRITE(NOUT) NCMAX,(TEMP(I),I=1,NCMAX)
      KC=1
      NEWNC=NEWNC+NCMAX
   52 TEMP(KC)=EC(J)
      ICDONE=ICDONE+LDAT
      IF(ICDONE.GE.NC) GOTO 70
      READ(NIN) LDAT,(EC(I),I=1,LDAT)
      IC=1
      GOTO 50
C
C          END OF DATA FOR EC.
   60 DO 62 J=M,MAX
      KC=KC+1
      IF(KC.LE.NCMAX) GOTO 62
      WRITE(NOUT) NCMAX,(TEMP(I),I=1,NCMAX)
      KC=1
      NEWNC=NEWNC+NCMAX
   62 TEMP(KC)=E(J)
C
C          WRITE LAST BLOCK AND UPDATE COUNTERS.
   70 NC=NEWNC+KC
      IF(NC.LE.NCMAX) GOTO 72
      WRITE(NOUT) KC,(TEMP(I),I=1,KC)
      NDS=NOUT
      NOUT=NIN
      NIN=NDS
```

```
      REWIND NIN                                                        1230
      REWIND NOUT                                                       1240
      GOTO 100                                                          1250
   72 DO 73 I=1,KC                                                      1260
   73 EC(I)=TEMP(I)                                                     1270
  100 ND=NDS                                                            1280
      NCU=NC                                                            1290
      RETURN                                                            1300
      END                                                               1310
```

```
      SUBROUTINE COMBP(X1P,YA1P,YB1P,X2P,YA2P,YB2P,IAP,IBP,OPER,CON,EPSP   10
     1,LCF,NBT,JNT,X,Y)                                                 20
C=====COMBINE TAB1 FUNCTIONS A AND B IN A SINGLE PANEL==================30
C     (DELETE THE TERMINAL P ON ITEMS IN LIST ABOVE. I.E.,READ X1P AS X1  40
C     1)                                                                50
C     X1 AND X2 ARE THE X LIMITS OF THE PANEL.                          60
C     YA1 AND YA2 ARE THE VALUES OF FUNCTION A AT X1 AND X2.            70
C     YB1 AND YB2 ARE THE VALUES OF FUNCTION B AT X1 AND X2.            80
C     IA AND IB ARE THE INTERPOLATION CODES FOR FUNCTIONS A AND B.      90
C     OPER IS THE NAME OF THE OPERATION TO BE PERFORMED.               100
C          YC=OPER(YA,YB,CON)                                          110
C     CON(N) IS AN ARRAY OF CONSTANTS REQUIRED BY FUNCTION OPER.       120
C     EPS IS THE RELATIVE ERROR CRITERION.                             130
C     THE ANSWER FUNCTION IS ADDED TO THE ANSWER FUNCTION FROM         140
C          PREVIOUS PANELS ALREADY STORED IN COMMON/RECS/ AND          150
C          ARRAYS NBT,JNT,X, AND Y.                                    160
C     LCF=0, NORMAL RETURN.                                            170
C        =1, ANSWER OVERFLOWS /RECS/.                                  180
C-----NO ERROR STOPS                                                   190
      COMMON/RECS/MAT,MF,MT,C1,C2,L1,L2,N1,N2                          200
      DIMENSION NBT(1),JNT(1),X(1),Y(1)                                210
      COMMON/INOUT/NR1(124),ICOMB,NBX,N1X,N2X,NR2(22),FAICOD           220
C=====ICOMB=0  THE INTERPOLATION SCHEME IS CHOOSEN BY THIS ROUTINE      230
C          =1  THE INTERPOLATION SCHEME IS ALWAYS LINEAR-LINEAR         240
      EXTERNAL OPER                                                    250
C=====INITIALIZE                                                       260
  100 LCF=0                                                            270
      X1=X1P                                                           280
      YA1=YA1P                                                         290
      YB1=YB1P                                                         300
      X2=X2P                                                           310
      YA2=YA2P                                                         320
      YB2=YB2P                                                         330
      IA=IAP                                                           340
      IB=IBP                                                           350
      EPS=EPSP                                                         360
      IEX=0                                                            370
      EPPS=1.E-6                                                       380
      IF(IA.EQ.1)YA2=YA1                                               390
      YA2E=YA2                                                         400
      IF(IB.EQ.1)YB2=YB1                                               410
      YB2E=YB2                                                         420
C-----CALCULATE VALUES AT END POINTS                                   430
      YC1=OPER(YA1,YB1,CON)                                            440
  110 YC2=OPER(YA2,YB2,CON)                                            450
C-----TEST FOR DISCONTINUITY                                           460
      IF(X1.NE.X2)GOTO201                                              470
      IC=2                                                             480
      IEX=1                                                            490
      GOTO500                                                          500
C-----TEST FOR EQUALITY OF X1 AND X2 UP TO 6 SIGNIFICANT DIGITS        510
  201 IF(ABS((X1-X2)/X1).GT.EPPS )GOTO 200                             520
      IEX=1                                                            530
      IC=2                                                             540
      GOTO 500                                                         550
C=====CALCULATE INTERMEDIATE VALUE, (X3,YC3)                           560
  200 X3=0.5*(X1+X2)                                                   570
      IF(ABS((X3-X1)/X2).GT.EPPS )GOTO 202                             580
      GOTO 900                                                         590
  202 CALL TERP1(X1,YA1,X2,YA2,X3,YA3,IA)                              600
      CALL TERP1(X1,YB1,X2,YB2,X3,YB3,IB)                              610
      YC3=OPER(YA3,YB3,CON)                                            620
C=====TEST TO SEE IF YC3 CAN BE OBTAINED BY INTERPOLATION              630
      IF(YC1.LE.0.0)GOTO310                                            640
      IF(YC2.LE.0.0)GOTO310                                            650
      IF(X1.LE.0.0)GOTO320                                             660
      IF(X2.LE.0.0)GOTO320                                             670
      GOTO340                                                          680
C-----LINEAR INTERPOLATION ONLY                                        690
  310 YC4=0.5*(YC1+YC2)                                                700
      IF(YC3.EQ.YC4) GO TO 335                                         710
      EPS4=ABS(YC4-YC3)*3.0/(ABS(YC1)+ABS(YC2)+ABS(YC3))               720
      GOTO330                                                          730
  320 YC4=0.5*(YC1+YC2)                                                740
      EPS4=ABS(YC4-YC3)/YC3                                            750
  330 IF(EPS4.GT.EPS)GOTO400                                           760
  335 IC=2                                                             770
      GOTO500                                                          780
C-----LAST INTERPOLATION CODE EPS4 AND BEST INTERPOLATION CODE EPS5     790
C-----IS PRODUCED. IF EPS4.LE.FAICOD*EPS5 THE LAST INTERPOLATION CODE  800
C-----IS USED. IN THE OTHER CASE THE BEST INTERPOLATION CODE IS USED.   810
  340 EPS5=1.0E+10                                                     820
      EPS4=1.0E+10                                                     830
      NFIN=5                                                           840
      IF(ICOMB.EQ.1) NFIN=2                                            850
      IF(N1.GE.1) IC4=JNT(N1)                                          860
      DO 360 N=2,NFIN                                                  870
      CALL TERP1(X1,YC1,X2,YC2,X3,YC4,N)                               880
      EPS6=ABS(YC4-YC3)/YC3                                            890
      IF(N1.GE.1.AND.IC4.EQ.N) EPS4=EPS6                               900
      IF(EPS6.GE.EPS5)GOTO360                                          910
      EPS5=EPS6                                                        920
      IC5=N                                                            930
  360 CONTINUE                                                         940
      IF(EPS5.GT.EPS) GOTO 400                                         950
      IF(N1.LT.1.OR.EPS4.GT.EPS) GOTO 341                              960
      IF(EPS4.LE.FAICOD*EPS5) GOTO 342                                 970
```

```
  341 IC=IC5                                                          980
      GOTO 500                                                        990
  342 IC=IC4                                                         1000
      GOTO 500                                                       1010
C=====REJECT VALUE, HALVE INTERVAL, AND TRY AGAIN                    1020
  400 X2=X3                                                          1030
      YA2=YA3                                                        1040
      YB2=YB3                                                        1050
      YC2=YC3                                                        1060
      GOTO200                                                        1070
C=====OK, ACCEPT VALUE YC2                                           1080
  500 IF(N2.GT.0)GOTO510                                             1090
      X(1)=X1                                                        1100
      Y(1)=YC1                                                       1110
      X(2)=X2                                                        1120
      Y(2)=YC2                                                       1130
      N2=2                                                           1140
      NBT(1)=2                                                       1150
      JNT(1)=IC                                                      1160
      N1=1                                                           1170
      GOTO700                                                        1180
  510 N2=N2+1                                                        1190
      IF(N2.GT.N2X)GOTO800                                           1200
      X(N2)=X2                                                       1210
      Y(N2)=YC2                                                      1220
      IF(IC.EQ.JNT(N1)) GO TO 520                                    1230
      N1=N1+1                                                        1240
      IF(N1.GT.N1X)GOTO800                                           1250
      JNT(N1)=IC                                                     1260
      NBT(N1)=N2                                                     1270
      GOTO700                                                        1280
  520 NBT(N1)=N2                                                     1290
      IF(IEX.EQ.1)GOTO 900                                           1300
C=====TEST TO SEE IF PREVIOUS POINT CAN BE ELIMINATED                1310
  600 IF(X1.NE.X2)GOTO610                                            1320
      IF(N1.LE.1)GOTO900                                             1330
      IF(NBT(N1-1).NE.N2-1)GOTO900                                   1340
      IF(JNT(N1-1).NE.1)GOTO900                                      1350
      X(N2-1)=X(N2)                                                  1360
      Y(N2-1)=Y(N2)                                                  1370
      N2=N2-1                                                        1380
      N1=N1-1                                                        1390
      NBT(N1)=N2                                                     1400
      GOTO900                                                        1410
  610 CONTINUE                                                       1420
C     FOLLOWING STATEMENTS DELETED JULY 1974 BY E.STEIN.             1430
C     THE STATEMENTS WILL BE REPLACED BY A CROP-CALL IN COMB.        1440
C 610 CALL TERP1(X(N2-2),Y(N2-2),X(N2),Y(N2),X(N2-1),YC6,IC)         1450
C     EPS6=ABS(YC6-Y(N2-1))                                          1460
C     IF(EPS6.EQ.0.0)GOTO620                                         1470
C     IF(Y(N2-1).EQ.0.0)GOTO700                                      1480
C     EPS6=EPS6/ABS(Y(N2-1))                                         1490
C     IF(EPS6.GT.EPS)GOTO700                                         1500
C 620 X(N2-1)=X(N2)                                                  1510
C     Y(N2-1)=Y(N2)                                                  1520
C     N2=N2-1                                                        1530
C     NBT(N1)=N2                                                     1540
C=====SEE IF MORE POINTS ARE NEEDED                                  1550
  700 IF(IEX.EQ.1)GOTO 900                                           1560
      IF(X2.LT.X2P)GOTO720                                           1570
      IF(YA2E.NE.YA2P)GOTO710                                        1580
      IF(YB2E.EQ.YB2P)GOTO900                                        1590
  710 YA2E=YA2P                                                      1600
      YB2E=YB2P                                                      1610
  720 X1=X2                                                          1620
      YA1=YA2                                                        1630
      YB1=YB2                                                        1640
      YC1=YC2                                                        1650
      X2=X2P                                                         1660
      YA2=YA2E                                                       1670
      YB2=YB2E                                                       1680
      GOTO110                                                        1690
C=====OVERFLOW                                                       1700
  800 LOF=1                                                          1710
C=====FINISHED                                                       1720
  900 RETURN                                                         1730
      END                                                            1740
```

```
      SUBROUTINE DENISE(MATP,LRP,LISRES,LISUNR,NIS,LFW,LNU,BK,NBT,JNT,    10
     *X,Y,B,A)                                                            20
      COMMON/DENS/NR3(402),MX                                             30
      COMMON/INOUT/NO,NR1(5),LEDITS,NR2(122),NBKX,NR4(10),EPSR            40
      DIMENSION BK(1),NBT(1),JNT(1),X(1),Y(1),B(1),A(1)                   50
      EPS=EPSR                                                            60
      NCMAX=NBKX/2                                                        70
      CALL IICOMB(NBT,JNT,X,Y,A,A)                                        80
      CALL IFETCH(NBT,JNT,X,Y,B,A,A)                                      90
      CALL ISTORE(NBT,JNT,X,Y,B,A,A)                                     100
      CALL IFPDS(A,A)                                                    110
      CALL IIPDS(A,A)                                                    120
      CALL IRRECS(NBT,JNT,X,Y,B)                                         130
      MX=100                                                             140
      CALL DELETE(0,A)                                                   150
      IF(LISRES.EQ.1.AND.LISUNR.EQ.1)GOTO 1                              160
      IF(LISRES.EQ.1)GOTO 20                                             170
      CALL MYRIAM(NIS,LISRES,LISUNR,4HSCAT,EL,EH,EPS,C,Y)                180
      CALL WRECS2(0,4HSGN ,4HRES ,EL,EH)                                 190
   20 IF(LISUNR.EQ.1)GOTO 21                                             200
      CALL MYRIAM(NIS,LISRES,LISUNR,4HSCAT,EL,EH,EPS,1,Y)                210
      CALL WRECS2(0,4HSGN ,4HSTAT,EL,EH)                                 220
   21 IF(LISRES.EQ.1)GOTO 22                                             230
      CALL MYRIAM(NIS,LISRES,LISUNR,4HCAPT,EL,EH,EPS,C,Y)                240
      CALL WRECS2(0,4HSGG ,4HRES ,EL,EH)                                 250
   22 IF(LISUNR.EQ.1)GOTO 23                                             260
      CALL MYRIAM(NIS,LISRES,LISUNR,4HCAPT,EL,EH,EPS,1,Y)                270
      CALL WRECS2(0,4HSGG ,4HSTAT,EL,EH)                                 280
   23 IF(LFW.NE.1)GOTO 2                                                 290
```

```
      IF(LISRES.EQ.1)GOTO 24                                            300
      CALL MYRIAM(NIS,LISRES,LISUNR,4HFISS,EL,EH,EPS,C,Y)               310
      CALL WRECS2(0,4HSGF  ,4HRES  ,EL,EH)                              320
   24 IF(LISUNR.EQ.1)GOTO 2                                             330
      CALL MYRIAM(NIS,LISPES,LISUNR,4HFISS,EL,EH,EPS,1,Y)               340
      CALL WRECS2(0,4HSGF  ,4HSTAT,EL,EH)                               350
    2 CALL XTEXT(4HRES  ,NAME2)                                         360
      IF(LISRES.EQ.1)GOTO 10                                            370
      CALL MARION(LFW,EL,EH,EPS,NAME2)                                  380
      IF(LEDITS.NE.1)GOTO 947                                           390
      WRITE(NO,3)                                                       400
    3 FORMAT(1H1,'POINT WISE TOTAL CROSS SECTION CALCULATED FROM',      410
     * ' RESOLVED RESONANCE PARAMETERS'//)                             420
      CALL WREC(3,NC,4)                                                 430
  947 CONTINUE                                                          440
      CALL WRECS2(0,4HSGT  ,4HRES  ,EL,EH)                              450
   10 IF(LISUNR.EQ.1)GOTO 1                                             460
      CALL XTEXT(4HSTAT,NAME2)                                          470
      EPS=EPSR                                                          480
      CALL MARION(LFW,EL,EH,EPS,NAME2)                                  490
      IF(LEDITS.NE.1)GOTO 948                                           500
      WRITE(NO,4)                                                       510
    4 FORMAT(1H1,'POINT WISE TOTAL CROSS SECTION CALCULATED FROM ',     520
     *   'STATISTICAL RESONANCE PARAMETERS'//)                          530
      CALL WREC(3,NO,4)                                                 540
  948 CONTINUE                                                          550
      CALL WRECS2(0,4HSGT  ,4HSTAT,EL,EH)                               560
C-----FILE 3                                                            570
    1 CALL SOPHIE(MATP,LISRES,LISUNR,LFW,LNU,EPSR,BK,BK,BK(1+NCMAX),    580
     *NCMAX,NBT,JNT,X,Y,B,B)                                            590
      RETURN                                                            600
      END                                                               610
```

```
      SUBROUTINE FETCH(MA,LNT)                                           10
C=====FETCH RECORD FROM DENSE STORAGE================================    20
C     MA - RECORD IDENT TO BE FETCHED                                    30
C     LNT=0, RECORD FOUND AND MOVED TO COMMON/RECS/ AND                  40
C            ARRAYS NBT,JNT,X, AND Y.                                    50
C        =1, RECORD NOT IN DENSE STORAGE                                 60
C-----NO ERROR STOPS                                                    70
      COMMON/RECS/MAT,MF,MT,C1,C2,L1,L2,N1,N2                            80
      COMMON/DENS/JMT(100),JAT(100),JTT(100),JLT(100),                   90
     1    JNS,MNS,MX                                                    100
      COMMON/INOUT/ NR1(125),NBX,N1X,N2X                                110
      DIMENSION LA(1),NBT(1),JNT(1),X(1),Y(1),B(1),A(1)                 120
C-----SEARCH FOR RECORD                                                 130
  100 LNT=0                                                             140
      IF(MNS.LE.1)GOTO120                                               150
      MNSP=MNS-1                                                        160
      DO110M=1,MNSP                                                     170
      MP=M                                                              180
      IF(JMT(M).EQ.MA)GOTO130                                           190
  110 CONTINUE                                                          200
  120 LNT=1                                                             210
      GOTO300                                                           220
  130 JA=JAT(MP)                                                        230
      JT=JTT(MP)                                                        240
      IF(JT.EQ.6)GOTO210                                                250
C-----MOVE FIRST 9 WORDS                                                260
      MAT=LA(JA)                                                        270
      MF=LA(JA+1)                                                       280
      MT=LA(JA+2)                                                       290
      C1=A(JA+3)                                                        300
      C2=A(JA+4)                                                        310
      L1=LA(JA+5)                                                       320
      L2=LA(JA+6)                                                       330
      N1=LA(JA+7)                                                       340
      N2=LA(JA+8)                                                       350
      JA=JA+9                                                           360
C-----TEST IF RECORD IS TOO LONG.                                       370
      IF((JT.EQ.2.OR.JT.EQ.5).AND.N1.GT.NBX) CALL ERROR(104)            380
      IF((JT.EQ.3.OR.JT.EQ.4).AND.N1.GT.N1X) CALL ERROR(103)            390
      IF(JT.EQ.3.AND.N2.GT.N2X) CALL ERROR(105)                         400
C-----MOVE REMAINDER OF RECORD                                          410
      GOTO(300,140,160,190,140),JT                                      420
  140 DO150N=1,N1                                                       430
      J=JA+N-1                                                          440
  150 B(N)=A(J)                                                         450
      GOTO300                                                           460
  160 DO170N=1,N1                                                       470
      J=JA+2*(N-1)                                                      480
      NBT(N)=LA(J)                                                      490
  170 JNT(N)=LA(J+1)                                                    500
      JA=JA+2*N1                                                        510
      DO180N=1,N2                                                       520
      J=JA+2*(N-1)                                                      530
      X(N)=A(J)                                                         540
  180 Y(N)=A(J+1)                                                       550
      GOTO300                                                           560
  190 DO200N=1,N1                                                       570
      J=JA+2*(N-1)                                                      580
      NBT(N)=LA(J)                                                      590
  200 JNT(N)=LA(J+1)                                                    600
      GOTO300                                                           610
C-----TRID RECORD                                                       620
  210 MAT=LA(JA)                                                        630
      MF=LA(JA+1)                                                       640
      MT=LA(JA+2)                                                       650
      IF(NBX.LT.17) CALL ERROR(104)                                     660
      DO220N=1,17                                                       670
      J=JA+2+N                                                          680
  220 B(N)=A(J)                                                         690
C-----FINISHED                                                          700
  300 RETURN                                                            710
      ENTRY IFETCH(NBT,JNT,X,Y,B,A,LA)                                  720
      GOTO 300                                                          730
      END                                                               740
```

```
      SUBROUTINE STORE(JT,MA,LOF)                                       10
C=====STORE RECORD IN DENSE STORAGE=================================    20
C     JT = RECORD TYPE, 1-CONT, 2-LIST, 3-TAB1, 4-TAB2, 5-HOL, 6-TPID   30
C     MA = IDENT OF RECORD TO BE STORED. IF A RECORD WITH                40
C          THE SAME IDENT IS ALREADY STORED, IT IS DELETED.              50
C     LOF= OVERFLOW INDICATOR NORMALLY ZERO. IF LOF=1, RECORD            60
C          WILL NOT FIT.                                                 70
C-----ERROR STOP 300, JT NOT IN RANGE 1-6                               80
C     ERROR STOP 301, MA=0 NOT ALLOWED                                  90
      COMMON/DENS/JMT(100),JAT(100),JTT(100),JLT(100),                  100
     1    JNS,MNS,MX                                                    110
      COMMON/INOUT/NR1(130),JX                                          120
      DIMENSION LA(1),NBT(1),JNT(1),X(1),Y(1),B(1),A(1)                 130
      COMMON/RECS/MAT,MF,MT,C1,C2,L1,L2,N1,N2                           140
C-----INITIALIZE ANT TEST JT AND MA                                     150
  100 LOF=0                                                             160
      IF(MA.EQ.0)CALL ERROR(301)                                        170
      IF((JT.LT.1).OR.(JT.GT.6))CALL ERROR(300)                         180
C-----TEST FOR RECORD WITH SAME IDENT                                   190
  110 IF(MNS.LE.1)GOTO130                                               200
      MNSP=MNS-1                                                        210
      DO120M=1,MNSP                                                     220
      IF(JMT(M).EQ.MA) CALL DELETE(MA,A)                                230
  120 CONTINUE                                                          240
      IF(MNS.GT.MX)GOTO320                                              250
C-----TEST FOR TOO MANY VALUES                                          260
  130 GOTO(140,150,160,170,150,180),JT                                  270
  140 JL=9                                                              280
      GOTO190                                                           290
  150 JL=9+N1                                                           300
      GOTO190                                                           310
  160 JL=9+2*(N1+N2)                                                    320
      GOTO190                                                           330
  170 JL=9+2*N1                                                         340
      GOTO190                                                           350
  180 JL=20                                                             360
  190 IF((JNS+JL-1).GT.JX)GOTO320                                       370
C-----MOVE FIRST 9 WORDS                                                380
  200 LA(JNS)=MAT                                                       390
      LA(JNS+1)=MF                                                      400
      LA(JNS+2)=MT                                                      410
      A(JNS+3)=C1                                                       420
      A(JNS+4)=C2                                                       430
      LA(JNS+5)=L1                                                      440
      LA(JNS+6)=L2                                                      450
      LA(JNS+7)=N1                                                      460
      LA(JNS+8)=N2                                                      470
C-----MOVE REST OF RECORD                                               480
  210 GOTO(310,220,240,270,220,290),JT                                  490
  220 DO230N=1,N1                                                       500
      J=JNS+8+N                                                         510
  230 A(J)=B(N)                                                         520
      GOTO310                                                           530
  240 DO250N=1,N1                                                       540
      J=JNS+7+2*N                                                       550
      LA(J)=NBT(N)                                                      560
  250 LA(J+1)=JNT(N)                                                    570
      JP=JNS+7+2*N1                                                     580
      DO260N=1,N2                                                       590
      J=JP+2*N                                                          600
      A(J)=X(N)                                                         610
  260 A(J+1)=Y(N)                                                       620
      GOTO310                                                           630
  270 DO280N=1,N1                                                       640
      J=JNS+7+2*N                                                       650
      LA(J)=NBT(N)                                                      660
  280 LA(J+1)=JNT(N)                                                    670
      GOTO310                                                           680
  290 DO300N=1,17                                                       690
      J=JNS+2+N                                                         700
  300 A(J)=B(N)                                                         710
C-----UPDATE TABLES AND COUNTERS                                        720
  310 JMT(MNS)=MA                                                       730
      JTT(MNS)=JT                                                       740
      JAT(MNS)=JNS                                                      750
      JLT(MNS)=JL                                                       760
      MNS=MNS+1                                                         770
      JNS=JNS+JL                                                        780
      GOTO400                                                           790
C-----OVERFLOW                                                          800
  320 LOF=1                                                             810
C-----FINISHED                                                          820
  400 RETURN                                                            830
      ENTRY ISTORE(NBT,JNT,X,Y,B,A,LA)                                  840
      GOTO 400                                                          850
      END                                                               860
```

```
      SUBROUTINE DELETE(MA,A)                                           10
C=====DELETE RECORD MA FROM DENSE STORAGE============================   20
C     IF MA=0, CLEAR DENSE STORAGE                                      30
C-----NO ERROR STOPS                                                    40
      COMMON/DENS/JMT(100),JAT(100),JTT(100),JLT(100),                  50
     1    JNS,MNS,MX                                                    60
      COMMON/INOUT/NR1(130),JX                                          70
      DIMENSION A(1)                                                    80
C=====TEST MA                                                           90
  100 IF(MA.NE.0)GOTO200                                                100
C=====CLEAR DENSE STORAGE                                               110
      MNS=1                                                             120
      JNS=1                                                             130
      DO110M=1,MX                                                       140
      JMT(M)=0                                                          150
      JAT(M)=0                                                          160
```

```
      JLT(M)=0                                                          170
  110 JTT(M)=0                                                          180
      DO120J=1,JX                                                       190
  120 A(J)=0.0                                                          200
      GOTO300                                                           210
C=====DELETE RECORD MA, SEARCH FOR LOCATION                            220
  200 IF(MNS.LE.1)GOTO300                                               230
      MNSP=MNS-1                                                        240
      DO210M=1,MNSP                                                     250
      MP=M                                                              260
      IF(JMT(M).EQ.MA)GOTO220                                           270
  210 CONTINUE                                                          280
      GOTO300                                                           290
  220 JA=JAT(MP)                                                        300
      JL=JLT(MP)                                                        310
C-----CLOSE UP ARRAY A(J)                                              320
      JP=JNS-JA-JL                                                      330
      IF(JP.LE.0)GOTO240                                                340
      DO230N=1,JP                                                       350
      J1=JA+N-1                                                         360
      J2=J1+JL                                                          370
  230 A(J1)=A(J2)                                                       380
  240 JNS=JNS-JL                                                        390
      DO250N=1,JL                                                       400
      J=JNS+N-1                                                         410
  250 A(J)=0.0                                                          420
C-----CLOSE UP J TABLES                                                430
      MXP=MNS-MP-1                                                      440
      IF(MXP.LE.0)GOTO270                                               450
      DO260M=1,MXP                                                      460
      M1=MP+M-1                                                         470
      JMT(M1)=JMT(M1+1)                                                 480
      JAT(M1)=JAT(M1+1)                                                 490
      JTT(M1)=JTT(M1+1)                                                 500
  260 JLT(M1)=JLT(M1+1)                                                 510
  270 MNS=MNS-1                                                         520
      JMT(MNS)=0                                                        530
      JAT(MNS)=0                                                        540
      JTT(MNS)=0                                                        550
      JLT(MNS)=0                                                        560
      MXP=MXP-1                                                         570
      IF(MXP.LE.0) GO TO 300                                            580
      DO 280 M=1,MXP                                                    590
      M1=MP+M                                                           600
  280 JAT(M1)=JAT(M1-1)+JLT(M1-1)                                       610
C=====FINISHED                                                         620
  300 RETURN                                                            630
      END
```

```
      SUBROUTINE FPDS(JA,NP,XP,YP,IP)                                    10
C=====FETCH POINT FROM /DENS/ STORAGE===================================  20
C     JA - STARTING INDEX IN /DENS/ ARRAY A OF THE TAB1 RECORD             30
C          CONTAINING ARRAYS X(N),Y(N),N=1,N2                              40
C     NP - DESIRED POINT NUMBER                                            50
C     XP - X VALUE CORRESPONDING TO NP                                     60
C     YP - Y VALUE CORRESPONDING TO NP                                     70
C     IP - INTERPOLATION CODE USED BETWEEN X(NP) AND X(NP+1)               80
C     THE FOLLOWING TABLE SUMMARIZES THE OUTPUT VALUES OF XP,YP,IP         90
C          NP              XP          YP       IP                       100
C     -----------------    --------    -----    --                       110
C     NP.LE.0             -1.0E+20     0.0      1                        120
C     NP.GT.0.AND.LT.N2     X(NP)      Y(NP)    I                        130
C     NP.EQ.N2              X(N2)      Y(N2)    2                        140
C     NP.EQ.N2+1            X(N2)      0.0      2                        150
C     NP.GT.N2+1           1.0E+20     0.0      2                        160
C-----NO ERROR STOPS                                                     170
      DIMENSION LA(1),A(1)                                               180
C-----INITIALIZE                                                         190
   10 N1=LA(JA+7)                                                        200
      N2=LA(JA+8)                                                        210
      J1=JA+9                                                            220
      J2=J1+2*N1                                                         230
C-----TEST NP                                                            240
   20 IF(NP.LE.0)GOTO30                                                  250
      IF(NP.LE.N2)GOTO40                                                 260
      IF(NP.EQ.N2+1)GOTO70                                               270
      GOTO80                                                             280
C-----NP.LE.0                                                            290
   30 XP=-1.0E+20                                                        300
      YP=0.0                                                             310
      IP=1                                                               320
      GOTO100                                                            330
C-----NP.GT.0 AND NP.LE.N2                                               340
   40 K=J2+2*(NP-1)                                                      350
      XP=A(K)                                                            360
      YP=A(K+1)                                                          370
      DO50M=1,N1                                                         380
      K=J1+2*(M-1)                                                       390
      IF(NP+1.LE.LA(K))GOTO60                                            400
   50 CONTINUE                                                           410
      IP=2                                                               420
      GOTO100                                                            430
   60 IP=LA(K+1)                                                         440
      GOTO100                                                            450
C-----NP.EQ.N2+1                                                         460
   70 K=J2+2*(N2-1)                                                      470
      XP=A(K)                                                            480
      YP=0.0                                                             490
      IP=2                                                               500
      GOTO100                                                            510
C-----NP.GT.N2+1                                                         520
   80 XP=1.0E+20                                                         530
      YP=0.0                                                             540
      IP=2                                                               550
```

```
C-----FINISHED                                                          560
  100 RETURN                                                            570
      ENTRY IFPDS(A,LA)                                                 580
      GOTO 100                                                          590
      END                                                               600
```

```
      SUBROUTINE IPDS(JA,NP,XP,YP,IP)                                    10
C=====INTERPOLATE POINT IN /DENS/ STORAGE============================     20
C     JA - STARTING INDEX IN /DENS/ ARRAY A OF THE TAB1 RECORD             30
C     XP - GIVEN VALUE OF X                                                40
C     YP - CORRESPONDING(DESIRED) VALUE OF Y                               50
C     IP - INTERPOLATION CODE USED TO COMPUTE YP                           60
C     NP - INDEX OF X AND Y ARRAYS SUCH THAT XP LIES                       70
C          BETWEEN X(NP) AND X(NP+1). IF NP.GT.0 ON INPUT, IT              80
C          IS USED TO START SEARCH.                                        90
C     JA, XP, AND NP(GUESSED VALUE OR 0) ARE GIVEN AS INPUT               100
C     YP, IP, AND NP(CORRECT VALUE) ARE OUTPUT FROM S.R.                  110
C     IF XP.LT.X(1),  YP=0.0, NP=0, IP=1                                  120
C     IF XP.GT.X(N2), YP=0.0, NP=N2+1, IP=1, WHERE N2 IS THE              130
C          NUMBER OF X,Y VALUES GIVEN IN THE TAB1 RECORD.                 140
C-----ERROR STOP 314, IMPROPER INTERPOLATION TABLE                       150
      DIMENSION LA(1),A(1)                                               160
C-----INITIALIZE                                                        170
   10 N1=LA(JA+7)                                                        180
      N2=LA(JA+8)                                                        190
      J1=JA+9                                                            200
      J2=J1+2*N1                                                         210
C-----TEST FOR XP OUT OF RANGE                                          220
   20 IF(XP.GE.A(J2))GOTO30                                              230
      YP=0.0                                                             240
      NP=0                                                               250
      IP=1                                                               260
      GOTO100                                                            270
   30 K=J2+2*(N2-1)                                                      280
      IF(XP.LE.A(K))GOTO40                                               290
      YP=0.0                                                             300
      NP=N2+1                                                            310
      IP=1                                                               320
      GOTO100                                                            330
C-----SEARCH TABLE FOR PROPER PANEL                                     340
   40 IF(NP.LE.0)NP=1                                                    350
      IF(NP.GE.N2)NP=N2-1                                                360
      K=J2+2*(NP-1)                                                      370
      IF(XP.LT.A(K))NP=1                                                 380
      N2P=N2-1                                                           390
      DO50N=NP,N2P                                                       400
      NPP=N+1                                                            410
      K=J2+2*N                                                           420
      IF(XP.LT.A(K))GOTO60                                               430
   50 CONTINUE                                                           440
      YP=A(K+1)                                                          450
      IP=2                                                               460
      NP=N2+1                                                            470
      GOTO100                                                            480
C-----FIND INTERPOLATION CODE                                           490
   60 DO70M=1,N1                                                         500
      MP=M                                                               510
      K=J1+2*(M-1)                                                       520
      IF(NPP.LE.LA(K))GOTO80                                             530
   70 CONTINUE                                                           540
      CALL ERROR(314)                                                    550
   80 IP=LA(K+1)                                                         560
C-----INTERPOLATE                                                       570
   90 K=J2+2*(NPP-2)                                                     580
      CALL TERP1(A(K),A(K+1),A(K+2),A(K+3),XP,YP,IP)                     590
      NP=NPP-1                                                           600
C-----FINISHED                                                          610
  100 RETURN                                                            620
      ENTRY IIPDS(A,LA)                                                  630
      GOTO 100                                                           640
      END                                                                650
```

```
      SUBROUTINE LRIDS(MA,JA,LNT)                                        10
C=====LOCATE RECORD IN DENSE STORAGE=================================    20
C     MA  - RECORD INDENT                                                  30
C     JA  - STARTING INDEX IN ARRAY A FOR RECORD MA                        40
C     LNT =0, NORMAL RETURN                                                50
C         =1, RECORD MA NOT IN DENSE STORAGE                               60
C-----NO ERROR STOPS                                                     70
      COMMON/DENS/JMT(100),JAT(100),JTT(100),JLT(100),                   80
     1    JNS,MNS,MX                                                      90
   10 LNT=0                                                              100
      MXP=MNS-1                                                          110
      IF(MXP.LE.0)GOTO30                                                 120
      IF(MA.EQ.0)GOTO30                                                  130
      DO20M=1,MXP                                                        140
      JA=JAT(M)                                                          150
      IF(JMT(M).EQ.MA)GOTO40                                             160
   20 CONTINUE                                                           170
   30 LNT=1                                                              180
   40 RETURN                                                             190
      END                                                                200
```

```
      SUBROUTINE SICKY(NIS,NISO,IF4,IF5,BK,TAB,ITAB)                     10
      COMMON/TAB/ENDTAB(25),KEDTAB(50),KEDR(240),NCNAME(39),NRIF3,NRIF4  20
      DIMENSION KDR1(123),KDR2(1)                                        30
      EQUIVALENCE (KEDR(1),KDR1(1)),(KEDR(124),KDR2(1))                  40
      INTEGER ENDTAB                                                     50
      COMMON/INOUT/NRE1(80),NAMKDK(2),MATKDK,NTYP,NAMISO(20),MATISO(10), 60
     *NRE2(14),NKED,NRE3(3),ITDIM                                        70
      NAMELIST/LIB/IFIRST,IMOD,IDAT                                      80
```

```
      REAL*8 MTNAM,MATTAB(20),DATE,TIME                                 90
      EQUIVALENCE(MTNAM,NAMKDK(1))                                     100
      DIMENSION IF1(100),IF2(100),IF3(100),KEDA(3),IN(20,2),           110
     *IT1(100),IT2(100),IT3(100),IT4(100),                             120
     *NR1(20),NTAB(7),BK(1),TAB(1),ITAB(1)                             130
      EQUIVALENCE(NINP,NI),(NOUTP,NO)                                  140
      INTEGER*2 S,LSED                                                 150
      EQUIVALENCE (LSED,NAME1)                                         160
      DATA S/'SE'/                                                     170
      DATA KEDA/'KEDA','BIBL','IOTH'/                                  180
      DATA NR1/    0,     4,                                           190
     *             0,     3,     0,     3,                            200
     1             1,     1,     0,     4,                            210
     2             1,     3,     3,     8,                            220
     3             2,     6,     0,     3,                            230
     4             3,     8/                                          240
      DIMENSION ISATZ(880),XSATZ(880)                                  250
      EQUIVALENCE(ISATZ(1),XSATZ(1))                                   260
  201 FORMAT('1 INSERTION OF TRANSLATED DATA INTO KEDAK'/2X,39('*')///) 270
      NINP=5                                                           280
      NOUTP=6                                                          290
      IFILE=1                                                          300
      IF11=11                                                          310
      L=1                                                              320
      NREAC2=69                                                        330
      NK=41                                                            340
      IF(NISO.EQ.0)NIS=1                                               350
      CALL RDA1(BK)                                                    360
      DO 200 I=1,880                                                   370
  200 XSATZ(I)=0.0                                                     380
      CALL DATUM(DATE,TIME)                                            390
      CALL SPLIT(DATE,IDAT1,IDAT2,IDAT3)                               400
      IDAT=IDAT1*10000+IDAT2*100+IDAT3                                 410
      WRITE(NOUTP,201)                                                 420
      CALL PRIZEI                                                      430
C-----IFIRST=0 LIBRARY STILL INITIALIZED                               440
C            1 LIBRARY TO BE INITIALIZED                               450
C-----IMOD=0 ISOTOPE NOT YET PRESENT IN LIBRARY                        460
C          1 ISOTOPE YET PRESENT IN LIBRARY AND TO BE REPLACED         470
C                                                                      480
C IDAT=DATE DE COMPILATION                                             490
C                                                                      500
      IFIRST=0                                                         510
      IMOD=0                                                           520
      READ(NINP,LIB)                                                   530
      WRITE(NOUTP,LIB)                                                 540
      MTK=MATKDK                                                       550
      NMTTAB=0                                                         560
      IF(NIS.GT.1)NIS=NIS+1                                            570
      DO 10 MIS=1,NIS                                                  580
      ICYFL=0                                                          590
      IF(NIS.EQ.1)GOTO 11                                              600
      IF(MIS.EQ.1)GOTO 11                                              610
      IFIRST=0                                                         620
      NAMKDK(1)=NAMISO((MIS-1)*2-1)                                    630
      NAMKDK(2)=NAMISO((MIS-1)*2)                                      640
      MTK=MATISO(MIS-1)                                                650
   11 CALL RDA3(MTK)                                                   660
      NTP3=NTYP                                                        670
      IF(MIS.EQ.1)NTYP=NTP3+NRIF4                                      680
      IF(IFIRST)210,210,220                                            690
  220 DO 251 I=1,3                                                     700
      ISATZ(I)=KEDA(I)                                                 710
  251 CONTINUE                                                         720
      NISOT=0                                                          730
      NISOT2=0                                                         740
      NXTR=2                                                           750
      GOTO 260                                                         760
  210 IF(IMOD.EQ.0)GOTO 327                                            770
      CALL LISA(NAMKDK(1),NAMKDK(2),KFIRST)                            780
      IF(KFIRST.EQ.0)GOTO 327                                          790
      IFIRST=1                                                         800
      GOTO 220                                                         810
  327 READ(IFILE'L)(ISATZ(I),I=1,880)                                  820
      NISOT=ISATZ(5)                                                   830
      NISOT2=ISATZ(11)                                                 840
      IF(NISOT.EQ.0.OR.NISOT2.EQ.0)GOTO 220                            850
      IF(NISOT.EQ.NISOT2) GOTO 326                                     860
      WRITE(NOUTP,1001) NISOT,NISOT2                                   870
 1001 FORMAT(' WARNING: NISOT=',I3,' IS NOT EQUAL NISOT2=',I3,'. THIS MA 880
     1Y PRODUCE ERRORS.')                                              890
  326 NXTR=ISATZ(880)                                                  900
      IF(NXTR.GT.1.AND.NXTR.LT.10000) GOTO 329                         910
      WRITE(NOUTP,1002)                                                920
 1002 FORMAT(' LIBRARY UNCHANGED, BECAUSE WORD 880 IN RECORD 1 IS WRONG. 930
     1')                                                               940
      GOTO 328                                                         950
  329 K=ISATZ(7)                                                       960
      JEND=3*NISOT                                                     970
C                                                                      980
C-----FIRST TABLE                                                      990
C                                                                     1000
      DO 290 J=1,NISOT                                                1010
      IF1(J)=ISATZ(K)                                                 1020
      IF2(J)=ISATZ(K+1)                                               1030
      IF3(J)=ISATZ(K+2)                                               1040
      K=K+3                                                           1050
  290 CONTINUE                                                        1060
      KC=ISATZ(13)                                                    1070
      JEND=4*NISOT2                                                   1080
C                                                                     1090
C-----THIRD TABLE                                                     1100
C                                                                     1110
      DO 300 J=1,NISOT2                                               1120
      IT1(J)=ISATZ(KC)                                                1130
      IT2(J)=ISATZ(KC+1)                                              1140
      IT3(J)=ISATZ(KC+2)                                              1150
      IT4(J)=ISATZ(KC+3)                                              1160
      KC=KC+4                                                         1170
  300 CONTINUE                                                        1180
```

```
  260 WRITE(NOUTP,270)NISOT                                              1190
  270 FORMAT(1H ,'NUMBER OF ISOTOPES STILL INCLUDED INTO THE LIBRARY=',  1200
     X       I6)                                                         1210
      N=1                                                               1220
      ISATZ(4)=IDAT                                                     1230
      ISATZ(5)=NISOT+N                                                  1240
      ISATZ(6)=1                                                        1250
      ISATZ(7)=14                                                       1260
      ISATZ(8)=NREAC2                                                   1270
      ISATZ(9)=1                                                        1280
      ISATZ(10)=3*ISATZ(5)+14                                           1290
      ISATZ(11)=NISOT2+N                                                1300
      ISATZ(12)=1                                                       1310
      ISATZ(13)=ISATZ(10)+NREAC2*3                                      1320
      IF(IFIRST)320,320,330                                             1330
  320 KC=KC+3*N                                                         1340
      K1=14                                                             1350
      INW=0                                                             1360
      DO 370 J=1,NISOT                                                  1370
      IF(INW.EQ.1)GOTO 371                                              1380
      IF(IF3(J).LT.MTK   )GOTO 371                                      1390
      IF(IF3(J).NE.MTK)GOTO 372                                         1400
      IF(IMOD.EQ.1)GOTO 372                                             1410
      WRITE(NO,325)NAMKDK(1),NAMKDK(2),MTK                              1420
  325 FORMAT(1HO,'THE ISOTOPE',2X,2A4,2X,I6,' IS YET IN THE LIBRARY.',  1430
     *' IMOD IS NOT SET TO 1. LIBRARY UNCHANGED')                       1440
      RETURN                                                            1450
  372 ISATZ(K1)=NAMKDK(1)                                               1460
      ISATZ(K1+1)=NAMKDK(2)                                             1470
      ISATZ(K1+2)=MTK                                                   1480
      K1=K1+3                                                           1490
      J1=J                                                              1500
      INW=1                                                             1510
      IF(J.GT.NISOT) GOTO 370                                           1520
  371 ISATZ(K1)=IF1(J)                                                  1530
      ISATZ(K1+1)=IF2(J)                                                1540
      ISATZ(K1+2)=IF3(J)                                                1550
      K1=K1+3                                                           1560
      IF(INW.EQ.0.AND.J.EQ.NISOT)GOTO 373                               1570
      GOTO 370                                                          1580
  373 J=J+1                                                             1590
      GOTO 372                                                          1600
  370 CONTINUE                                                          1610
      KC=ISATZ(13)                                                      1620
      INW=0                                                             1630
      DO 380 J=1,NISOT2                                                 1640
      IF(J1.NE.J)GOTO 381                                               1650
  382 ISATZ(KC)=MTK                                                     1660
      ISATZ(KC+1)=NTYP                                                  1670
      ISATZ(KC+2)=NXTR                                                  1680
      ISATZ(KC+3)=1                                                     1690
      KC=KC+4                                                           1700
      INW=1                                                             1710
  381 ISATZ(KC)=IT1(J)                                                  1720
      ISATZ(KC+1)=IT2(J)                                                1730
      ISATZ(KC+2)=IT3(J)                                                1740
      ISATZ(KC+3)=IT4(J)                                                1750
      KC=KC+4                                                           1760
      IF(INW.EQ.0.AND.J.EQ.NISOT2)GOTO 382                              1770
  380 CONTINUE                                                          1780
      GOTO 360                                                          1790
  330 K=14                                                              1800
      KC=ISATZ(13)                                                      1810
      ISATZ(K)=NAMKDK(1)                                                1820
      ISATZ(K+1)=NAMKDK(2)                                              1830
      ISATZ(K+2)=MTK                                                    1840
      K=K+3                                                             1850
      ISATZ(KC)=MTK                                                     1860
      ISATZ(KC+1)=NTYP                                                  1870
      ISATZ(KC+2)=2                                                     1880
      ISATZ(KC+3)=1                                                     1890
      KC=KC+4                                                           1900
  360 K2=ISATZ(10)                                                      1910
      J2=NREAC2*3                                                       1920
      DO 390 J=1,J2                                                     1930
      ISATZ(K2)=KEDR(J)                                                 1940
      K2=K2+1                                                           1950
  390 CONTINUE                                                          1960
C  ADRESSE DE LA FIN DE LA TROISIEME TABLE                              1970
      KC=ISATZ(13)+4*ISATZ(11)-1                                        1980
C  IMPRESSION DES TABLES                                                1990
      WRITE(NOUTP,400)(ISATZ(I),I=1,3)                                  2000
  400 FORMAT(1X,3A4)                                                    2010
      WRITE(NOUTP,410)(ISATZ(I),I=4,13)                                 2020
  410 FORMAT(1X,10I6)                                                   2030
      ITAB1=ISATZ(7)                                                    2040
      ITAB2=ISATZ(10)-1                                                 2050
      DO 430 K=1,2                                                      2060
      WRITE(NOUTP,420)(ISATZ(I),I=ITAB1,ITAB2)                          2070
  420 FORMAT(1X,2A4,6X,I6)                                              2080
      ITAB1=ISATZ(10)                                                   2090
      ITAB2=ISATZ(13)-1                                                 2100
      WRITE(NO,412)                                                     2110
  412 FORMAT(1HO)                                                       2120
  430 CONTINUE                                                          2130
      ITAB1=ISATZ(13)                                                   2140
      WRITE(NOUTP,411)(ISATZ(I),I=ITAB1,KC)                             2150
  411 FORMAT(4I10)                                                      2160
      WRITE(NO,412)                                                     2170
      NT=0                                                              2180
      NTAB(6)=NXTR                                                      2190
      NTAB(7)=1+7*NTYP                                                  2200
C INDEX D'ECRITURE DANS ITAB                                            2210
      JINIT=1                                                           2220
      NTD=NTAB(7)                                                       2230
      K=1                                                               2240
      IF(IF4.EQ.1.OR.IF5.EQ.1)GOTO 805                                  2250
C                                                                       2260
C     DATA FROM FILES 1,2,3                                             2270
C                                                                       2280
```

```
      DO 530 I=1,NK                                                     2290
      NAME1=KEDR(K)                                                     2300
      NAME2=KEDR(K+1)                                                   2310
      CALL RDA2(MTK    ,MIS,NAME1,NAME2,LFT)                            2320
      IF(LFT.EQ.1)GOTO 531                                              2330
      NT=NT+1                                                           2340
      NTDS=NTD+NKED-1                                                   2350
      NTAB(1)=KEDR(K+2)                                                 2360
      IF(NTAB(1).LE.21550) GOTO 570                                     2370
      NTAB(3)=1                                                         2380
      NTAB(4)=1                                                         2390
      GOTO580                                                           2400
  570 KTI=(K+2)*2/3                                                     2410
      NTAB(3)=NR1(KTI-1)                                                2420
      NTAB(4)=NR1(KTI)                                                  2430
  580 IF(NTAB(1).EQ.30050                        ) GOTO 590             2440
      NTAB(2)=0                                                         2450
      GOTO 610                                                          2460
  590 NTAB(2)=1                                                         2470
      NTAB(5)=NRIF3                                                     2480
      GOTO 830                                                          2490
  610 NTAB(5)=NKED/(NTAB(3)+NTAB(4))                                    2500
  830 JP=1                                                              2510
      DO 760 JK1=NTD,NTDS                                               2520
      IF(IOVFL.EQ.0)GOTO 913                                            2530
      IF(JK1-(ITDIM*(IOVFL+1)-IOVFL*880))910,910,911                    2540
  913 IF(JK1-ITDIM)910,910,911                                          2550
  911 IOVFL=IOVFL+1                                                     2560
      IECR=1                                                            2570
      IF(IOVFL.GT.1)IECR=881                                            2580
      IRECR=NXTR+(IOVFL-1)*ITDIM/880                                    2590
      IF(IOVFL.GT.1)IRECR=IRECR-(IOVFL-2)                               2600
      WRITE(IFILE'IRECR)(TAB(IJKL),IJKL=IECR,ITDIM)                     2610
  910 IREEL=JK1-IOVFL*(ITDIM-880)                                       2620
      TAB(IREEL)=BK(JP)                                                 2630
      JP=JP+1                                                           2640
  760 CONTINUE                                                          2650
      IF(NTAB(1).NE.30050)GOTO 841                                      2660
      IS=NTD                                                            2670
      DO 821 IRL=1,NRIF3                                                2680
      ITAB(IS+2)=ITAB(IS+2)+NTAB(6)-1                                   2690
      ITAB(IS+3)=ITAB(IS+3)+NTAB(7)-1                                   2700
      IF(ITAB(IS+3).LE.880)GOTO 822                                     2710
      ITAB(IS+2)=ITAB(IS+2)+1                                           2720
      ITAB(IS+3)=ITAB(IS+3)-880                                         2730
  822 IS=IS+4                                                           2740
  821 CONTINUE                                                          2750
  841 JLS=JINIT+6                                                       2760
      KP=1                                                              2770
      DO 640 J=JINIT,JLS                                                2780
      ITAB(J)=NTAB(KP)                                                  2790
      KP=KP+1                                                           2800
      IF(J-ITDIM)640,641,641                                            2810
  640 CONTINUE                                                          2820
      NE=NKED+NTAB(7)                                                   2830
      IF(NE-880)650,650,660                                             2840
  650 NTAB(7)=NE                                                        2850
      GOTO 670                                                          2860
  660 NTAB(6)=NTAB(6)+(NE-1)/880                                        2870
      NTAB(7)=MOD(NE-1,880)+1                                           2880
  670 JINIT=JINIT+7                                                     2890
      NTD=NTDS+1                                                        2900
  531 K=K+3                                                             2910
  530 CONTINUE                                                          2920
      IF(IOVFL.EQ.0)GOTO 914                                            2930
      IRECR=NXTR+IOVFL*ITDIM/880                                        2940
      IF(IOVFL.GT.1)IRECR=IRECR-(IOVFL-1)                               2950
      WRITE(IFILE'IRECR)(TAB(I),I=881,IREEL)                            2960
      WRITE(IFILE'NXTR)(TAB(I),I=1,880)                                 2970
      GOTO 805                                                          2980
  914 WRITE(IFILE'NXTR)(TAB(I),I=1,NTDS)                                2990
C                                                                       3000
C     DATA FROM FILE 4                                                  3010
C                                                                       3020
  805 IF(NRIF4.EQ.0)GOTO 902                                            3030
      IF(MIS.GT.1)GOTO 901                                              3040
      IF(IF4.EQ.1.OR.IF5.EQ.1)GOTO 806                                  3050
      LASTW=MOD(NTDS-1,880)+1                                           3060
      LASTR=NXTR+(NTDS-1)/880                                           3070
      NTD=LASTW+1                                                       3080
      IF(LASTR.EQ.NXTR) GOTO 807                                        3090
      READ(IFILE'LASTR)(TAB(I),I=881,1760)                              3100
      NTD=LASTW+881                                                     3110
      GOTO 807                                                          3120
  806 LASTW=7*NTYP                                                      3130
      LASTR=NXTR                                                        3140
      JINIT=1                                                           3150
      NTD=1+7*NTYP                                                      3160
  807 NTAB(6)=LASTR                                                     3170
      NTAB(7)=LASTW+1                                                   3180
      IF(NTAB(7).LE.880)GOTO 812                                        3190
      NTAB(7)=1                                                         3200
      NTAB(6)=NTAB(6)+1                                                 3210
  812 K=1                                                               3220
      DO 801 IK=1,39                                                    3230
      IF(NCNAME(IK).EQ.0)GOTO 817                                       3240
      NTAB(5)=NCNAME(IK)                                                3250
      NAME1=KDR2(K)                                                     3260
      NAME2=KDR2(K+1)                                                   3270
      NT=NT+1                                                           3280
      CALL RDA4(MTK,1,NAME1,NAME2,LNT,IR)                               3290
      NTDS=NTD+NKED-1                                                   3300
      IF(NTDS.GE.ITDIM)GOTO 641                                         3310
      READ(IF11'IR)(TAB(I),I=NTD,NTDS)                                  3320
C-----SET NUMERICAL NAME                                                3330
      NTAB(1)=KDR2(K+2)                                                 3340
C-----SET NUMBER OF ARGUMENTS                                           3350
      NTAB(3)=1                                                         3360
C-----SET NUMBER OF FUNCTIONAL VALUES                                   3370
      NTAB(4)=1                                                         3380
```

```
      IF(LSED.EQ.S.AND.NTAB(1).GE.50000)NTAB(4)=3
C-----SET NUMBER OF ADD. NAMES
      IF(NTAB(1).GE.40051)GOTO 808
      NTAB(2)=1
      GOTO 809
  808 IF(NTAB(1).GE.44651)GOTO 810
      NTAB(2)=2
      GOTO 809
  810 IF(NTAB(1).GE.50000)GOTO 811
      NTAB(2)=3
      GOTO 809
  811 NTAB(2)=1
  809 JLS=JINIT+6
      KP=1
      DO 813 J=JINIT,JLS
      ITAB(J)=NTAB(KP)
      KP=KP+1
      IF(J-ITDIM)813,641,641
  813 CONTINUE
C     RELOCATION OF THE REACTION TYPE TABLE
      NIRL=NCNAME(IK)
      IS=NTD
      IS=IS+NTAB(2)-1
      DO 818 IRL=1,NIRL
      ITAB(IS+2)=ITAB(IS+2)+NTAB(6)-1
      ITAB(IS+3)=ITAB(IS+3)+NTAB(7)-1
      IF(ITAB(IS+3).LE.880)GOTO 820
      ITAB(IS+2)=ITAB(IS+2)+1
      ITAB(IS+3)=ITAB(IS+3)-880
  820 IS=IS+NTAB(2)+3
  818 CONTINUE
      IS=IS-1
C
      NKED=NKED+NTAB(7)
      IF(NKED-880)814,814,815
  814 NTAB(7)=NKED
      GOTO 816
  815 NTAB(6)=NTAB(6)+(NKED-1)/880
      NTAB(7)=MOD(NKED-1,880)+1
  816 JINIT=JINIT+7
      NTD=NTDS+1
  817 K=K+3
  801 CONTINUE
  901 JINIT=JINIT-1
      WRITE(NO,900)(ITAB(I),I=1,JINIT)
  900 FORMAT(7I12)
      IF(NT.NE.NTYP)WRITE(NO,740)NT,NTYP
  740 FORMAT(1H1,'NT=',I6,'  NTYP=',I6//)
      JP=JINIT+1
C
C     WRITING OF DATA IN KEDAK
C
      IF(MIS.GT.1)GOTO 902
      IF(IF4.EQ.1.OR.IF5.EQ.1.OR.LASTR.EQ.NXTR) GOTO 819
      WRITE(IFILE'NXTR)(TAB(I),I=1,880)
      WRITE(IFILE'LASTR)(TAB(I),I=881,NTDS)
      NXTRN=(NTDS-881)/880+1+LASTR
      GOTO 900
  819 WRITE(IFILE'NXTR)(TAB(I),I=1,NTDS)
  902 NXTRN=(NTDS-1)/880+1+NXTR
  900 ISATZ(880)=NXTRN
      WRITE(IFILE'1)(ISATZ(I),I=1,880)
      NMTTAB=NMTTAB+1
      MATTAB(NMTTAB)=MTNAM
      WRITE(NO,750)NXTRN
  750 FORMAT(1H1,'NEXT AVAILABLE RECORD IN THE KEDAK LIBRARY',I6//)
   10 CONTINUE
  328 CALL TABPRT(MATTAB,NMTTAB,IFILE)
      RETURN
  641 WRITE(NO,642)
  642 FORMAT(1H1,'OVERFLOW IN TAB VECTOR')
      RETURN
      END
```

```
      SUBROUTINE LISA(INAM1,INAM2,IFIRST)
C     J.DANIELS    AUGUST  1971
C  NTEL(1)=LENGTH OF MATERIAL CONVERSION TABLE/NTEL(2-3)=ADDRESS
C  NTEL(4)=LENGTH OF TYPE     CONVERSION TABLE/NTEL(5-6)=ADDRESS
C  NTEL(7)=LENGTH OF MATERIAL ADDRESS TABLE / NTEL(8-9)=ADDRESS
C  ISATZ(880)=TOTREC=FIRST AVAILABLE RECORD
      INTEGER OLDREC,TOTREC,TEMP
      DIMENSION ISATZ(880),NRECO(62),NREAC(62),NTEL(9),NTEST(62)
      DIMENSION IS2(880)
      NC=6
      NT=1
      NEWREC=1
      OLDREC=1
      ILAST=0
      IFIRST=0
      DO 303 J=1,62
  303 NTEST(J)=0
C  READ FIRST RECORD  ************************************************
      READ(NT'OLDREC)(ISATZ(I),I=1,880)
      DO 301 J=1,9
  301 NTEL(J)=ISATZ(J+4)
      TOTREC=ISATZ(880)
      M=NTEL(1)
      N=NTEL(3)
C  TEST ON FIRST TABLE  **********************************************
      DO 2 J=1,M
      IF(INAM1.EQ.ISATZ(N+3*J-3).AND.INAM2.EC.ISATZ(N+3*J-2))GO TO 6
    2 CONTINUE
      WRITE(NO,10)INAM1,INAM2
   10 FORMAT('  THIS NAME ',2A4,'   NOT FOUND IN TABLE 1 OF THIS LIBRARY
     1'//'  LIBRARY UNCHANGED')
      RETURN
    6 NUM=ISATZ(N+3*J-1)
```

```
      INDEX1=N+3*J-3                                                    340
      INDEX3=N+3*J                                                      350
      M=NTEL(7)                                                         360
      N=NTEL(9)                                                         370
      DO 300 J=1,M                                                      380
      NRECO(J)=ISATZ(N+4*J-2)                                           390
  300 NREAC(J)=ISATZ(N+4*J-3)                                           400
C   SECOND TABLE UNCHANGED  ********************************************* 410
C   TEST ON THIRD TABLE  ************************************************ 420
      DO 3 J=1,M                                                        430
      IF(NUM.EQ.ISATZ(N+4*J-4))GO TO 8                                  440
    3 CONTINUE                                                          450
      WRITE(NO,11)NUM                                                   460
   11 FORMAT('  THE NUMBER CORRESPONDING THE NAME OF THE ASKED ELEMENT', 470
     1I10,'   IS NOT FOUND IN TABLE 3'//'  LIBRARY IS UNCHANGED')       480
      RETURN                                                            490
    8 INDEX2=N+4*J-4                                                    500
      INDEX4=N+4*J                                                      510
      KN=J                                                              520
      IF(M.GT.1)GOTO 200                                                530
      IFIRST=1                                                          540
      RETURN                                                            550
  200 IF(KN.EQ.M)ILAST=1                                                560
      NRECM=NRECO(KN)                                                   570
C  CHANGING FIRST RECORD(BEGIN + TABLE 1 + TABLE 2 TILL THE WRIGHT      580
C                  PLACE IN TABLE 3 )    ********************************* 590
      ISATZ(5)=ISATZ(5)-1                                               600
      ISATZ(10)=ISATZ(10)-3                                             610
      ISATZ(11)=ISATZ(11)-1                                             620
      ISATZ(13)=ISATZ(13)-3                                             630
      NTOT=INDEX2-4                                                     640
      DO 20 J=INDEX1,NTOT                                               650
   20 ISATZ(J)=ISATZ(J+3)                                               660
      IF(ILAST.EQ.1)GOTO 101                                            670
      NINIT=NTOT+1                                                      680
      NFIN=ISATZ(13)+4*ISATZ(11)-1                                      690
      DO 100 J=NINIT,NFIN                                               700
  100 ISATZ(J)=ISATZ(J+7)                                               710
C-----ARRANGEMENT DES NUMEROS DE RECORDS EN ORDRE CROISSANT             720
  101 M1=M-1                                                            730
      DO 102 J=1,M1                                                     740
      J1=J+1                                                            750
      DO 102 JL=J1,M                                                    760
      IF(NRECO(J).LT.NRECO(JL))GOTO 102                                 770
      TEMP=NRECO(J)                                                     780
      NRECO(J)=NRECO(JL)                                                790
      NRECO(JL)=TEMP                                                    800
      TEMP=NREAC(J)                                                     810
      NREAC(J)=NREAC(JL)                                                820
      NREAC(JL)=TEMP                                                    830
  102 CONTINUE                                                          840
      WRITE(NO,1500)(NRECO(J),NREAC(J),J=1,M)                           850
 1500 FORMAT(1X,2I6)                                                    860
C-----L'ISOTOPE A S-T-IL LA PLUS HAUTE ADRESSE                          870
      IF(NRECO(M).GT.NRECM)GOTO 103                                     880
      ISATZ(880)=NRECM                                                  890
      WRITE(NT'1)(ISATZ(I),I=1,880)                                     900
      RETURN                                                            910
  103 DO 104 J=1,M                                                      920
      IF(NRECO(J).EQ.NRECM)GOTO 105                                     930
  104 CONTINUE                                                          940
      STOP                                                              950
  105 KN=J+1                                                            960
      NDIFF=NRECO(KN)-NRECM                                             970
C-----MODIFICATION DE LA 3E TABLE DU RECORD 1                           980
      DO 108 J=1,M1                                                     990
      KLM=ISATZ(13)+J*4-2                                              1000
      IF(ISATZ(KLM).LE.NRECM)GOTO 108                                  1010
      ISATZ(KLM)=ISATZ(KLM)-NDIFF                                      1020
  108 CONTINUE                                                         1030
      ISATZ(880)=ISATZ(880)-NDIFF                                      1040
      WRITE(NT'1)(ISATZ(I),I=1,880)                                    1050
C-----DEPLACEMENT DES ISOTOPES DONT L'ADRESSE EST SUP. A NRECM         1060
      DO 107 J=KN,M                                                    1070
      N1=NRECO(J)                                                      1080
      IF(J.LT.M)GOTO 109                                               1090
      N2=TOTREC-1                                                      1100
      GOTO 110                                                         1110
  109 N2=NRECO(J+1)-1                                                  1120
  110 DO 107 K=N1,N2                                                   1130
      READ(NT'K)(ISATZ(I),I=1,880)                                     1140
      IF(K.GT.N1)GOTO 111                                              1150
      JK=NREAC(J)                                                      1160
      DO 112 JR=1,JK                                                   1170
      J6=JR*7-1                                                        1180
      IF(ISATZ(JR*7-5).EQ.0)GOTO 120                                   1190
      NFNAM=ISATZ(JR*7-5)                                              1200
      NRFN=ISATZ(JR*7-1)                                               1210
      NWFN=ISATZ(JR*7)                                                 1220
      NLONG=ISATZ(JR*7-2)                                              1230
      READ(NT'NRFN)(IS2(I),I=1,880)                                    1240
      NRFNN=NRFN                                                       1250
      DO 121 NLO=1,NLONG                                               1260
      INLO=NWFN+NLO*(3+NFNAM)-2                                        1270
      LESDIF=(INLO-1)/880                                              1280
      INLOLO=INLO-LESDIF*880                                           1290
      IF((NRFN+LESDIF).EQ.NRFNN) GOTO 415                              1300
      WRITE(NT'NRFNN) (IS2(I),I=1,880)                                 1310
      NRFNN=NRFN+LESDIF                                                1320
      READ(NT'NRFNN) (IS2(I),I=1,880)                                  1330
  415 IS2(INLOLO)=IS2(INLOLO)-NDIFF                                    1340
      IF(K.EQ.NRFNN.AND.INLO.LE.880) ISATZ(INLO)=ISATZ(INLO)-NDIFF     1350
  121 CONTINUE                                                         1360
      WRITE(NT'NRFNN)(IS2(I),I=1,880)                                  1370
  120 ISATZ(J6)=ISATZ(J6)-NDIFF                                        1380
  112 CONTINUE                                                         1390
  111 KL=K-NDIFF                                                       1400
      WRITE(NT'KL)(ISATZ(I),I=1,880)                                   1410
      WRITE(NO,1051)K,KL                                               1420
 1051 FORMAT(1X,'K=',I6,'  KL=',I6)                                    1430
```

```
  107 CONTINUE                                                          1440
      RETURN                                                            1450
      END                                                               1460
```

```
      SUBROUTINE TABPRT(MAT,AMAT,IFIL)                                    10
      DIMENSION W(880),IW(880),RW(440),WTYP(1),WMAT(1),XALPH(1),TW(7),    20
     1 ITW(7)                                                             30
      REAL*8 RW,MAT(1),ALL/'ALL'/,MATNAM(100),TYP(100),ALPHAN,            40
     1 NF/'MISSING'/                                                      50
      REAL*8 TABLES/'TABLES'/,TITLE,END/'  END'/,BLANK/'          '/      60
      INTEGER AMACON(2),ATYCON(2),AMATAB(2),ISNAM(100),LISTYP(100),       70
     1AISTYP(2,100),ADDNAM(7,10),ITYP(100),IMAT(100),XALPH                80
      EQUIVALENCE (W(1),IW(1),RW(1)),(WTYP(1),TYP(1)),                    90
     1 (WMAT(1),MATNAM(1)),(XALPH(1),ALPHAN),(TW(1),ITW(1))              100
      DATA LRECD/880/,ILM/58/                                            110
      KOUT=6                                                             120
      CALL INITPG(IFIL)                                                  130
      CALL PAGE(TABLES,IL)                                               140
      WRITE(KOUT,623)                                                    150
  623 FORMAT(   /2(10X,80('*')/),30X,'PRINTOUT OF KEDAKTABLES.'/         160
     1 2(10X,80('*')/))                                                  170
      CALL PRIZEI                                                        180
      IL=IL+10                                                           190
      WRITE(KOUT,624)                                                    200
  624 FORMAT(/2X,'***** THE PAIR PRECEDING EACH LINE OF INFORMATION ARE  210
     1THE ENTRYPOINTS,WHERE IT WAS RETRIEVED FROM. *****'/               220
     22X,'*****     ADRESSES ARE ALWAYS GIVEN AS PAIR (RECORD,WORD).',   230
     3T102,5('*')//)                                                     240
      IL=IL+5                                                            250
      READ(IFIL'1) RW                                                    260
      WRITE(KOUT,600) (W(I),I=1,3)                                       270
  600 FORMAT(5X,'(  1,  1) IDENTIFICATION:',1X,3A4/)                     280
      WRITE(KOUT,601) IW(4)                                              290
  601 FORMAT(5X,'(  1,  4) DATE=',I7/)                                   300
      IL=IL+2                                                            310
C                                                                        320
C         FETCH ADRESSES OF FIRST THREE TABLES.                          330
C                                                                        340
      LMACON=IW(5)                                                       350
      AMACON(1)=IW(6)                                                    360
      AMACON(2)=IW(7)                                                    370
      WRITE(KOUT,602) LMACON,AMACON                                      380
  602 FORMAT(5X,'(  1,  5) LENGTH OF MATCONVERSION-TABLE=',I5/           390
     1 5X,'(  1,  6) ADRESS OF MATCONVERSION-TABLE= (',I3,',',I3,')')    400
      IL=IL+2                                                            410
      LTYCON=IW(8)                                                       420
      ATYCON(1)=IW(9)                                                    430
      ATYCON(2)=IW(10)                                                   440
      WRITE(KOUT,603) LTYCON,ATYCON                                      450
      IL=IL+3                                                            460
      LMATAB=IW(11)                                                      470
      AMATAB(1)=IW(12)                                                   480
      AMATAB(2)=IW(13)                                                   490
      WRITE(KOUT,604) LMATAB,AMATAB                                      500
      IL=IL+3                                                            510
  603 FORMAT(/5X,'(  1,  8) LENGTH OF TYPECONVERSION-TABLE=',I5/         520
     1 5X,'(  1,  9) ADRESS OF TYPECONVERSION-TABLE= (',I3,',',I3,')')   530
  604 FORMAT(/5X,'(  1, 11) LENGTH OF MATERIALADRESSTABLE=',I5/          540
     1 5X,'(  1, 12) ADRESS OF MATERIALADRESSTABLE= (',I3,',',I3,')')    550
C                                                                        560
C         FETCH MATERIAL CONVERSION TABLE.                               570
C                                                                        580
      LX=1                                                               590
      IF(AMACON(1).EQ.LX) GOTO 10                                        600
      LX=AMACON(1)                                                       610
      READ(IFIL'LX) RW                                                   620
   10 LS=AMACON(2)                                                       630
      LE=LMACON*3+LS-1                                                   640
      CALL PAGE(TABLES,IL)                                               650
      WRITE(KOUT,605)                                                    660
  605 FORMAT(//5X,10('*'),2X,'MATERIALCONVERSIONTABLE:',2X,10('*')//11X, 670
     1'ALPHA',T21,'NUMERIC'/)                                            680
      IL=IL+6                                                            690
      J=1                                                                700
      K=1                                                                710
      M=LS-1                                                             720
      DO 20 I=LS,LE,3                                                    730
      M=M+1                                                              740
      IF(M.LE.LRECD) GOTO 12                                             750
      LX=LX+1                                                            760
      READ(IFIL'LX) RW                                                   770
      M=1                                                                780
   12 MS=M                                                               790
      LXS=LX                                                             800
      WMAT(J)=W(M)                                                       810
      M=M+1                                                              820
      IF(M.LE.LRECD) GOTO 14                                             830
      LX=LX+1                                                            840
      READ(IFIL'LX) RW                                                   850
      M=1                                                                860
   14 WMAT(J+1)=W(M)                                                     870
      M=M+1                                                              880
      IF(M.LE.LRECD) GOTO 16                                             890
      LX=LX+1                                                            900
      READ(IFIL'LX) RW                                                   910
      M=1                                                                920
   16 IMAT(K)=IW(M)                                                      930
      IF(IL.GT.ILM) CALL PAGE(TABLES,IL)                                 940
      IL=IL+1                                                            950
      WRITE(KOUT,615) LXS,MS,WMAT(J),WMAT(J+1),IMAT(K)                   960
      J=J+2                                                              970
      K=K+1                                                              980
   20 CONTINUE                                                           990
  615 FORMAT(1X,'(',I3,',',I3,')',1X,2A4,T20,I7)                        1000
C                                                                       1010
C         FETCH TYPE CONVERSION TABLE.                                  1020
C                                                                       1030
```

```
      IF(ATYCON(1).EQ.LX) GOTO 110                                      1040
      LX=ATYCON(1)                                                      1050
      READ(IFIL'LX) RW                                                  1060
  110 LS=ATYCON(2)                                                      1070
      LE=LTYCON*3+LS-1                                                  1080
      CALL PAGE(TABLES,IL)                                              1090
      WRITE(KOUT,606)                                                   1100
  606 FORMAT(//5X,10('*'),2X,'TYPECONVERSIONTABLE:',2X,10('*')//11X,    1110
     1'ALPHA',T21,'NUMERIC'/)                                           1120
      J=1                                                               1130
      K=1                                                               1140
      M=LS-1                                                            1150
      DO 120 I=LS,LE,3                                                  1160
      M=M+1                                                             1170
      IF(M.LE.LRECD) GOTO 112                                           1180
      LX=LX+1                                                           1190
      READ(IFIL'LX) RW                                                  1200
      M=1                                                               1210
  112 MS=M                                                              1220
      LXS=LX                                                            1230
      WTYP(J)=W(M)                                                      1240
      M=M+1                                                             1250
      IF(M.LE.LRECD) GOTO 114                                           1260
      LX=LX+1                                                           1270
      READ(IFIL'LX) RW                                                  1280
      M=1                                                               1290
  114 WTYP(J+1)=W(M)                                                    1300
      M=M+1                                                             1310
      IF(M.LE.LRECD) GOTO 116                                           1320
      LX=LX+1                                                           1330
      READ(IFIL'LX) RW                                                  1340
      M=1                                                               1350
  116 ITYP(K)=IW(M)                                                     1360
      IF(IL.GT.ILM) CALL PAGE(TABLES,IL)                                1370
      IL=IL+1                                                           1380
      WRITE(KOUT,616) LXS,MS,WTYP(J),WTYP(J+1),ITYP(K)                  1390
  616 FORMAT(1X,'(',I3,',',I3,')',1X,2A4,T20,I7)                        1400
      K=K+1                                                             1410
      J=J+2                                                             1420
  120 CONTINUE                                                          1430
C                                                                       1440
C         FETCH MATERIAL ADRESS TABLE.                                  1450
C                                                                       1460
      IF(AMATAB(1).EQ.LX) GOTO 210                                      1470
      LX=AMATAB(1)                                                      1480
      READ(IFIL'LX)RW                                                   1490
  210 LS=AMATAB(2)                                                      1500
      LE=LMATAB*4+LS-1                                                  1510
      J=0                                                               1520
      CALL PAGE(TABLES,IL)                                              1530
      WRITE(KOUT,617)                                                   1540
  617 FORMAT(//5X,10('*'),2X,'MATERIALADRESSTABLE:',2X,10('*')//12X,    1550
     1'NAME',T30,'LENGTH',T40                                           1560
     1,'ADRESS',T20,'ALPHA'/)                                           1570
      M=LS-1                                                            1580
      DO 220 I=LS,LE,4                                                  1590
      J=J+1                                                             1600
      M=M+1                                                             1610
      IF(M.LE.LRECD) GOTO 208                                           1620
      LX=LX+1                                                           1630
      READ(IFIL'LX) RW                                                  1640
      M=1                                                               1650
  208 MS=M                                                              1660
      LXS=LX                                                            1670
      ISNAM(J)=IW(M)                                                    1680
      M=M+1                                                             1690
      IF(M.LE.LRECD) GOTO 211                                           1700
      LX=LX+1                                                           1710
      READ(IFIL'LX) RW                                                  1720
      M=1                                                               1730
  211 LISTYP(J)=IW(M)                                                   1740
      M=M+1                                                             1750
      IF(M.LE.LRECD) GOTO 212                                           1760
      LX=LX+1                                                           1770
      READ(IFIL'LX) RW                                                  1780
      M=1                                                               1790
  212 AISTYP(1,J)=IW(M)                                                 1800
      M=M+1                                                             1810
      IF(M.LE.LRECD) GOTO 214                                           1820
      READ(IFIL'LX) RW                                                  1830
      M=1                                                               1840
      LX=LX+1                                                           1850
  214 AISTYP(2,J)=IW(M)                                                 1860
      DO 215 L=1,LMACON                                                 1870
      IF(IMAT(L).NE.ISNAM(J)) GOTO 215                                  1880
      ALPHAN=MATNAM(L)                                                  1890
      GOTO 216                                                          1900
  215 CONTINUE                                                          1910
  216 IF(IL.GT.ILM) CALL PAGE(TABLES,IL)                                1920
      IL=IL+1                                                           1930
      WRITE(KOUT,607) LXS,MS,ISNAM(J),ALPHAN,LISTYP(J),(AISTYP(K,J),K=1,1940
     12)                                                                1950
  607 FORMAT(1X,'(',I3,',',I3,')',1X,I7,T20,A8,T30,I5,T38,'(',I3,',',I3,1960
     1')')                                                              1970
  220 CONTINUE                                                          1980
C                                                                       1990
C         FOR EACH MATERIAL PRINT ADRESSTABLE OF TYPES.                 2000
C                                                                       2010
      DO 1000 L=1,LMATAB                                                2020
      DO 225 I=1,LMACON                                                 2030
      IF(IMAT(I).NE.ISNAM(L)) GOTO 225                                  2040
      ALPHAN=MATNAM(I)                                                  2050
      GOTO 226                                                          2060
  225 CONTINUE                                                          2070
      ALPHAN=NF                                                         2080
  226 IF(MAT(1).EQ.ALL) GOTO 230                                        2090
      DO 240 I=1,NMAT                                                   2100
      IF(MAT(I).NE.ALPHAN    ) GOTO 240                                 2110
      MAT(I)=BLANK                                                      2120
      GOTO 230                                                          2130
```

```
  240 CONTINUE                                                          2140
      GOTO 1000                                                         2150
  230 TITLE=ALPHAN                                                      2160
      CALL PAGE(TITLE,IL)                                               2170
      WRITE(KOUT,618) ISNAM(L),ALPHAN                                   2180
  618 FORMAT(//5X,10('*'),2X,'TYPETABLE FOR:',I7,'    ALPHA NAME: ',A8,  2190
     12X,10('*')//                                                      2200
     1T15,'TYPE',T33,'ADD.NAMES',T45,'ARG.NUM',T55,'FW.NUM',T65,'DP.NUM' 2210
     2,T75,'ADRESS',T21,'ALPHA'/)                                       2220
      IL=IL+6                                                           2230
      IF(AISTYP(1,L).EQ.LX) GOTO 300                                    2240
      LX=AISTYP(1,L)                                                    2250
      READ(IFIL'LX) RW                                                  2260
  300 LS=AISTYP(2,L)                                                    2270
      LE=LISTYP(L)*7+LS-1                                               2280
      KW=0                                                              2290
      M=LS-1                                                            2300
      DO 310 I=LS,LE,7                                                  2310
      M=M+1                                                             2320
      IF(M.LE.LRECD) GOTO 302                                           2330
      LX=LX+1                                                           2340
      READ(IFIL'LX) RW                                                  2350
      M=1                                                               2360
  302 MS=M                                                              2370
      LXS=LX                                                            2380
C                                                                       2390
C         ALPHA NAME OF THIS TYPE.                                      2400
      DO 305 K=1,LTYCON                                                 2410
      IF(ITYP(K).NE.IW(M)) GOTO 305                                     2420
      ALPHAN=TYP(K)                                                     2430
      GOTO 306                                                          2440
  305 CONTINUE                                                          2450
      ALPHAN=NF                                                         2460
  306 ITW(1)=IW(M)                                                      2470
      DO 308 J=2,7                                                      2480
      M=M+1                                                             2490
      IF(M.LE.LRECD) GOTO 307                                           2500
      LX=LX+1                                                           2510
      READ(IFIL'LX) RW                                                  2520
      M=1                                                               2530
  307 ITW(J)=IW(M)                                                      2540
  308 CONTINUE                                                          2550
      IF(IL.GT.ILM) CALL PAGE(TITLE,IL)                                 2560
      WRITE(KOUT,608) LXS,MS,ITW(1),ALPHAN,(ITW(J),J=2,7)               2570
  608 FORMAT(1X,'(',I3,',',I3,')',1X,I7 ,2X,A8,3I10,I12,2X,'(',I3,',',I3 2580
     1,')')                                                             2590
      IL=IL+1                                                           2600
      NW=ITW(2)                                                         2610
      IF(NW.EQ.0) GOTO 310                                              2620
      KW=KW+1                                                           2630
      ADDNAM(1,KW)=ITW(1)                                               2640
      ADDNAM(2,KW)=NW                                                   2650
      ADDNAM(3,KW)=ITW(5)                                               2660
      ADDNAM(4,KW)=ITW(6)                                               2670
      ADDNAM(5,KW)=ITW(7)                                               2680
      ADDNAM(6,KW)=XALPH(1)                                             2690
      ADDNAM(7,KW)=XALPH(2)                                             2700
  310 CONTINUE                                                          2710
      IF(KW.EQ.0) GOTO 1000                                             2720
      IF(IL.GT.45) CALL PAGE(TITLE,IL)                                  2730
      WRITE(KOUT,610)                                                   2740
      IL=IL+4                                                           2750
      DO 400 K=1,KW                                                     2760
      IF(ADDNAM(4,K).EQ.LX) GOTO 406                                    2770
      LX=ADDNAM(4,K)                                                    2780
      READ(IFIL'LX) RW                                                  2790
  406 NW=ADDNAM(2,K)                                                    2800
      LS=ADDNAM(5,K)                                                    2810
      LC=NW+3                                                           2820
      LE=ADDNAM(3,K)*LC+LS-1                                            2830
      IF(IL.GT.50) CALL PAGE(TITLE,IL)                                  2840
      WRITE(KOUT ,611) ADDNAM(1,K),ADDNAM(6,K ),ADDNAM(7,K )            2850
      IL=IL+4                                                           2860
      M=LS-1                                                            2870
      DO 420 I=LS,LE,LC                                                 2880
      LE1=I+NW-1                                                        2890
      MS=M+1                                                            2900
      LXS=LX                                                            2910
      IF(MS.GT.LRECD) LXS=LXS+1                                         2920
      IF(MS.GT.LRECD) MS=1                                              2930
      KJ=0                                                              2940
      DO 410 J=I,LE1                                                    2950
      KJ=KJ+1                                                           2960
      M=M+1                                                             2970
      IF(M.LE.LRECD) GOTO 408                                           2980
      LX=LX+1                                                           2990
      READ(IFIL'LX) RW                                                  3000
      M=1                                                               3010
  408 TW(KJ)=W(M)                                                       3020
  410 CONTINUE                                                          3030
       IF(IL.GT.ILM) CALL PAGE(TITLE,IL)                                3040
      WRITE(KOUT,612) LXS,MS,(TW(J),J=1,KJ)                             3050
      JS=I+NW                                                           3060
      LE1=JS+2                                                          3070
      MS=M+1                                                            3080
      LXS=LX                                                            3090
      IF(MS.GT.LRECD) LXS=LXS+1                                         3100
      IF(MS.GT.LRECD) MS=1                                              3110
      KJ=0                                                              3120
      DO 416 J=JS,LE1                                                   3130
      KJ=KJ+1                                                           3140
      M=M+1                                                             3150
      IF(M.LE.LRECD) GOTO 414                                           3160
      LX=LX+1                                                           3170
      READ(IFIL'LX) RW                                                  3180
      M=1                                                               3190
  414 ITW(KJ)=IW(M)                                                     3200
  416 CONTINUE                                                          3210
      WRITE(KOUT,613) LXS,MS,(ITW(J),J=1,KJ)                            3220
      IL=IL+3                                                           3230
```

```
  420 CONTINUE                                                          3240
  400 CONTINUE                                                          3250
  610 FORMAT(///5X,5('*'),2X,'TABLES FOR TYPES WITH MORE THAN ONE NAME',3260
     12X,5('*'))                                                        3270
  611 FORMAT(//5X,'** TYPE',I7,'    ALPHANAME:',2A4,' **'/)             3280
  612 FORMAT(10X,'(',I3,',',I3,')',4X,' ADD. NAMES:',5(1PE13.5))        3290
  613 FORMAT(10X,'(',I3,',',I3,')',4X,'DP.NUM,ADR.:',   I5,2X,'(',I3,',' 3300
     1,I3,')'/)                                                         3310
 1000 CONTINUE                                                          3320
      IF(MAT(1).EQ.ALL) GOTO 1600                                       3330
      K=0                                                               3340
      DO 1100 I=1,NMAT                                                  3350
      IF(MAT(I).EQ.BLANK) GOTO 1100                                     3360
      IF(K.EQ.0) CALL PAGE(END,IL)                                      3370
      K=1                                                               3380
      IF(IL.GT.ILM) CALL PAGE(END,IL)                                   3390
      WRITE(KOUT,622) MAT(I)                                            3400
      IL=IL+5                                                           3410
  622 FORMAT(//10X,'***** ',A8,' NOT FOUND *****'//)                    3420
 1100 CONTINUE                                                          3430
      IF(K.EQ.1) GOTO 2000                                              3440
 1600 CALL PAGE(END,IL)                                                 3450
      WRITE(KOUT,620)                                                   3460
  620 FORMAT(//////'    ***** NORMAL END OF TABPRT *****')              3470
 2000 RETURN                                                            3480
      END                                                               3490
```

```
      SUBROUTINE PAGE(TITLE,IL)                                         10
      REAL*8 TITLE                                                      20
      DATA KOUT/6/                                                      30
      NPG=NPG+1                                                         40
      WRITE(KOUT,600) IFIL,NPG,TITLE                                    50
  600 FORMAT('1',91X,'FT-NUM-',I2,'    PAGE-',I3,'    ***',A8,'***'//)  60
      IL=3                                                              70
      RETURN                                                            80
      ENTRY INITPG(IFIL)                                                90
      NPG=0                                                             100
      RETURN                                                            110
      END                                                               120
```

```
      SUBROUTINE SPLIT(A,I1,I2,I3)                                      10
      REAL*8 A,AA                                                       20
      INTEGER*2 I(4),IPZ(10),IZP(10)                                    30
      EQUIVALENCE (AA,I(1))                                             40
      DATA IZP/'0','1','2','3','4','5','6','7','8','9'/                50
      DATA IPZ/'.0','.1','.2','.3','.4','.5','.6','.7','.8','.9'/      60
      AA=A                                                              70
      CALL CON(I(1),I1)                                                 80
      CALL CON(I(4),I3)                                                 90
      I2E=0                                                             100
      DO 1 J=1,10                                                       110
      IF(I(3).EQ.IZP(J))GOTO 2                                          120
      I2E=I2E+1                                                         130
    1 CONTINUE                                                          140
    2 CONTINUE                                                          150
      I2Z=0                                                             160
      DO 3 J=1,10                                                       170
      IF(I(2).EQ.IPZ(J))GOTO 4                                          180
      I2Z=I2Z+1                                                         190
    3 CONTINUE                                                          200
    4 CONTINUE                                                          210
      I2=I2E+10*I2Z                                                     220
      IF(I2E.GE.10.OR.I2Z.GE.10)WRITE(6,100)I2E,I2Z                     230
  100 FORMAT(///' FEHLER IN SPLIT . I2E=',I4,'  I2Z=',I4)               240
      RETURN                                                            250
      END                                                               260
```

```
      SUBROUTINE CON(IA,IN)                                             10
      INTEGER*2 IA,IN2(2),II(10)                                        20
      DATA IFOR/'I2  '/                                                 30
      DATA II/'00','01','02','03','04','05','06','07','08','09'/       40
      IM=0                                                              50
      IN=0                                                              60
      DO 1 J=1,100                                                      70
      IF(J.GT.10)GOTO 3                                                 80
      IN2(1)=II(J)                                                      90
      GOTO 4                                                            100
    3 CONTINUE                                                          110
      CALL CONVX(IM,IN2,IFOR)                                           120
    4 CONTINUE                                                          130
      IF(IN2(1).EQ.IA)GOTO 2                                            140
      IM=IM+1                                                           150
      IN=IN+1                                                           160
    1 CONTINUE                                                          170
    2 CONTINUE                                                          180
      IF(IN.GE.100)WRITE(6,100)IA,IN                                    190
  100 FORMAT(///' FEHLER IN CON . IA=',A4,'  IN=',I4)                   200
      RETURN                                                            210
      END                                                               220
```

```
      SUBROUTINE PRIEIN(IEIN,IOUT,ZEILMA)                               10
C                                                                       20
C     AUSDRUCK DER VON DER EINHEIT IEIN KOMMENDEN EINGABE AUF IOUT.     30
C     DIE EINGABE-EINHEIT MUSS DIE DCB-SPEZIFIKATIONEN RECFM=F BZW.     40
C     RECFM=FB SOWIE LRECL=80 HABEN.                                    50
C     WEITERHIN SOLLTE DIE DATEI AN IHREM LOGISCHEN ANFANG STEHEN.      60
C     DIE AUSGABEEINHEIT MUSS EINE LOGISCHE SATZLAENGE VON MINDESTENS   70
C     91 BYTE HABEN.                                                    80
```

```
C                                                                       90
      INTEGER ZEIL,ZEILMA                                              100
      DIMENSION A(20)                                                  110
      LOGICAL*4 SECOND                                                 120
      SECOND=.FALSE.                                                   130
      IF(ZEILMA.LT.10.OR.ZEILMA.GT.60) ZEILMA=60                       140
    1 WRITE(IOUT,100)                                                  150
  100 FORMAT(1H1)                                                      160
      WRITE(IOUT,101)                                                  170
  101 FORMAT(' ',90('*'))                                              180
      WRITE(IOUT,102)                                                  190
  102 FORMAT(' ',34('*'),' AUSDRUCK DER EINGABE ')                     200
      IF(.NOT.SECOND) WRITE(IOUT,103)                                  210
  103 FORMAT('+',56X,34('*'))                                          220
      IF(SECOND) WRITE(IOUT,104)                                       230
  104 FORMAT('+',56X,'( FORTSETZUNG ) ',18('*'))                       240
      WRITE(IOUT,101)                                                  250
      WRITE(IOUT,105)                                                  260
  105 FORMAT(' ')                                                      270
      ZEIL=4                                                           280
      IF(SECOND) GOTO 2                                                290
      N=0                                                              300
      SECOND=.TRUE.                                                    310
    3 READ(IEIN,106,END=10) A                                          320
  106 FORMAT(20A4)                                                     330
      N=N+1                                                            340
    2 ZEIL=ZEIL+1                                                      350
      IF(ZEIL.GT.ZEILMA) GOTO 1                                        360
      WRITE(IOUT,107) A,N                                              370
  107 FORMAT(' ',20A4,'     #',I5)                                     380
      GOTO 3                                                           390
   10 WRITE(IOUT,105)                                                  400
      WRITE(IOUT,101)                                                  410
      WRITE(IOUT,102)                                                  420
      WRITE(IOUT,108)                                                  430
  108 FORMAT('+',56X,' ( ENDE ) ',24('*'))                             440
      WRITE(IOUT,101)                                                  450
      REWIND IEIN                                                      460
      RETURN                                                           470
      END                                                              480
```

```
BUFREG   START                                                          10
*        AUFRUF: CALL BUFREG(INTREG,&LABEL)                             20
*        INTREG ENTHAELT NACH DEM AUFRUF IN K-BYTE DIE REGION, DIE FUER 30
*        PUFFER BENOETIGT WIRD.                                         40
*        IN DER TIOT WERDEN ALLE VORHANDENEN DD-NAMEN DER FORM FTXXF001 50
*        MIT AUSNAHME DES DATASETS FUER DIE DRUCKAUSGABE                60
*        ( DIE FILENUMMER STEHEN IM COMMON INOUT )                      70
*        DURCHGEGANGEN, DIE BLKSIZE GEHOLT, DIE PUFFERANZAHL GLEICH 2   80
*        GESETZT, DER FUER DIE PUFFER BENOETIGTE PLATZ BERECHNET UND    90
*        DIESER ZU INTREG ADDIERT.                                     100
*        BEI VORKOMMEN DER DDNAMEN SYSUDUMP BZW. SYSABEND WERDEN JE 2K 110
*        ZU INTREG ADDIERT.                                            120
*        TRITT EIN FEHLER AUF, SO WIRD RETURN 1 GEMACHT, WOBEI DIE IN  130
*        INTREG STEHENDE NEGATIVE ZAHL DEN FEHLERGRUND ANGIBT.         140
*              2     ''        ''  AUSGABE   ''    ''   .             150
*              5     ''        ''  AUSGABE   ''    ''   .             160
*              7   PROGRAMM- ODER MASCHINENFEHLER.                     170
*              9   FEHLERMELDUNG STEHT IN SYSMSG.                      180
*                                                                      190
Q        EQU   3              HILFSSPEICHER. ( KURZZEITSPEICHER ) .    200
AUS      EQU   5              FILENUMMER VON AUSGABE.                  210
REGION   EQU   11             SUMME DER PUFFERREGION.                  220
DDLAUF   EQU   5                                                       230
DDINCR   EQU   6                                                       240
I        EQU   8                                                       250
ACHT     EQU   12                                                      260
J        EQU   9                                                       270
NEWOLD   EQU   4                                                       280
BRANCH1  EQU   14                                                      290
BRANCH2  EQU   2                                                       300
RBLKS    EQU   7                                                       310
*                                                                      320
*                                                                      330
*                                                                      340
         SAVE  (14,12),,*                                              350
         USING BUFREG,10                                               360
         EXTRN INOUT                                                   370
         LR    10,15                                                   380
*        SAVE-AREA-VERKETTUNG.                                         390
         LR    3,13                                                    400
         LA    13,SAVE                                                 410
         ST    13,8(0,3)                                               420
         ST    3,4(0,13)                                               430
         L     Q,0(1)          ADRESSE DER ZU UEBERGEBENDEN REGION.    440
         ST    Q,PARMADR                                               450
         L     Q,=A(INOUT)     ADRESSE VON COMMON INOUT.               460
         L     AUS,0(Q)        FILE-NUMMER VON AUSGABE.                470
         C     AUS,=F'1'                                               480
         BL    FEHLER2                                                 490
         C     AUS,=F'99'                                              500
         BH    FEHLER5                                                 510
*        KONVERTIERUNG DER FILENUMMER IN DEZIMALDARSTELLUNG UND        520
*        SPEICHERN NACH FTAUS.                                         530
         CVD   AUS,DHILF                                               540
         UNPK  DHILF+6(2),DHILF+6(2)                                   550
```

```
         MVZ   DHILF+7(1),DHILF+6                                       560
         MVC   FTAUS+2(2),DHILF+6                                       570
*        FREIE REGION VOR GETMAIN.                                      580
         LA    1,PARM1                                                  590
         L     15,=V(FREESP)                                            600
         BALR  14,15                                                    610
*        ADRESSE DER TIOT HOLEN. ADRESSE DES 1. DDNAMEN SETZEN.         620
         SR    REGION,REGION  PUFFERPLATZ NULL SETZEN.                  630
*        MISREG NULL SETZEN.                                            640
         ST    REGION,MISREG                                            650
*        ERROR NULL SETZEN.                                             660
         ST    REGION,ERROR                                             670
         EXTRACT  TIOTADR,FIELDS=(TIOT)                                 680
         L     Q,TIOTADR                                                690
         LA    DDLAUF,24(Q)   ADRESSE DES 1. DD-ENTRY.                  700
*        PRUEFEN OB LETZTER DD-ENTRY.                                   710
         LA    ACHT,8                                                   720
WEITER   SR    Q,Q                                                      730
         IC    Q,0(DDLAUF)                                              740
         C     Q,=F'0'                                                  750
         BE    RETURN                                                   760
         LR    DDINCR,Q       LAENGE DES DD-ENTRIES.                    770
*        DDNAMEN HOLEN.                                                 780
         MVC   DDNAM(8),4(DDLAUF)                                       790
         LD    0,DDNAM                                                  800
*        VERGLEICH MIT FTAUS.                                           810
         CD    0,FTAUS                                                  820
         BE    ERHOEHE                                                  830
*        HAT NAMEN DIE FORM FTXXFYY, SYSABEND BZW. SYSUDUMP?            840
         CD    0,SYSAB                                                  850
         BE    ADD2K          ES WERDEN 2K-BYTE ADDIERT.                860
         CD    0,SYSUD                                                  870
         BE    ADD2K          ES WERDEN 2K-BYTE ADDIERT.                880
         LH    Q,DDNAM                                                  890
         C     Q,=XL4'FFFFC6E3'                                         900
         BNE   ERHOEHE                                                  910
         SR    Q,Q                                                      920
         IC    Q,DDNAM+4                                                930
         C     Q,=XL4'000000C6'                                         940
         BNE   ERHOEHE                                                  950
         L     I,=F'-8'                                                 960
         LA    J,2                                                      970
LABWEI   IC    Q,DDNAM(J)                                               980
         C     Q,=XL4'000000F0'                                         990
         BL    ERHOEHE                                                 1000
         C     Q,=XL4'000000F9'                                        1010
         BH    ERHOEHE                                                 1020
         AR    I,ACHT                                                  1030
         B     LABEL(I)                                                1040
LABEL    LA    J,3                                                     1050
         B     LABWEI                                                  1060
         LA    J,5                                                     1070
         B     LABWEI                                                  1080
         LA    J,6                                                     1090
         B     LABWEI                                                  1100
         LA    J,7                                                     1110
         B     LABWEI                                                  1120
DDWRIT   EQU   *                                                       1130
*        DDNAME IN DCB BRINGEN.                                         1140
         MVC   DCBALL+40(8),DDNAM                                      1150
*        LESEN VON JFCB.                                                1160
         RDJFCB  DCBALL                                                1170
*        RDJFCB ERFOLGREICH?                                            1180
         LTR   15,15                                                   1190
         BNZ   FEHLER7                                                 1200
*        DATASET NEW ODER OLD? ( NEWOLD=1 DATASET IST OLD.             1210
*                               NEWOLD=0 DATASET IST NEW. )            1220
         LA    BRANCH1,M7                                              1230
         LA    BRANCH2,M6                                              1240
         SR    NEWOLD,NEWOLD                                           1250
         TM    JFCB+87,X'C0'                                           1260
*        BEI OLD UND MOD WIRD NEWOLD=1 GESETZT. BEI CC SPRUNG NACH      1270
*        FEHLER7.                                                       1280
         BO    M1                                                      1290
         BZ    FEHLER7                                                 1300
         LA    NEWOLD,1                                                1310
M1       EQU   *                                                       1320
*        HOLEN VON BLKSIZE AUS JFCB.                                    1330
         L     Q,JFCB+100                                              1340
         SLL   Q,16                                                    1350
         SRL   Q,16                                                    1360
         ST    Q,BLKSZ                                                 1370
         LTR   Q,Q                                                     1380
         BNP   NOBLKS                                                  1390
M2       EQU   *                                                       1400
*        AUF VIELFACHE VON 4 AUFRUNDEN.                                 1410
         LR    I,Q                                                     1420
         SRL   Q,2                                                     1430
         SLL   Q,2                                                     1440
         CR    I,Q                                                     1450
         BE    M3                                                      1460
         LA    Q,4(Q)                                                  1470
M3       EQU   *                                                       1480
*        BUFFER ALIGNMENT PRUEFEN. GEGEBENENFALLS Q AUF VIELFACHE       1490
*        VON 8 AUFRUNDEN.                                               1500
         TM    JFCB+92,X'01'                                           1510
         BO    M4                                                      1520
         LR    I,Q                                                     1530
         SRL   Q,3                                                     1540
         SLL   Q,3                                                     1550
         CR    I,Q                                                     1560
         BE    M4                                                      1570
         LA    Q,8(Q)                                                  1580
M4       EQU   *                                                       1590
         LR    RBLKS,Q                                                 1600
         B     0(BRANCH2)                                              1610
*        WENN DATASET NEW, DANN SPRUNG NACH GTMN. SONST                 1620
*        OLD DATASET LABEL LESEN UND PRUEFEN OB BLKSIZE UEBER=          1630
*        EINSTIMMT.                                                     1640
M6       LTR   NEWOLD,NEWOLD                                           1650
```

```
         BZ    GTMN                                                    1660
*        LABEL VORHANDEN?                                              1670
         TM    JFCB+66,X'11'                                           1680
         BZ    M5                                                      1690
         B     GTMN                                                    1700
M5       EQU   *                                                       1710
*        DIRECT ACCESS DEVICE OR MAGNETIC TAPE?                        1720
         L     Q,16(DDLAUF)          ADRESSE VON UCB.                1730
         TM    18(Q),X'20'      DIRECT ACCESS DEVICE?                1740
         BO    M8                                                      1750
         TM    18(Q),X'80'      MAGNETIC TAPE?                       1760
         BO    M9                                                      1770
         B     FEHLER7                                                 1780
M9       EQU   *                                                       1790
*        MAGNETIC TAPE.                                                1800
         OPEN  DCBALL                                                  1810
         RDJFCB  DCBALL                                                1820
*        RDJFCB ERFOLGREICH?                                           1830
         LTR   15,15                                                   1840
         BNZ   FEHLER7                                                 1850
         CLOSE DCBALL                                                  1860
         L     Q,JFCB+100      BLKSIZE                                 1870
         B     M13                                                     1880
M8       EQU   *                                                       1890
*        LESEN VON FORMAT 1 DSCB. ZUERST VOL-SER-NUMBER AUS            1900
*        UCB NACH VOLSER SPEICHERN. Q ENTHAELT ADRESSE VON UCB.        1910
         MVC   VOLSER(6),28(Q)  VOL-SER-NUMBER MOVEN.                  1920
         OBTAIN  DSCB                                                  1930
*        OBTAIN ERFOLGREICH?                                           1940
         LTR   15,15                                                   1950
         BNZ   FEHLER7                                                 1960
         L     Q,WKA+40                                                1970
M13      SLL   Q,16                                                    1980
         SRL   Q,16                                                    1990
         B     0(BRANCH1)                                              2000
M7       C     Q,BLKSZ                                                 2010
         BNE   MSG1                                                    2020
         B     GTMN                                                    2030
NOBLKS   EQU   *                                                       2040
*        WENN DATASET OLD UND LABEL VORHANDEN BRANCH1 MIT M2 LADEN     2050
*        BRANCH2 MIT GTMN LADEN UND SPRUNG NACH M5.                    2060
*        WENN DATASET OLD UND KEIN LABEL VORHANDEN SPRUNG NACH         2070
*        MSG4.                                                         2080
*        WENN DATASET NEW PRUEFEN OB DDNAME GLEICH FT07F001.           2090
*        WENN JA BLKSIZE GLEICH 80 SETZEN. SPRUNG NACH GTMN.           2100
*        WENN DATASET NEW UND DDNAME GLEICH FT05F001 BLKSIZE GLEICH 80 2110
*        SETZEN, WENN DDNAME GLEICH FT07F001 BLKSIZE GLEICH 936 SETZEN,2120
*        SONST BLKSIZE GLEICH 800 SETZEN. DANACH SPRUNG NACH GTMN.     2130
*        BENUTZER WIRD BENACHRICHTIGT VOM EINSETZEN DER DEFAULT-WERTE. 2140
         LA    BRANCH1,M2                                              2150
         LA    BRANCH2,GTMN                                            2160
*        LABEL VORHANDEN?                                              2170
         TM    JFCB+66,X'11'                                           2180
         BZ    M11                                                     2190
         LTR   NEWOLD,NEWOLD                                           2200
         BP    MSG4                                                    2210
         B     M12                                                     2220
M11      LTR   NEWOLD,NEWOLD                                           2230
         BP    M5                                                      2240
M12      CD    0,FT07F001                                              2250
         BE    SET80                                                   2260
         CD    0,FT05F001                                              2270
         BE    SET80                                                   2280
         CD    0,FT06F001                                              2290
         BE    SET936                                                  2300
         B     SET800                                                  2310
SET80    LA    RBLKS,80                                                2320
         B     NAC                                                     2330
SET936   LA    RBLKS,936                                               2340
         B     NAC                                                     2350
SET800   LA    RBLKS,800                                               2360
NAC      EQU   *                                                       2370
*        DDNAME IN NACHRICHT MOVEN. BLKSIZE IN NACHRICHT MOVEN.        2380
         MVC   WTO5+51(8),DDNAM                                        2390
         LTR   RBLKS,RBLKS                                             2400
         BM    FEHLER7                                                 2410
         C     RBLKS,=F'9999'                                          2420
         BH    FEHLER7                                                 2430
         CVD   RBLKS,DHILF                                             2440
         LA    15,HIER                                                 2450
STEZA    SR    I,I                                                     2460
         L     J,DHILF+4                                               2470
         SRL   J,4                                                     2480
STEZA1   LA    I,1(I)                                                  2490
         SRL   J,4                                                     2500
         LTR   J,J                                                     2510
         BP    STEZA1                                                  2520
         UNPK  DHILF(5),DHILF+5(3)                                     2530
         MVZ   DHILF+4(1),DHILF+3                                      2540
         BR    15                                                      2550
HIER     C     I,=F'4'                                                 2560
         BE    MOV4                                                    2570
         C     I,=F'3'                                                 2580
         BE    MOV3                                                    2590
         C     I,=F'2'                                                 2600
         BE    MOV2                                                    2610
         C     I,=F'1'                                                 2620
         BNE   FEHLER7                                                 2630
MOV1     MVC   WTO5+72(1),DHILF+4                                      2640
         B     WTO5                                                    2650
MOV2     MVC   WTO5+71(2),DHILF+3                                      2660
         B     WTO5                                                    2670
MOV3     MVC   WTO5+70(3),DHILF+2                                      2680
         B     WTO5                                                    2690
MOV4     MVC   WTO5+69(4),DHILF+1                                      2700
WTO5     WTO   '      MISSING BLKSIZE IN DDCARD WITH DDNAME=       . BL$ 2710
               KSIZE       ASSUMED.',ROUTCDE=11                       2720
GTMN     EQU   *                                                       2730
*        RBLKS ENTHAELT BLKSIZE. ES WIRD ANGENOMMEN, DASS              2740
*        2 PUFFER UND EIN KONTROLLBLOCK MIT 8 BYTE ANGELEGT            2750
```

```
*        WERDEN SOLLEN.                                              2760
*        WENN BUFL GROESSER ALS BLKSIZE WIRD BUFL GENOMMEN.          2770
         L     Q,JFCB+88                                             2780
         SLL   Q,16                                                  2790
         SRL   Q,16                                                  2800
         CR    RBLKS,Q                                               2810
         BNL   TAKEBL                                                2820
         LR    I,Q                                                   2830
         TM    JFCB+92,X'01'                                         2840
         BO    ALIGN4                                                2850
         SRL   Q,3                                                   2860
         SLL   Q,3                                                   2870
         CR    I,Q                                                   2880
         BE    SETBLK                                                2890
         LA    Q,8(Q)                                                2900
         B     SETBLK                                                2910
ALIGN4   SRL   Q,2                                                   2920
         SLL   Q,2                                                   2930
         CR    I,Q                                                   2940
         BE    SETBLK                                                2950
         LA    Q,4(Q)                                                2960
SETBLK   LR    RBLKS,Q                                               2970
TAKEBL   AR    RBLKS,RBLKS                                           2980
         LA    RBLKS,8(RBLKS)                                        2990
         A     RBLKS,=XL4'80000000'                                  3000
         ST    RBLKS,PUFFER                                          3010
         GETMAIN  LC,LA=PUFFER,A=ADRESS                              3020
         LTR   15,15                                                 3030
         BZ    M10                                                   3040
*        PUFFER-ERVGM ZU MISREG ADDIEREN.                            3050
         L     Q,PUFFER                                              3060
         LA    Q,0(Q)                                                3070
         S     Q,ERVGM                                               3080
         BM    FEHLER7                                               3090
         A     Q,MISREG                                              3100
         ST    Q,MISREG                                              3110
         B     ERHOEHE                                               3120
M10      EQU   *                                                     3130
*        FREIE REGION NACH GETMAIN FESTSTELLEN.                      3140
         LA    1,PARM2                                               3150
         L     15,=V(FREESP)                                         3160
         BALR  14,15                                                 3170
*        DIFFERENZ ZWISCHEN VORHER UND NACHHER IN FREIER             3180
*        REGION BESTIMMEN.                                           3190
         L     Q,ERVGM                                               3200
         S     Q,ERNGM                                               3210
         BM    FEHLER7                                               3220
         AR    REGION,Q                                              3230
         FREEMAIN  L,LA=PUFFER,A=ADRESS                              3240
ERHOEHE  AR    DDLAUF,DDINCR                                         3250
         B     WEITER                                                3260
ADD2K    LA    REGION,2(REGION)                                      3270
         B     ERHOEHE                                               3280
RETURN   EQU   *                                                     3290
         L     Q,MISREG                                              3300
         LTR   Q,Q                                                   3310
         BNZ   MSG3                                                  3320
         TM    ERROR,X'01'                                           3330
         BO    FEHLER8                                               3340
         L     Q,PARMADR                                             3350
         ST    REGION,0(Q)                                           3360
         L     13,4(13)                                              3370
         RETURN  (14,12),T,RC=0                                      3380
RETURN1  L     Q,PARMADR                                             3390
         ST    0,0(Q)                                                3400
         L     13,4(13)                                              3410
         RETURN  (14,12),T,RC=4                                      3420
FEHLER2  L     0,=F'-2'                                              3430
         B     RETURN1                                               3440
FEHLER5  L     0,=F'-5'                                              3450
         B     RETURN1                                               3460
FEHLER7  L     0,=F'-7'                                              3470
         B     RETURN1                                               3480
FEHLER8  L     0,=F'-8'                                              3490
         B     RETURN1                                               3500
MSG1     EQU   *                                                     3510
         LA    I,MSG10                                               3520
         B     MSGW                                                  3530
MSG3     EQU   *                                                     3540
*        REGION MUSS UM MINDESTENS Q BYTE ERHOEHT WERDEN.            3550
         B     FEHLER8                                               3560
MSG4     EQU   *                                                     3570
         LA    I,MSG40                                               3580
         B     MSGW                                                  3590
MSGW     EQU   *                                                     3600
*        DDNAMEN IN MESSAGE MOVEN.                                   3610
         MVC   WTO1+41(8),DDNAM                                      3620
WTO1     WTO   '      ERROR IN DDCARD WITH DDNAME=         .',ROUTCDE=11  3630
         B     0(I)                                                  3640
MSG10    EQU   *                                                     3650
*        BLKSIZE AUS DD-KARTE UNGLEICH BLKSIZE AUS                   3660
*        OLD-DATASET-LABEL.                                          3670
*        Q ENTHAELT BLKSIZE AUS OLD DATASET LABEL                    3680
         LTR   Q,Q                                                   3690
         BM    FEHLER7                                               3700
         C     Q,=XL4'0000FFFF'                                      3710
         BH    FEHLER7                                               3720
         CVD   Q,DHILF                                               3730
         LA    15,DA                                                 3740
         B     STEZA                                                 3750
DA       EQU   *                                                     3760
         C     I,=F'5'                                               3770
         BE    MOVDA5                                                3780
         C     I,=F'4'                                               3790
         BE    MOVDA4                                                3800
         C     I,=F'3'                                               3810
         BE    MOVDA3                                                3820
         C     I,=F'2'                                               3830
         BE    MOVDA2                                                3840
         C     I,=F'1'                                               3850
```

```
         BE    MOVDA1                                                   3860
         B     FEHLER7                                                  3870
MOVDA1   MVC   WTO4+55(1),DHILF+4                                      3880
         B     WTO4                                                     3890
MOVDA2   MVC   WTO4+54(2),DHILF+3                                      3900
         B     WTO4                                                     3910
MOVDA3   MVC   WTO4+53(3),DHILF+2                                      3920
         B     WTO4                                                     3930
MOVDA4   MVC   WTO4+52(4),DHILF+1                                      3940
         B     WTO4                                                     3950
MOVDA5   MVC   WTO4+51(5),DHILF                                        3960
WTO4     WTO   '     BLKSIZE OF THE ( OLD ) DATASET EQUALS       BYTE.'$ 3970
               ,ROUTCDE=11                                              3980
         B     MSG                                                      3990
MSG40    EQU   *                                                        4000
*        DATASET IST OLD. KEIN LABEL VORHANDEN. KEINE BLKSIZE           4010
*        IST AUF DER DDKARTE ANGEGEBEN.                                 4020
WTO2     WTO   '     MISSING BLKSIZE ON DDCARD FOR OLD DATASET WITH NO $ 4030
               LABEL.',ROUTCDE=11                                       4040
         B     MSG                                                      4050
MSG      MVI   ERROR,X'01'                                              4060
         B     ERHOEHE                                                  4070
DHILF    DC    D'0'                                                     4080
FTAUS    DC    C'FT00F001'                                              4090
SYSAB    DC    C'SYSABEND'                                              4100
SYSUD    DC    C'SYSUDUMP'                                              4110
FT05F001 DC    C'FT05F001'                                              4120
FT06F001 DC    C'FT06F001'                                              4130
FT07F001 DC    C'FT07F001'                                              4140
DDNAM    DC    D'0'                                                     4150
TIOTADR  DS    F                                                        4160
PUFFER   DC    F'0'                                                     4170
ERROR    DC    F'0'                                                     4180
BLKSZ    DC    F'0'                                                     4190
EXL      DC    XL1'87'                                                  4200
         DC    AL3(JFCB)                                                4210
ADRESS   DC    F'0'                                                     4220
MISREG   DC    F'0'                                                     4230
FRVGM    DC    F'0'                                                     4240
FRNGM    DC    F'0'                                                     4250
PARMADR  DC    F'0'                                                     4260
PARM1    DC    XL1'80'                                                  4270
         DC    AL3(FRVGM)                                               4280
PARM2    DC    XL1'80'                                                  4290
         DC    AL3(FRNGM)                                               4300
DCBALL   DCB   DSORG=PS,MACRF=R,EXLST=EXL                              4310
JFCB     DC    44F'0'                                                   4320
DSCB     CAMLST  SEARCH,JFCB,VOLSER,WKA                                 4330
VOLSER   DC    CL8'00000000'                                            4340
WKA      DS    0D                                                       4350
         DS    148C                                                     4360
SAVE     DS    18F                                                      4370
         END                                                            4380
```

This page has been left internally blank.

Section 3

Production of KEMA-input from BRIGITTE-output by the program COPEND

I. Langner, R. Meyer[+]

Revised by
E. Stein

[+]Present address: Software AG, Darmstadt

Survey

To insert the translated (see BRIGITTE section 2.) ENDF/B-data into the local public KEDAK-library, the program COPEND is used. This program produces for one or several materials KEMA input (see II) out of the translated ENDF/B-data, which are stored in a KEDAK-library created by BRIGITTE (see 2.2.4). To control the program only the material name (resp. names) is (resp. are) necessary. All datatypes available for a specified material are processed. The output of the program COPEND can be read by KEMA (see II) without performing any changes.

The following description is in German, because the program has been first described by the author[+] in an internal report, from which the text was taken.

For the user of the program we will give some helps in English. The input and the JCL-cards (Job Control Cards) are described in 3.2.2 and 3.2.3. The JCL-cards are selfexplaining (for an IBM-user). The input consists of the number of materials, which are to be copied, followed by the material names (see 3.2.3 after //G.SYSIN DD *). The rest of this section describes the subroutines of COPEND, which are also listed as source programs.

[+] The author has left the KFK.

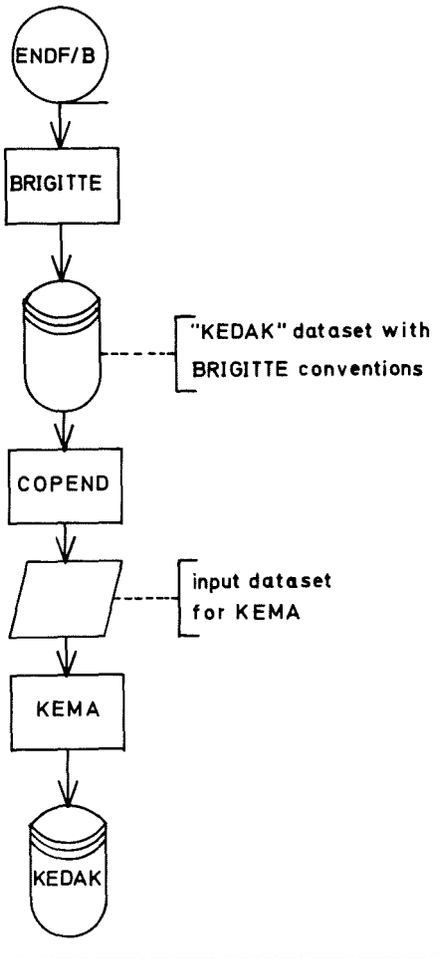
Inhalt

3.1 Einleitung

Das in Abschnitt 2. beschriebene Programm BRIGITTE überträgt Daten aus der ENDF-Darstellung in die KEDAK-Darstellung.

BRIGITTE schreibt diese Daten nicht direkt in die Kerndatenbibliothek KEDAK, sondern im FORMAT+ der KEDAK Bibliothek auf eine temporäre Datei, um die Korrektur möglicher Fehler in den konvertierten Daten zu erleichtern.

Aus diesem Grund ist ein Anschlußprogramm erforderlich, das es gestattet, diese konvertierten Datensätze in die offizielle KEDAK-Bibliothek zu übernehmen. Das hierfür geschriebene Programm COPEND ist im folgenden beschrieben. Es erstellt aus den konvertierten Datensätzen eine Datei im Eingabeformat für das Verwaltungsprogramm der Kerndatenbibliothek KEMA (siehe II oder /2/), mit dessen Hilfe die Daten schließlich in KEDAK eingebracht werden. Nachfolgendes Flußdiagramm soll diesen Ablauf veranschaulichen.



+Dieses Format unterscheidet sich geringfügig von dem "KEMA" KEDAK-Format (siehe 2.2.4).

3.2 Das Programm COPEND

3.2.1 Die Funktion des Programmes COPEND

COPEND ist ein Programm, welches ENDF/B-Daten /1/ aus einer von BRIGITTE (siehe 2.) erstellten Datei in eine Datei im Eingabeformat für KEMA (siehe II oder /2/) umwandelt.

Die Datei der konvertierten ENDF/B-Daten wird auf Einheit Nr. 1 (≙ data set reference number = 1) erwartet, die Datei im KEMA-Eingabeformat wird auf die Einheit Nr. 10 geschrieben. Die Eingabe für COPEND ist formatfrei, es gelten die FREEFO-Konventionen /3/. Für FREEFO muß die Einheit Nr. 8 bereitgestellt werden.

Das Programm COPEND benutzt das Retrieval Paket LDFPAC (siehe III.2) und die Subroutine FREEFO /3/.

3.2.2 Beschreibung der Eingabe

NMAT	≤ 30, Anzahl der von der Datei der ENDF/B-Daten zu kopierenden Isotope (Materialien)
NATNAM(I) (I=1,NMAT)	Doppelworte, die Materialnamen in KEDAK-Konvention (siehe II, III.1, /2/) enthalten.

3.2.3 Die Jobkontrollkarten

Liste der Jobkontrollkarten mit einem Eingabebeispiel:

```
//INR048PA JOB (0048,101,P6M1A),LANGNER,CLASS=A
// EXEC FGG,LIB=NUSYS,NAME=COPEND
//G.FT08F001 DD UNIT=SYSDA,SPACE=(TRK,1)
//G.FT01F001 DD UNIT=2314,VOL=SER=GFK050,DSN=KEDAK3,DISP=SHR
//G.FT10F001 DD UNIT=SYSDA,SPACE=(TRK,20)
//G.SYSIN DD +
2   'AL 27    '  'C  12    '
/*
```

Das Programm COPEND befindet sich in der Bibliothek LOAD.NUSYS; es ist 74 k Bytes lang

3.3 Die einzelnen Subroutinen von COPEND

3.3.1 Die Subroutine INPDAT

3.3.1.1 Die Organisation der ENDF/B-Daten für WRIREC

Die Subroutine INPDAT wird für jedes gewünschte Isotop einmal aufgerufen. Mit Hilfe der Subroutine LDFITN (siehe III.2) wird festgestellt, welche Reaktionstypen auf dem von BRIGITTE erzeugten KEDAK-File vorhanden sind; die Namen dieser Typen werden in dem COMMON /TYPE/ im Feld TYPES abgelegt.

Die Daten für jeden vorhandenen Reaktionstyp werden von den Subroutinen LOCDAT (s. 3.3.3) und DAT (s. 3.3.4) in das Feld X gebracht, und von der Subroutine WRIREC (s. 3.4.2) auf die Einheit Nr. 10 geschrieben.

3.3.1.2 Die Argumentliste der Subroutine INPDAT

Der Aufruf der Subroutine INPDAT lautet:
CALL INPDAT(MATNA,NTYP,MAXNUM,NTAPE)

MATNA	- Doppelwort, Isotopname (KEDAK-Konvention)
NTYP	- Anzahl der Reaktionstypen in der Liste ((s. 3.2.2), z.Z. = 40)
MAXNUM	- maximale Anzahl der Daten, die für einen Eingabesatz für KEMA geliefert werden dürfen
NTAPE	- Nummer der Einheit, auf welche die umgewandelten Daten geschrieben werden sollen

3.3.2 Die Subroutine WRIREC

3.3.2.1 Schreiben eines Satzes im KEMA-Eingabeformat

Die Subroutine WRIREC schreibt jeweils einen Satz im KEMA-Eingabeformat auf die Einheit NTAPE und berücksichtigt dabei, daß die Anzahl der Worte in diesem Satz nicht größer als 2000 sein darf. Außerdem werden diese Daten über die Ausgabeeinheit Nr. 6 ausgedruckt.

Das Herausschreiben auf NTAPE geschieht über die Subroutine WRIII (N,Z,NTAPE). Dabei ist N $\leq$ 2000 die Anzahl der Worte im Satz, Z(N) das Feld der Daten des Satzes, NTAPE die Nummer der Einheit,auf die geschrieben wird.

3.3.2.2 Die Argumentliste der Subroutine WRIREC

Der Aufruf für WRIREC lautet:
CALL WRIREC(NARG,NAMES,X,NUMX,NTAPE)

NARG	Feld von 3 Worten, in NARG(1) steht die Anzahl der Namen für die gewünschte Namenskombination (vgl. 3.3.3.2)
NAMES	Doppelwortfeld für die Namen (KEDAK-Konvention) (vgl. 3.3.3.2)
X	Feld für die Daten (Dimension $\leq$ 2000)
NUMX	Anzahl der gelieferten Daten
NTAPE	die Nummer der Einheit, auf die geschrieben werden soll.

3.3.2.3 Der Aufbau des herausgeschriebenen Satzes

Die von WRIREC geschriebenen Sätze sind ADD-Sätze für KEMA (II, /2/). Sie haben den folgenden Aufbau:

N	Anzahl der Worte im Satz
ADD	= 'ADD ', Doppelwort
NNAM	Anzahl der Namen in der Namenskombination
NAMES(I) (I=1,NNAM)	Namen (KEDAK-Konvention)
NARG	Anzahl der Argumente
NWERT	Anzahl der Werte
X(I) (I=1,NUMX)	Feld der Daten

Anmerkung: Für NARG = 0, wird als formales Argument der Wert 'ARG ' gesetzt.

3.3.3 Die Subroutine LOCDAT mit Entry NXTDAT

3.3.3.1 Die Funktion der Subroutine LOCDAT

Die Subroutine LOCDAT mit Entry NXTDAT liest Daten für einen beliebigen Reaktionstyp von einer Datei im KEDAK-Format in ein Feld X.

LOCDAT wird für den ersten Aufruf einer Namenskombination benutzt. Für eventuell nötige weitere Aufrufe für die gleiche Namenskombination muß das Entry NXTDAT benutzt werden. LOCDAT verwendet die Subroutinen LDFLOC und LDFNXT (siehe III.2).

Das Einlesen wird abgebrochen, wenn das Feld X soweit gefüllt ist, daß ein weiterer logischer Satz der KEDAK-Bibliothek, bestehend aus (NARG + NWERT) Daten, keinen Platz darin findet (vgl. 3.3.2.3).

3.3.3.2 Argumentliste der Subroutine LOCDAT

Der Aufruf lautet:
CALL LOCDAT(NARG,NAMES,NAMNXT,X,NUMX,MAXNUM,NR)

NARG	Feld der Länge 3. NARG(1) enthält die Anzahl der Namen des gewünschten Datentypes, NARG(2) und NARG(3) werden mit der Anzahl der Argumente und Werte des Datentyps gefüllt.
NAMES	Feld von Doppelworten. Muß vor dem Aufruf mit den alphanumerischen Namen des gewünschten Datentyps gefüllt werden
NAMNXT	falls NR = 2, enthält NAMNXT die Namen des als nächsten zu lesenden Datentyps
X	Feld, das mit den gelesenen Daten gefüllt wird.
NUMX	Anzahl der gelesenen Datenworte
MAXNUM	Länge des Feldes X

NR Returncode

= 0 für die angegebenen Namen gibt es noch weitere Daten, die im Feld X keinen Platz mehr fanden; zum Lesen dieser Daten ist NXTDAT aufzurufen

= 2 der gewünschte Datentyp hat auch numerische Namen, für die angegebenen numerischen gibt es keine weiteren Daten, aber NAMNXT enthält die numerischen Namen der als nächstes zu lesenden Daten; hierfür ist LOCDAT mit den geänderten Namen zu benutzen

= 3 Datentyp hat keine numerischen Namen, alle Daten für den gewünschten Datentyp wurden übertragen

Der Returncode wird von der Subroutine DAT (vgl. 3.3.4) interpretiert und verändert.

3.3.3.3 Argumentliste des Entry NXTDAT

Der Aufruf für NXTDAT lautet:
CALL NXTDAT(NUMX,NR)

Die Argumente haben die gleiche Bedeutung wie in 3.3.3.2 beschrieben. Das Entry NXTDAT ist immer dann aufzurufen, wenn der vorher von LOCDAT oder NXTDAT zurückgegebene Returncode NR = 0 war.

3.3.4 Die Subroutine DAT und Entry REPDAT

3.3.4.1 Aufgabe der Subroutine DAT

Unter Verwendung von LOCDAT und NXTDAT stellen DAT und REPDAT die Daten für einen im Aufruf spezifizierten Datentyp in einem im Aufruf angegebenen Datenfeld zur Verfügung.

Für jeden Datentyp ist zunächst ein Aufruf von DAT erforderlich. Um die Daten des Datentyps vollständig zu lesen, ist REPDAT solange aufzurufen, wie der zuletzt zurückgegebne Returncode ≤ 1 ist (s.u.).

Falls für einen Datentyp numerische Namen existieren, sind diese beim ersten Aufruf (von DAT) mit Null zu füllen. Die tatsächlichen numerischen Namen werden von DAT und REPDAT eingefüllt.

3.3.4.2 Die Argumentliste der Routine DAT

Der Aufruf von DAT lautet:
CALL DAT(NARG,NAMES,X,NUMX,MAXNUM,NR)

NR Returncode

= 0 für die spezifizierte Kombination von alphanumerischen Namen gibt es noch weitere Daten, die im Feld X keinen Platz mehr fanden; zum Lesen dieser Daten ist REPDAT aufzurufen.

= 1 für die angegebene Kombination von alphanumerischen Namen sind alle Daten in das Feld X übertragen worden; für diesen Datentyp gibt es jedoch noch weitere Kombinationen von numerischen Namen; um diese numerischen Namen und die Daten hierfür zu erhalten, ist REPDAT aufzurufen

= 2,3 alle Daten des gewünschten Datentyps wurden in das Feld X übertragen.

Bemerkung: Die anderen obigen Argumente sind in 3.3.3.2 beschrieben.

3.3.4.3 Die Argumentliste der Routine REPDAT

Der Aufruf lautet:
CALL REPDAT(NUMX,NR)

Bezüglich der Interpretation der Argumentse s.o. REPDAT wird immer dann aufgerufen, wenn der Returncode des vorhergehenden Aufrufes von DAT oder REPDAT $\leq$ 1 war.

Literaturhinweise

/1/ M.K. Drake, Data Formats and Procedures for the ENDF Neutron Cross Section Library, National Neutron Cross Section Center, BNL 50274 (T-601), 1970

/2/ B. Krieg, Handling and Service Programs for the Karlsruhe Nuclear Data File KEDAK. Part I: Management and Retrieval Programs. Institut für Neutronenphysik und Reaktortechnik, Gesellschaft für Kernforschung mbH., Karlsruhe, KFK 1725, Juni 1973

3.5 Quellprogrammlisten

```
C
C
C                          PROGRAM COPEND
C      THIS PROGRAM WRITES ENDF/B DATA IN KEMA-INPUT-FORMAT ON A DATA SET
C      WITH THE REFERENCE NUMBER = 10
C
       COMMON /TYPE/ TYPES
       REAL*8 NAMES(5),TYPES(70),MATNAM(30),MATNA,ENDE
       DIMENSION F(100),NF(100),NP(70)
       EQUIVALENCE (F(1),NF(1))
       DATA ENDE/8HENDE     /
C
       KNDF=1
       NIN=5
       NOUT=6
       INP=8
       NTAPE=10
C
       MAXNUM=1970
       NT=70
C
       CALL LDFOPN (KNDF,NADAT,&1000)
C
       CALL FREEFO(NIN,INP,NOUT,F,F,F)
C
       READ (INP) NMAT,(MATNAM(I),I=1,NMAT)
       DO 10 IMAT=1,NMAT
       MATNA=MATNAM(IMAT)
C
       CALL LDFITN(NR,MATNA,TYPES,NP,NTYP,NT,2)
       IF(NR.EQ.0) GO TO 11
       CALL SHELL(TYPES,NTYP,1,NTYP)
C
       CALL INPDAT(MATNA,NTYP,MAXNUM,NTAPE,&1000)
       GO TO 10
C
   11  IF(NTYP.GT.70) GO TO 12
       WRITE (NOUT,101) MATNA
  101  FORMAT(1X,'FUER ',A8,' SIND KEINE DATENTYPEN VORHANDEN')
       GO TO 10
   12  WRITE (NOUT,102) MATNA,NT
  102  FORMAT (1X,'FUER ',A8,' SIND ',I4,' (MAX=70) DATENTYPEN VORHANDEN'
      1)
   10  CONTINUE
C
       N=2
       WRITE (NTAPE) N,ENDE
 1000  STOP
       END
```

```
C
C
      SUBROUTINE INPDAT(MATNA,NTYP,MAXNUM,NTAPE,*)
C
      COMMON /TYPE/ TYPES
      DIMENSION           NARG(3),X(1990)
      REAL*8 TYPES(70),MATNA,NAMES(10),BLANK
      DATA BLANK/8H        /
C
      NOUT=6
      NARG(1)=2
C
      DO 10 ITYP=1,NTYP
      IN=0
      NAMES(1)=MATNA
      NAMES(2)=TYPES(ITYP)
C
      CALL DAT (NARG,NAMES,X,NUMX,MAXNUM,NR)
C
      NZ=NARG(1)
      IF(NR-3)11,12,13
   12 WRITE (NOUT,101)(NAMES(L),L=1,NZ)
  101 FORMAT(1X,'NO DATA FOR THIS NAME COMBINATION - ',4A8)
      GO TO 10
C
   15 WRITE (NOUT,102) (NAMES(L),L=1,NZ)
  102 FORMAT(1X,'DIMENSION  OF THE WORKAREA FOR LDFLCC OR LDFNXT TO SMAL
     1L, THE REACTION TYPE IS ',4A8)
      GO TO 10
C
   16 CALL REPDAT(NUMX,NR)
      IF(NR-3)11,12,13
C
   11 CALL WRIREC  (NARG,NAMES,X,NUMX,MAXNUM,NTAPE)
      IF (NR.EQ.1) GO TO 17
      IF (NR.EQ.0) GO TO 16
      IF (NR.EQ.2) GO TO 20
      WRITE (NOUT,103) NR,(NAMES(L),L=1,NZ)
  103 FORMAT(1X,'RETURN-CODE = ',I3,'THE REACTION TYPE = ',4A8)
      RETURN 1
C
   13 IF(NR.EQ.4) GO TO 15
   17 IN=IN+1
      GO TO 16
C
   20 IF(NZ.GT.2) IN=IN+1
      WRITE (NOUT,104) (NAMES(L),L=1,2),IN
  104 FORMAT (1X,'THE REACTION TYPE ',A8,1X,A8,' HAS ',I5,' DIFFERENT CO
     1MBINATIONS OF SECONDARY NAMES')
   10 CONTINUE
      RETURN
      END
```

```
C
C
      SUBROUTINE WRIREC (NARG,NAMES,X,NUMX,MAXNUM,NTAPE)
C
C     WRITREC WRITES A RECORD IN KEMA-INPUT-FORMAT ON A DATA SET WITH
C     THE REFERENCE NUMBER NTAPE, THE NUMBER OF FULLWORDS IN THE RECORD
C     MAY BE LESS THAN 2000
C
C
      REAL*8 NAMES(1),ADD,ZD(2)
      DIMENSION NARG(1),X(1)
      DIMENSION Z(2100),ZX(4),IZ(2100)
      DATA ADD/8HADD     /,ARG/4HARG /
      EQUIVALENCE (Z(1),ADD),(Z(3),NNAM),(ZX(1),ZD(1)),(Z(4),ZX(1)),
     1(Z(1),IZ(1))
C
      NOUT=6
      NNAM=NARG(1)
      NARG2=NARG(2)
      NWERT=NARG(3)
      NNAM2=NNAM-2
      ZD   (1)=NAMES(1)
      ZD   (2)=NAMES(2)
C
      IF(NUMX-MAXNUM)14,14,15
   15 WRITE (NOUT,112) NUMX
  112 FORMAT(1X,'THE NUMBER OF DATA =',I6,'IS GREATER THAN ALLOWED')
      GO TO 3333
C
   14 IF(NNAM2)11,12,13
   11 WRITE (NOUT,111) NAMES
  111 FORMAT(1X,'THE NUMBER OF NAMES LESS THAN 2',4A8)
      GO TO 3333
C
C     NUMBER OF NAMES  EQUAL 2
C
   12 N=NUMX+9
      IZ(8)=NARG2
      IZ(9)=NWERT
      I1=9
      ASSIGN 19 TO I2
      IF (NARG2.EQ.0) GO TO 16
C
   18 DO 20 I=1,NUMX
   20 Z(I1+I)=X(I)
      GO TO I2,(19,21,22,23,25)
   16 I1=10
      N=N+1
      Z(I1)=ARG
      ASSIGN 21 TO I2
      GO TO 18
C
C     THE NUMBER OF NAMES GREATER THAN 2
C
   13 N=NUMX+9+NNAM2
```

```
      DO 10 INAM=1,NNAM2
   10 Z(INAM+7)=NAMES(2+INAM)
      I3=8+NNAM2
      IZ(I3)=NARG2
      IZ(I3+1)=NWERT
      I1=I3+1
      IF(NARG2.EQ.0) GO TO 24
      ASSIGN 22 TO I2
      IF(NNAM2.EQ.1) ASSIGN 25 TO I2
      GO TO 18
C
   24 N=N+1
      I1=I1+1
      Z(I1)=ARG
      ASSIGN 23 TO I2
      GO TO 18
C
   19 WRITE (NOUT,1) N,(Z(I),I=1,N)
      GO TO 17
   21 WRITE (NOUT,3) N,(Z(I),I=1,N)
      GO TO 17
   22 WRITE (NOUT,2) N,(Z(I),I=1,N)
      GO TO 17
   23 WRITE (NOUT,4) N,(Z(I),I=1,N)
      GO TO 17
   25 WRITE (NOUT,5) N,(Z(I),I=1,N)
    1 FORMAT(1X,I5,1X,2A4,1X,I5,1X,2A4,1X,2A4,1X ,2I5/(1X,10E13.5))
    2 FORMAT(1X,I5,1X,2A4,1X,I5,1X,2A4,1X,2A4,1X ,2E15.6,2I5/(1X,10E13.5
     *))
    3 FORMAT(1X,I5,1X,2A4,1X,I5,1X,2A4,1X,2A4,1X,2I5,1X,A4/(1X,10E13.5))
    4 FORMAT(1X,I5,1X,2A4,1X,I5,1X,2A4,1X,2A4,1X,2E15.6,2I5,1X,A4/(1X,10
     *E13.5))
    5 FORMAT(1X,I5,1X,2A4,1X,I5,1X,2A4,1X,2A4,1X,E15.6,2I5/(1X,10E13.5))
C
   17 CALL WRIT1 (N,Z,NTAPE)
 3333 RETURN
      END




C
C
      SUBROUTINE WRIT1 (N,X,NTAPE)
      DIMENSION X(N)
      WRITE (NTAPE) N,X
      RETURN
      END
```

```
C
C
      SUBROUTINE DAT(NARG,NAMES,X,NUMX,MAXNUM,NR)
C
      REAL*8 NAMES(1),NAMNXT(10)
      DIMENSION X(1),NARG(1)
C
C         USE DAT,REPDAT FOR RETRIEVAL OF KEDAK DATA OF ANY TYPE,WHICH
C         SHOULD BE FILLED IN SUCCESSIVE ORDER INTO X.
C         USE DAT FOR FIRST CALL WITH THIS FIRST TWO NAMES.
C         USE REPDAT FOR FURTHER CALLS.
C         NR=1 MEANS,THAT THE LAST DATA FOR THIS NAME COMBINATION ARE
C            TRANSMITTED.THERE ARE HOWEVER FURTHER DATA WITH SAME
C            FIRST TWO NAMES.TO GET THESE AND THE NEW NAMES
C            CALL REPDAT.
C            REST OF RETURNCODE SETTING SEE IN LOCDAT.
C
      NAMZ=NARG(1)
      IF(NAMZ.LE.2) GOTO 10
      DO 5 I=3,NAMZ
    5 NAMES(I)=0.
   10 CALL LOCDAT(NARG,NAMES,NAMNXT,X,NUMX,MAXNUM,NR)
        NAMZ=NARG(1)
   12 IF(NAMZ.LE.2) GOTO 100
      IF(NR.NE.2) GOTO 100
      DO 20 I=1,NAMZ
      IF(NAMNXT(I).EQ.NAMES(I)) GOTO 20
      GOTO 30
   20 CONTINUE
      GOTO 100
   30 NR=1
      GOTO 100
C
      ENTRY REPDAT(NUMX,NR)
      IF(NR.NE.1) GOTO 50
      DO 40 I=1,NAMZ
   40 NAMES(I)=NAMNXT(I)
      GOTO 10
   50 CALL NXTDAT(NUMX,NR)
      GOTO 12
  100 RETURN
      END
C
C
      SUBROUTINE LOCDAT(NARG,NAMES,NAMNXT,X,NUMX,MAXNUM,NR)
C
      DIMENSION X(1),Z(20),NARG(1)
      REAL*8 NAMES(1),NAMNXT(1),NAMSV(10)
C
C         LOCDAT,NXTDAT MAY BE USED TO RETRIEVE DATA FOR ANY TYPE
C         FROM KEDAK.THE DATA ARE FILLED IN SUCCESSIVE ORDER INTO X.
```

```
C          LOCDAT MUST BE USED FOR FIRST CALL WITH THIS COMBINATION
C          OF NAMES,NXTDAT FOR FURTHER CALLS.
C          NR ..... RETURN CODE.
C          0 = THERE ARE FURTHER DATA FOR THIS COMBINATION OF NAMES.
C          2 = NO SUCH DATA
C          3 = NO DATA FOR THIS NAME COMBINATION.
C        4 = DIMENSION OF Z EXCEEDED.
C          READING STOPS WHEN LENGTH OF X(MAXNUM) WOULD BE EXCEEDED.
C          RETURN CODE IS SET 0.NUMX IS SET LENGTH OF DATA RETRIEVED.
C          IN CASE OF NR=2 NAMNXT CONTAINS THE NAMES OF THE NEXT TYPE.
C
      NAMZ=NARG(1)
      I=0
      CALL LDFLOC(NERR,NARG,NAMES,Z)
       NAMZ=NARG(1)
      IF(NERR.EQ.0) GOTO 30
      DO 10 L=1,NAMZ
   10 NAMSV(L)=NAMES(L)
      LL=NARG(2)+NARG(3)
      IF(LL.GT.20) GOTO 40
      GOTO 21
   20 CALL LDFNXT(NERR,NARG,NAMES,Z)
      IF(NERR.EQ.0) GOTO 25
   21 IE=I+LL
      IF(IE.GT.MAXNUM) GOTO 35
      DO 22 L=1,LL
      I=I+1
   22 X(I)=Z(L)
      GOTO 20
C
      ENTRY NXTDAT(NUMX,NR)
      I=0
      GOTO 21
C
   25 NR=2
      DO 26 L=1,NAMZ
      NAMNXT(L)=NAMES(L)
   26 NAMES(L)=NAMSV(L)
      GOTO 200
   30 NR=3
      GOTO 200
   35 NR=0
      GOTO 200
   40 NR=4
      GOTO 200
  200 NUMX=I
      RETURN
      END
```

```
      SUBROUTINE SHELL(LISTE,L,LGA,LGE)
C
C     SORTIEREN NACH DEM SHELL-VERFAHREN.
C
      DIMENSION LISTE(L)
      REAL*8 LISTE,HELP
      INTEGER S
C
      S=L/2
    1 LGZ=LGA
    2 LTZ=LGZ
    3 IF(LISTE(LTZ).GE.LISTE(LTZ+S)) GOTO 4
      HELP=LISTE(LTZ)
      LISTE(LTZ)=LISTE(LTZ+S)
      LISTE(LTZ+S)=HELP
      LTZ=LTZ-S
      IF(LTZ.GE.LGA) GOTO 3
    4 LGZ=LGZ+1
      IF((LGZ+S).LE.LGE) GOTO 2
      S=S/2
      IF(S.NE.0) GOTO 1
      RETURN
      END
```

Section 4

Translating from KEDAK into ENDF/B (Version 1) format by the program KTOE

G. C. Panini[+]

[+]C.N.E.N., C.D.C., Bologna

The only program, which is available for the automated translation from KEDAK to ENDF/B, is the program KTOE. A NEA abstract of this code, which is distributed by the NEA Computer Programme Library, will be given.

It is planned, either to extend this code in order to fulfill ENDF/B-IV conventions, and above all to refine the rough translation, or to develop own methods.

NEA COMPUTER PROGRAMME LIBRARY 12/72 ABSTRACT NO. NEA 342

1. NAME OR DESIGNATION OF PROGRAMME. KTOE

2. COMPUTER FOR WHICH THE PROGRAMME IS DESIGNED AND OTHERS UPON WHICH IT IS OPERABLE. COMPLETELY MACHINE INDEPENDENT

3. NATURE OF PHYSICAL PROBLEM SOLVED. THIS CODE PERFORMS A FULLY AUTOMATED TRANSLATION FROM KEDAK INTO ENDF/B (VERSION 1)FORMAT. OUTPUT IS ON TAPE IN DECIMAL CARD IMAGE FORMAT.

4. METHOD OF SOLUTION. BEFORE TRANSLATION EACH REACTION IS CROPPED IN ORDER TO REDUCE THE AMOUNT OF DATA POINTS OVER WHICH IT IS SPECIFIED. LINEAR-LINEAR INTERPOLATION RULE IS PRESERVED. ELASTIC ANGULAR DISTRIBUTIONS ARE GIVEN IN TABULATED FORM. μ_L AND ξ_{el} ARE DIRECTLY CALCULATED. THE RESONANCE DATA BOTH, RESOLVED AND UNRESOLVED, ARE ALSO TRANSLATED; THIS LEADS TO A REDUNDANCY OF INFORMATION SINCE THE CORRESPONDING TABULATED CROSS-SECTIONS ARE GIVEN TOO.

5. RESTRICTIONS ON THE COMPLEXITY OF THE PROBLEM. THE EXISTING RESTRICTIONS BOTH ON KEDAK AND ENDF/B HAVE BEEN APPLIED TO THE ARRAY SIZES USED EVERYWHERE IN THE CODE.

6. TYPICAL RUNNING TIME. 3 TO 15 MINUTES PER NUCLIDE ON THE IBM 7094.

7. UNUSUAL FEATURES OF THE PROGRAMME. FIVE SIGNIFICANT FIGURES ARE ALLOWED FOR A REAL ITEM IN THE FORMAT.
SECONDARY ENERGY DISTRIBUTION LAWS FOR THE (N,N) AND (N,2N) PROCESSES ARE GENERATED BY ASSUMING AN EVAPORATION MODEL WITH THE TEMPERATURE EXPRESSED AS A FUNCTION OF THE INCIDENT MASS.

8. RELATED AND AUXILIARY PROGRAMMES. DICTON SHOULD BE USED TO INSERT AN INDEX OF THE CONTENTS IN THE FILE.
CHECKER SHOULD BE USED TO TEST THE TRANSLATED FILE.

9. STATUS. IN USE.

10. REFERENCES. ENDF/B AND KEDAK DESCRIPTION REPORTS.

11. MACHINE REQUIREMENTS. 32K OF DIRECTLY ADDRESSABLE CORE LOCATIONS
2 TAPE UNITS AND 5 TEMPORARY TAPES WHICH CAN BE SIMULATED ON
DISK OR OTHER DRIVES.

12. PROGRAMMING LANGUAGE USED. FORTRAN IV

13. OPERATING SYSTEM OR MONITOR UNDER WHICH PROGRAMME IS EXECUTED.
NONE IN PARTICULAR.

14. ANY OTHER PROGRAMMING OR OPERATING INFORMATION OR RESTRICTIONS.
THE CODE CAN TRANSLATE ONE NUCLIDE EACH TIME.

15. NAME AND ESTABLISHMENT OF AUTHOR. GIAN CARLO PANINI, C.N.E.N.,
C.D.C., BOLOGNA.

16. MATERIAL AVAILABLE.

17. CATEGORY. M
KEYWORDS. KEDAK,ENDF/B