KERNFORSCHUNGSZENTRUM

KARLSRUHE

April 1977 KFK 2387/V

Institut für Neutronenphysik und Reaktortechnik
Projekt Schneller Brüter

The KEDAK Program Compendium
Part V
KEDAK Evaluation Aids

Compiled by: F. H. Fröhner
with contributions from
C. Broeders, I. Broeders, B. Goel, I. Langner,
R. Meyer, H. W. Wiese

GESELLSCHAFT
FÜR
KERNFORSCHUNG M.B.H.

KARLSRUHE

GESELLSCHAFT FÜR KERNFORSCHUNG M. B. H.

KARLSRUHE

KERNFORSCHUNGSZENTRUM KARLSRUHE

KFK 2387/V

Institut für Neutronenphysik und Reaktortechnik

Projekt Schneller Brüter

The KEDAK Program Compendium

Part V

KEDAK Evaluation Aids

Compiled by

F. H. Fröhner

with Contributions from

C. Broeders, I. Broeders, B. Goel, I. Langner,

R. Meyer*, H. W. Wiese

*Present address: Software AG, Darmstadt

Gesellschaft für Kernforschung mbH., Karlsruhe

Abstract

Part V of the KEDAK Program Compendium contains descriptions of the more important codes that are used in KEDAK evaluation work at Karlsruhe, in particular: check programs, programs performing arithmetic operations on whole data sets, programs for weighting and smoothing and programs for estimation of nuclear-model and cross section parameters.

Das KEDAK-Programm-Compendium

Teil V

Programmhilfsmittel für die KEDAK-Auswertearbeit

Zusammenfassung

Teil V des KEDAK-Programm-Compendiums enthält Beschreibungen der wichtigeren Programme, die bei der KEDAK-Auswertearbeit in Karlsruhe benutzt werden - speziell Prüfprogramme, Programme für arithmetische Operationen an ganzen Datensätzen, Programme zum Wichten und Glätten und Programme für die statistische Schätzung von Kernmodell- und Wirkungsquerschnitts-Parametern.

V. KEDAK Evaluation Aids

Contents:

1. GENERAL REMARKS

The KEDAK library of evaluated neutron data must be regularly revised, updated and expanded in response to the users' needs. A continuous evaluation effort is necessary to maintain the usefulness of the file. The following sections deal with computer programs that were developped or adapted in the course of this evaluation work. It should be pointed out that as in previous chapters the program descriptions are little more than a collection from various sources and differ considerably with respect to depth and detail. References are found after each program description.

Zum Druck eingereicht am 17.3.1977

2. CHECK PROGRAMS

The large size and rapid growth of modern neutron data files make it difficult to ensure formal correctness and consistency with the laws of physics in evaluated files. Efficient check programs are required to ease the burden on the compiler. Two such programs are being used in KEDAK avaluation word:

- PRTSTKED for angle-integrated cross sections and related data such as cross section ratios that depend on incident energy only, and

- SELPLO for differential elastic-scattering cross sections.

They are described in Sects. 2.1 and 2.2.

2.1 THE GENERAL-PURPOSE CHECK PROGRAM PRTSTKED

Contents:

2.1.1 Introduction

The KEDAK print and test program PRTSTKED, developped in 1972 by R. Meyer, is designed to print a large number of KEDAK reaction types and to perform consistency checks and physics test. Among the data that cannot be tested yet are individual and statistical resonance parameters, angular distributions and secondary-particle spectra.

2.1.2 Consistency checks and physics tests

As explained in the appended PRTSTKED input/output description the program is sectioned into a number of modules. The actual test operations are performed in the program modules ISOPAC, TSTBL1 and TSTBL2:

- ISOPAC tests nuclear properties (charge, mass, spin, neutron binding energy, nuclear radius etc.);

- TSTBL1 checks individual energy-dependent cross sections;

- TSTBL2 checks consistency between individual partial cross sections and derived quantities such as the total cross section or the capture-to-fission ratio α.

2.1.2.1 Tests of nuclear properties

The following tests are perfomed by ISOPAC:

- Is the isotope in question on file?

- Does the charge number Z satisfy the condition
$1 \leq Z \leq 104$?

- Is Z integer?

- In case both isotopic and elemental data are stored, is the isotopic Z equal to the elemental Z?

- Is the nucleon number A=N+Z compatible with the atomic weight?

- Is Z consistent with the chemical symbol?

- Is the combination Z,A inside the range of the nuclide chart?

- Does Z,A belong to an unstable isotope? If yes, is the isotope listed as component of an element?

- Is the atomic weight consistent with the nuclide chart?

- Is the ground state spin I integer for even A, half-integer for odd A? Is I=0 for even A?

- Is the condition $0 \leq I \leq 10$ fulfilled?

- Is the nuclear radius inside the expected range?

- Is the neutron binding energy inside the expected range?

- Is $\lambda\sqrt{E}(1+m/M)$ consistent with the atomic weight?
(m: neutron mass, M: nuclear mass)

- Do the isotopic abundances a_i for an element fulfill the conditions
$0 < a_i \leq 1$, $\sum_i a_i = 1$?

- Is more than 1 isotopic component given for an isotope?

2.1.2.2 Checks on partial cross sections

Partial cross sections must be on file in the order of ascending energy. Tests are performed by TSTBL1 as follows:

- Energy values are tested with respect to sign and order and reordered if necessary.

- The occurrence of double or multiple energy values is tested.

- Completeness of the energy range (1 meV - 15 MeV) is checked. Suspicious gaps or clusters are identified in the sequence of energies.

- It is checked whether cross section values fulfill the condition

$$10^{-10}\text{b} < \sigma < 10^{6}\text{b}$$

- In case of angular distributions the condition

$$-1 \leq \mu = \cos\ \theta \leq +1$$

is checked.

- In case of inelastic scattering thresholds are checked against excitation energies of the residual nucleus.

2.1.2.3 Consistency checks on redundant quantities

The following tests are performed by TSTBL2:

- Is the total cross section equal to the sum of all partial cross sections,

$$\sigma_T = \sum_c \sigma_c \quad ?$$

- Is the nonelastic cross section σ_{nonel} consistent with

$$\sigma_{nonel} = \sigma_T - \sigma_n = \sum_{c \neq n} \sigma_c \quad ?$$

- Is the absorption cross section σ_{abs} consistent with

$$\sigma_{abs} = \sigma_T - \sigma_n - \sigma_{n'} - \sigma_{2n} - \sigma_{3n} \quad ?$$

- Is the transport cross section σ_{tr} consistent with

$$\sigma_{tr} = \sigma_T - \bar{\mu}\, \sigma_n \quad ?$$

($\bar{\mu}$: mean cosine of lab scattering angle)

- Is the capture-to-fission ratio α consistent with

$$\alpha = \frac{\sigma_\gamma}{\sigma_f} \quad ?$$

- Is the quantity η consistent with

$$\eta = \frac{\bar{\nu}\sigma_f}{\sigma_f + \sigma_\gamma} = \frac{\bar{\nu}}{1+\alpha}$$

($\bar{\nu}$: average number of emitted neutrons per fission) ?

- In the case of cross sections for process like (n,n'α), (n,α2n) it is checked whether

$$\sigma_{n'\alpha} < \sigma_{n''},$$

$$\sigma_{\alpha 2n} < \sigma_{2n}.$$

- It is checked whether the total inelastic cross section, $\sigma_{n''}$, is equal to the sum over all partial inelastic cross sections

$$\sigma_{n''} = \sum_i \sigma_{n''}^{(i)} + \sigma_{n''}^{(c)}$$

($\sigma_{n'}^{(i)}$: contribution from the i-th excited level of the target nucleus, $\sigma_{n'}^{(c)}$: contribution from excited levels in the continuum). Threshold energies are checked against the discrete level energies.

2.1.3 Input/output specification

A brief description of the more computer-technical aspects of the program including input/output options is appended. This description is based on an unpublished internal report written by R. Meyer, the author of the program.

Appendix

Brief User's Guide to the

KEDAK print and test program PRTSTKED

BRIEF USER'S GUIDE TO THE KEDAK PRINT AND TEST PROGRAM PRTSTKED

1) Introduction

At the panel on evaluation problems in Vienna in September 1971 a number of testing programs were presented for ENDF/B and UKNDL and it was felt that such testing programs were essential in ensuring the correctness of large amounts of data. No testing program for KEDAK was available at that time.

At the end of 1971 a number of ENDF/B2 materials were converted with a very preliminary version of BRIGITTE /3/ and the lack of a testing program again turned out to be a drawback.

At about the same time it became obvious, that it would be impossible to maintain the KEDAK conventions of tabulating neutron cross section data at one and the same energy scale for all reaction types of one material. Also, this convention would be violated by the converted ENDF/B materials . Therefore, the available printing codes became obsolete.

For this reason, in December 1971 the author started programming of the printing and testing code, the first

step of which after thorough testing was operational in August 72.

This code, the input and operation of which is described in the succeeding paragraphs, was designed for printing of a large number of KEDAK reaction types and for performing a number of consistency checks and physical tests.

The version referred to hereafter is of a limited scope in that it does provide printout only of the most important data types plus a few more, but still a number of quantities cannot be handled. Also, tests are restricted to cross section data, and testing of resonance data and distribution data has not been implemented.

Since these extensions of the program had been planned from the very beginning, the program is ready to accept them. And by using the program resources already implemented, as error handling, condition handling, editing, retrieval etc., these extensions are fairly easy to make.

However, since presently it cannot be foreseen at what time work on the program can be resumed it was decided to provide a preliminary issue of input and operation description.

A detailed description of the individual program sections is in preparation and will be published separately.

2) Sectioning

The program is sectioned into the following modules:

CALPAC
ERRPAC
ISØPAC
LDFPAC
PRIPAC
RETPAC
TSTBL1
TSTBL2
TSTKED
CØNPAC

where TSTKED contains the control program and control input processing program, LDFPAC contains an overlayable system of library-oriented retrieval routines.

RETPAC contains an overlayable system of user-oriented retrieval programs.

PRIPAC contains the editing routines for the printing option.

ISØPAC performs testing of nuclear-property data. For certain detected errors ISOPAC performs corrective actions needed for the resonance testing package (which is not yet finished).

TSTBL1 performs physics checks and numeric checks on the individual, energy-dependent cross section types.

TSTBL2 performs consistency checks on redundant quantities.

CALPAC is a package of support routines for consistency tests permitting arithmetic operations on complete cross section sets.

ERRPAC is a routine package to save detected errors and to edit error messages.

CONPAC is a routine package for handling of condition codes set by detected errors.

3) Unresolved references:

Routines not in the package and supplied separately:

DEFI (A) : A routine to make the DEFINE FILE statement dynamical. A description and a FORTRAN replacement routine to simulate its functions are available.

CONVY (A) : a routine for conversion of character strings to numeric data and vice versa. A desription can be supplied on request.

XTAREA (A) (+ Entry REXTAR): A routine to dynamically allocate core storage to the program via GETMAIN and FREEMAIN macros. A description is available.

Advice how to simulate the functions of the routine with FORTRAN statements can be given on request.

FREESP (A) : A routine to determine the free core storage available for allocation of data sets. A description can be distributed. Advice how to replace the routine by a FORTRAN routine simulating its function can be given.

DATUM (A) : A routine to retrieve the current date and time. A description is available and can be distributed on request.
Advice how to replace the routine by a FORTRAN routine simulating its function can be given.

A8FØRM (with entries RAMANF, RAMØUT) (F) : routines for printing text with capital letters (12 lines high).

All simulator routines offered do only simulate the functions of the above routines for the special purposes of the testing and printing program.

Note : (A) denotes Assembler routines
(F) Fortran routines.

4) Overlay:

An overlay package is included. Together with the dynamic allocation of core storage it ensures that the program can be run within 122K region on an IBM - 370 computer if the compilation is done be the H - extended compiler. The program requires 84K core storage with overlay and 256K without, to which 10K system+buffers and core storage for dynamic allocation must be added, amounting to at least 38K.

5) JCL

The following JCL cards are required (for IBM-360 users only)

//JOB card
//_EXEC_FHLG,PARM.L = (MAP, LIST, ØVLY) if the program must be linked or
//_EXFC_FHG,LIB = ...
if a catalogued library is used to store the load module.
in the former case also

6) REGIØN AND TIME

The minimum region in which the program can be run (IBM-370, H-extended compiler) is about 122K with overlay and about 300K without Execution however may be speeded up appreciably if the region is increased. The maximum region that can be used is 8 * NP + 16 *NCS (K) where NP is the maximum number of points for a single reaction type for that material and NCS is the length of the complete energy scale (both in thousands) eg. to test Ta-181 converted from ENDF/B2 takes 48 sec of CPU time with the maximum region and 90 sec of CPU time with the minimum region, core times changing quite similarly (IBM-360/65). The running time for the printing option is not influenced by the region parameter and the option will always use 106K exactly with overlay. The use of the overlay slows down the program execution (CPU and incore time) by a factor of 1.5 - 2. Because the overlay is a multiple region overlay calls across region boundaries may occur causing additional delay. The overlay has been optimized to keep the number of such calls low. Nevertheless it may happen in very exceptional cases that excecution time exceeds reasonable limits. The only advice that can be given is to cancel the overlay option in such cases.

7) Printing with stand alone code.

In case the printing option is the only one desired, the printing part can be run on a stand-alone basis. The overlay option may then be cancelled and the program will occupy exactly 120K.

The stand-alone code can be easily constructed by replacing routines TSTKED and TSTINP in module TSTKED by a dummy, discarding modules CALPAC, ERRPAC, ISØPAC, TSTBL1, TSTBL2 and CØNPAC and selecting from RETPAC only those routines actually requested. It is suggested that the dummy routine TSTKED issues an error message if called by the control program due to erroneous control input.

```
//L.LIB_DD_defining the program library
//L.SYSIN_DD_ *
  INCLUDE cards
  ENTRY MAIN
/ *
go step:
//G.FTO1FOO1_UNIT=2314, DISP=SHR,VØL=SER=..., DSN=...
  Kedak direct access library containing the
  material to be tested/printed
//G.FTO3FOO1_UNIT=SYSDA,SPACE=(2008, 100),
  DCB=(RECFM=VBS,BLKSIZE=2008)
//G.FTO4FOOO1_DD_ like FTO3
  two scratch data sets required by the testing program
  in case the incore storage is not sufficient to hold
  all data.
//G.FTO9FOO1_DD_UNIT=SYSDA,SPACE=(80,99),DCB=(RECFM=F,
  BLKSIZE=80)
  FTO9 is a scratch data set to accept the control input
  data read from SYSIN. The latter is closed immidiately
  after copying.
//G.SYSIN_DD_*
  control input
/*
//job end card
```

Notes: -An underscore in the above text is used to denote a blank.

-Do not alter any blocksizes in case you do not wish to risk an 80A or an 806.

-Karlsruhe users: The load modules of the program sections can be included from the partitioned data set INR.STEIN.LOEAD on GFKO29. Note,that TSTKED and CONPAC have been merged into one member TSTKED. The overlay package is a sequential data set INR.STEIN.OVERLAY on GFKO29.

Load modules of the complete program are stored in STEINPDS on KAPRØS under the names PRTSTKED (version without overlay) and PRTSTKEQ (version with overlay). The overlay version should only be used if the printing option is needed. If the testing option is chosen, however, input/output becomes very expensive with the overlay version and the version without overlay is recommended.

8) Control input.(++)

Control input is provided by Fortran namelist input. Two namelists are used:

PRINT for printing
TSTKED for testing

The syntax rules are shortly summarized here.
Input is of the form:
_&'NAMELISTNAME' _ 'PARAMETERLIST' &END

where: 'NAMELISTNAME' is to be replaced by the respective namelist name (PRINT, TSTKED).
'PARAMETERLIST' is optional and may be replaced by a list of keys and associated parameters in the form:

'KEY1'=PARM1$_1$, PARM1$_2$,...,'KEY2'=PARM2,...

where 'KEY1', 'KEY2' are to be replaced by the names of the respective keys and PARM1$_1$, PARM1$_2$,... are numeric or character data assigned to them. Character data must be enclosed within quotes. Numeric data are coded as usual with the trailing comma delimiting the field width (trailing blanks are treated as zeroes).
Each key is optional and if not coded its value(s) remain(s) unchanged.

&END indicates the end of the respective namelist.
Continuation cards may be used and must start in column two. However, neither keys nor parameter values may start on one card and be continued on the next:

(++) An underscore indicates a blank hereafter.

A) PRINT (+) :

_&PRINT_NAMZ='I_1', NAMEN="MAT", "TYP", 'X', 'Y', EMIN='E_{min}',
EMAX='E_{max}', FAST=$\left\{\begin{matrix}T\\F\end{matrix}\right\}$, PAGE='$I_2$', &END

or

_&PRINT_NAMZ='I_1', MAT="MAT", TYP="TYP", EXC='X',.....as above....., &END

where

NAMZ gives the number of names for the reaction type to be printed
default: NAMZ = 2

NAMEN gives the names of the reaction types to be printed.
'MAT' ist the material name
'TYP' is the reaction type name

'X', 'Y' are eventual further (numeric) names, e.g. 'X' might be an excitation energy

alternatively

MAT= "MAT" may be used to enter a material name

TYP= "TYP" may be used to enter a reaction type name

EXC= 'X1' may be used to enter a numeric name, e.g. an excitation energy.

In case more than one single data type is to be printed, alternatively the following keys may be used:

MATS="MAT1", "MAT2", ... up to ten material names my be entered.

TYPS="TYP1", "TYP2", ... up to ten reaction names may be entered.

EXCS='X1', 'X2', ... up to ten (excitation) energies may be entered.

Printing in this case will loop upon the specified reaction types as follows.

(+) The material and reaction type names are given in Ref. /1/, /2/.

'MAT1' : 'TYP1'
'TYP2'
.
.
.
.

'MAT2' : 'TYP1'
'TYP2'
.
.
.
.

and if one of the reaction types is SGIZ, the inelastic level excitation, printing will loop on 'X1', 'X2' given.

[Default: See note 3)]

FAST = { T fast printing option
F slow printing option: data will be edited with exponents in multiples of three (eV, keV, MeV and μb, mb, b, etc.).

default: FAST=F

PAGE='I_2' indicates that page numbering should start with 'I_2'.
Page is used only, if a new material name is encountered.

default: PAGE=1

EMIN='Emin' are the energy limits (in eV) between which
EMAX='Emax' the data shall be printed. Printing will start with the last energy $\leq$ 'Emin' and will stop with the first energy $\geq$ 'Emax'.
If 'Emin' >'Emax' no energy limits are applied.

default: EMIN=0. EMAX=-1.

Notes: 1) The printing program only can access the following data types presently:
AASTATUS, ISØT1, ISØT2, ISØT3, all energy-dependent tabulated cross section data and resolved-resonance parameters.
Not included are: SGNC, LEGNC, (elastic distribution data), or unresolved-resonance data.

2) To print ISØT1 - ISØT3 code TYP='ISØT' or
TYPS = ..., 'ISØT',....,
coding 'ISØT1' or 'ISØT2' or 'ISØT3' will cause an error.

3) If TYP= 'ALL' is entered, all data types for the specified material are printed which the printing program can access.

NAMZ is ignored for TYP = 'ALL'.

If EXC = 'ALL' is coded and reaction type 'SGIZ' has been specified explicitely or is implied by TYP = 'ALL' printing will loop on all level energies for that material.

To print all informations for a given material accessible for the program, TYP='ALL',
EXC = 'ALL' may be coded.

These are the default values for TYP and EXC. No default value for the material name is supplied.

4) The current value of a parameter is default and is used if the respective key is not in the input namelist. The initial defaults at program start were given above.

5) NAMZ is only required, if not the retrieval package LDFPAC included in the program is used. Else it will be ignored.

Alias: FROM for EMIN, TO for EMAX

Abbreviations accepted:

&P for &PRINT
M for MAT
T for TYP
F for FRØM

B) TSTKED (+)

```
_&TSTKED_MAT='MAT', TSTESC = { 'NØ'
                             { 'TYP1', 'TYP2',...
                             { 'EXCLUDE', 'TYP1',...
                             { 'ALL'

      MAX='I1', EPSIL='EPS', BØUNDS='B1','B2', &END
      FROM='EFROM', TO='ETO', &END
```

where

MAT='MAT' gives the material name. Only one per list may be entered.

MAX='I1' gives the maximum number of points for a single KEDAK type.
If the retrieval routine package of the testing program is used, this parameter is ignored and will be determined by the program.
Else if set to 0 the program will assume a number of 2o ooo. MAX is only important for program efficiency optimization and an erroneous specification will not lead to an error provided it is large enough. Otherwise an 80A or 806 error might occur.

default: MAX = 0

EPSIL='EPS' specifies a boundary value for the tests on the validity of linear interpolation and gives the requested accuracy of linear representation in percent.
The program will print a warning message when the shape of the (assumed) true curve deviates from the linear interpolated curve by more than 'EPS'.

(+) For references of material and reaction type names see /1/, /2/

default: EPSIL = 5.,

BOUNDS = 'B1', 'B2'

gives the percent accuracy requested for the consistency of redundant quantities (in percent). Violation of consistency by more than 'B1' will cause a warning, by more than 'B2' an error message to be issued. ('B1' < 'B2')

default: B = 0.1, 5.,

FROM='EFROM' and
TØ='ETØ' specify lower and upper boundary of the energy range in which testing is requested. For ETØ-values less than or equal to zero the data are tested at all available energies.

defaults: FRØM=0.0, TØ=-1.0

TSTESC=
- 'NØ' no test on completeness of energy scale
- 'TYP1','TYP2',... test whether energy scales of these types are complete. Up to 5 reaction types may be entered.
- 'EXCLUDE','TYP1',... test completeness of energy scales of all types but 'TYP1',... up to 4 reaction types may be entered.
- 'ALL' test completeness of energy scale of all types.

Since one of the former conventions of the KEDAK library was that all reaction type data be tabulated along the same energy mesh points, TSTESC offers a possibility to test this convention. Violation of this convention in present files may be an indication also of other errors.
It is however of no use to test more recent files or files converted from ENDF/B on this convention.
default: TSTESC = 'NØ',

Abbreviation accepted:

M	for	MAT
TE	for	TSTESC
E	for	EPSIL
B	for	BØUNDS

Note:
For use of defaults see Note 4) of 8B).

9) Output:

It is hoped that the output in all cases is self-explanatory In most cases input errors are recognized, either by the program or by the IBM namelist processing routine and will give rise to an error message.

printing module:

large letters indicating the beginning of a new material allow easy separation. The date of the Kedak version printed together with the date of printout is displayed for each material on a heading page.

Each reaction type will be preceded by a heading to identify the data being printed. The heading will be repeated on continuation pages. Normally each reaction type will start with a new page, except for SGIZ when the level data to be printed still would fit into the current page.

For easier reference data points are numbered. Numbering will restart with 1 after 9999 had been reached.

Pages are numbered consecutively,the starting number being specified by the user. The page number is preceded by the name of the material currently being printed.

At the bottom of each page a line is displayed giving the physical units of the data printed.

testing module:

Three levels of messages are maintained according to the associated degree of error: warning, error and serious error messages. To avoid too much printout, each error has been assigned a maximum number of error messages; if exceeded the user will be told but no further messages for this error are printed.

Because large amounts of data will not usually be held in storage at any given time, error messages would be mixed if on-line printing were used. Instead, errors are collected and after finishing a class of tests, the error messages are sorted and then printed.
The limitation of the number of error messages of each type, although in most cases very useful, has also some disadvantages, e.g. violation of linear interpolation within the resonance range may happen quite frequently and indicates that the number of points chosen to represent that region does not fulfill the accuracy requirements. The individual messages however will not have much importance. In the fast energy range such a violation however could indicate a mispunch or a scaling error. Due to a large number of errors detected in the resonance range these violations might not be printed, if testing is done in the whole energy range. But it is possible to limit the tests on the fast energy range by using the input parameters FROM and TO.

Each material is preceded by separating pages containing the material name in large letters and a heading page giving the date of the KEDAK version used, the date and time of test and the status of the current program control parameters.

Each section of tests is preceded by a heading page shortly informing the user of the kind of tests performed in the succeeding section, and is ended by a page giving an account of the number of errors which occured and their error levels, where a level ≤ 4 pertains to warning messages, ≤ 8 to error messages and to serious error messages else.

Pages are numbered and the page number is preceded by the name of the material currently under test.

10) Dispatching to foreign users:

Foreign users may receive the code on request from the nuclear data evaluation group in Karlsruhe. Since the printing program may be used as a stand-alone code also, the requested program package should be indicated.

References:

/1/ D. Woll, KFK 880, 1968

/2/ B. Krieg, KFK 1725/I, 1973

/3/ J.C. Schepers, private communications and report to be published.

2.2 SELPLO: Checking and processing program for differential elastic-scattering data

Contents:

2.2 SELPLO: Checking and processing program for differential elastic-scattering data

The FORTRAN IV program SELPLO was written by R. Meyer for processing, checking and display of elastic-scattering angular distributions from and for the KEDAK library. The following sections contain descriptions of

- applications,
- input (including an example),
- output,
- main program and subroutines,
- storage requirements and running times,
- external storage devices and job control language.

A program listing is appended.

2.2.1 Applications of SELPLO

The program permits processing of differential elastic-scattering data in the center-of-mass system. These can be

- differential elastic-scattering cross sections $\frac{d\sigma_n(E,\Omega)}{d\Omega}$ $(\frac{mb}{sterad})$,

- angular distribution functions $p(\mu)$ $(-)$.

Both data types are related by

$$\frac{d\sigma_n(E,\Omega)}{d\Omega} = \frac{\sigma_n(E)}{2\pi} p(\mu),$$

with

$$\int_{2\pi} \frac{d\sigma_n(E,\Omega)}{d\Omega} d\Omega = \sigma_n(E),$$

and

$$\int_{-1}^{1} p(\mu)d\mu = 1 ,$$

where μ is the cosine of the scattering angle,

$\sigma_n(E)$ the (angle-integrated) elastic-scattering cross section

KEDAK nomenclature:

SGNC for $p(\mu)$, with μ given in the center-of-mass system.
SGN for $\sigma_n(E)$, with E given in the lab system.

Note: In the sequel the general notation SGNC will be used for both types of differential (angle-dependent) data, i.e. for $d\sigma_n/d\Omega$ and for p.

The differential elastic scattering data may be

- read from the KEDAK library or
- read from other external sources (tape, disc) or
- calculated from Legendre coefficients that were read from cards.

The following options exist:

- plotting,

- integration over all angles and normalization check whether the integral is equal to unity (for $p(\mu)$) or to the angle-integrated elastic-scattering cross section σ_n stored on KEDAK (for $d\sigma_n/d\Omega$);

- transformation to laboratory angles, calculation of the average lab cosine $\bar{\mu}_L$ (KEDAK name: MUEL) and consistency check against MUEL as stored on KEDAK;

- generation of statistics on the deviations between σ_n(or $\bar{\mu}_L$) as calculated from SGNC and σ_n (or $\bar{\mu}_L$) as stored on KEDAK;

- calculation of SGNC in the center-of-mass system from Legendre coefficients for an input cosine grid with check whether this grid is fine enough;

- normalization of SGNC to unity and storage in KEMA input format for later incorporation in KEDAK.

2.2.2 SELPLO input

Input quantities other than SGNC are read according to the FORTRAN NAMELIST conventions into NAMELIST/IN/. The (card) input has the form

_&IN_MAT=...,EMIN=...,EMAX=...,other names, END

where underscores _ denote blanks.

The quantities and options which can appear in the IN name list are as follows:

Name	Explanation	Default value	Type
MAT	material name (KEDAK name)		REAL*8
EMIN	lower energy limit in eV	0.	REAL*4
EMAX	upper energy limit in eV	20.E+6	"
PLOT	= { T : plot / F : do not plot } SGNC	F	LOGICAL*1
SIGMEL	= { T : compare / F : do not compare } the angular integral of SGNC with SGN as stored on KEDAK	F	"
MUL	= { T : compare / F : do not compare } MUEL calculated from SGNC with MUEL as stored on KEDAK	F	"
NORM	= { T : normalize / F : do not normalize } SGNC to unity, write result in KEMA input format on unit 2, prepare punching from unit 3.	F	"
STATIS	= { T : generate statistics / F : no statistics } on deviations between integrals calculated from SGNC and SGN values on KEDAK if SIGMEL=T, or between corresponding MUEL values if MUL=T.	F	"

Name	Explanation	Default value	Type
INTYP	= 'KEDA' read SGNC from KEDAK, unit 1 'TAPE' " SGNC " tape 'CARD' " SGNC " cards	'KEDA'	REAL*4
A	atomic weight of material specified by MAT (^{12}C system) if it is not to be read from KEDAK		"
DRUCK	= T : print / F : do not print Legendre coefficients read from cards and calculated SGNC values	F	LOGICAL*1
NUMKED	number ≤ 101 of cosine grid points only required if INTYP='CARD'		INTEGER*4
ITAP	unit > 8 from which SGNC values are to be read if INTYP='TAPE'.		"
PLOFOR	= T : ink plot / F : pencil plot if PLOT=T	T	REAL*4

For INTYP='KEDAK' or INTYP='TAPE' another _&IN_... namelist can follow. If not, the end of input is reached. For INTYP='CARD' the name list NAMELIST/CD/ must be specified for each incident energy by

_&CD_MODE=...,ENERGY=...,other names, END.

The names that can appear in the &CD name list are as follows:

Name	Explanation	Default value	Type
MODE	= 'LEG' : read Legendre coefficients 'TAB' : read tabulated SGCN values (not active in present code verison)	'LEG'	REAL*8
ENERGY	incident-neutron energy, EMIN≤ENERGY≤EMAX		REAL*4

Name	Explanation	Default value	Type
NUMLC	number of Legendre coefficients, $\leq$ 20, if MODE='LEG'		INTEGER*4
NEWTAB	= 'NO' : use old cosine grid, 'READ' : read cosine grid values 'EQI' : calculate grid of equi-distant cosines	'NO'	REAL*8
NUMCOS	number $\leq$ 101 of cosine grid values for NEWTAB='READ' or ='EQI'		INTEGER*4
COSTAB	list of NUMCOS cosine grid values separated by commas, in descending order, for NEWTAB='READ'.		REAL*4 array(101)
LEGC	list of NUMLC Legendre coefficients separated by commas, in descending order, for MODE='LEG'		

If another material or the present material with different input parameters is to be treated a new _&IN_MAT=... name list follows, otherwise the end of input is reached. Input of tabulated SGNC values from cards is not possible with the present version of SELPLO. Input of tabulated SGNC values via tape utilizes KEMA input format. For each energy the following quantities are read:

NUM	INTEGER*4	number of 4-byte words following NUM in this record, i.e. NUM=10+2*NUMCOS where NUMCOS is the number of cosine values for which SGNC values are to be read
'ADD'	REAL*8	constant
3	INTEGER*4	constant
MAT	REAL*8	material name, e.g. 'FE__'
'SGNC'	REAL*8	constant
ENERGY	REAL*4	incident energy
1	INTEGER*4	constant
1	"	"
(X(I),Y(I), I=1,NUMCOS)	REAL*4	abscissae (cosines) and ordinates (SGNC values)

These data are read on unit ITAP and must be written with LRECL=80. The first byte of the first logical record must be occupied, all following records must begin with a blank.

Input Example:

```
_&IN_MAT='FE,EMIN=1.E+4,EMAX=1.E+5,
-PLOT=T,SIGMEL=T,MUL=T,STATIS=T, END
```

2.2.3 SELPLO output

The output is essentially self-explanatory. If $\bar{\mu}_L^S$ calculated from SGNC and $\bar{\mu}_L^K$ stored on KEDAK are compared, the following quantity is printed:

$$\frac{\delta\bar{\mu}_L}{1-\bar{\mu}_L^K} = \frac{\left|\bar{\mu}_L^K - \bar{\mu}_L^S\right|}{1-\bar{\mu}_L^K} = \left|\frac{\sigma_{tr}^S - \sigma_{tr}^K}{\sigma_{tr}^K}\right|$$

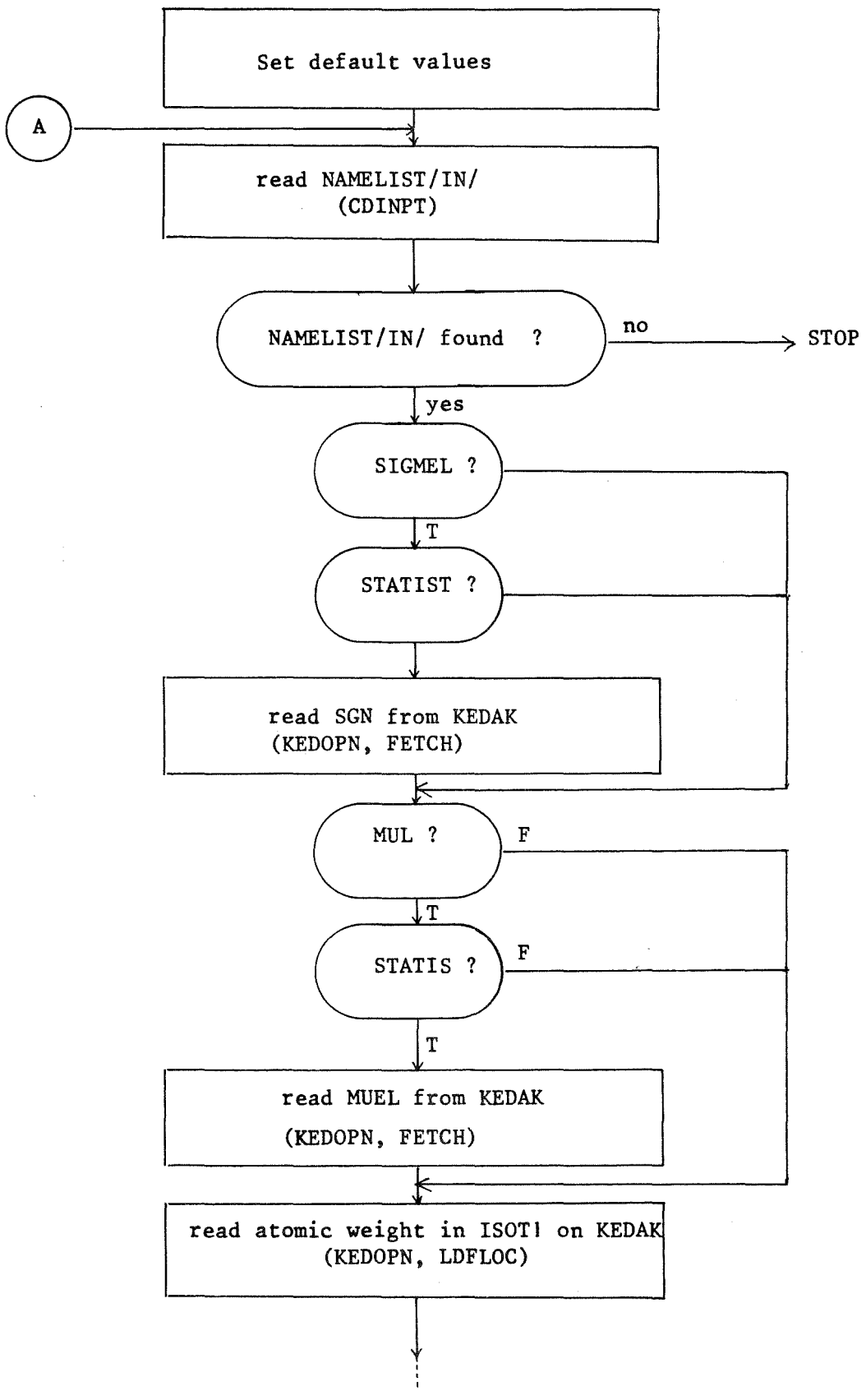
which gives the relative deviation of the elastic transport cross section

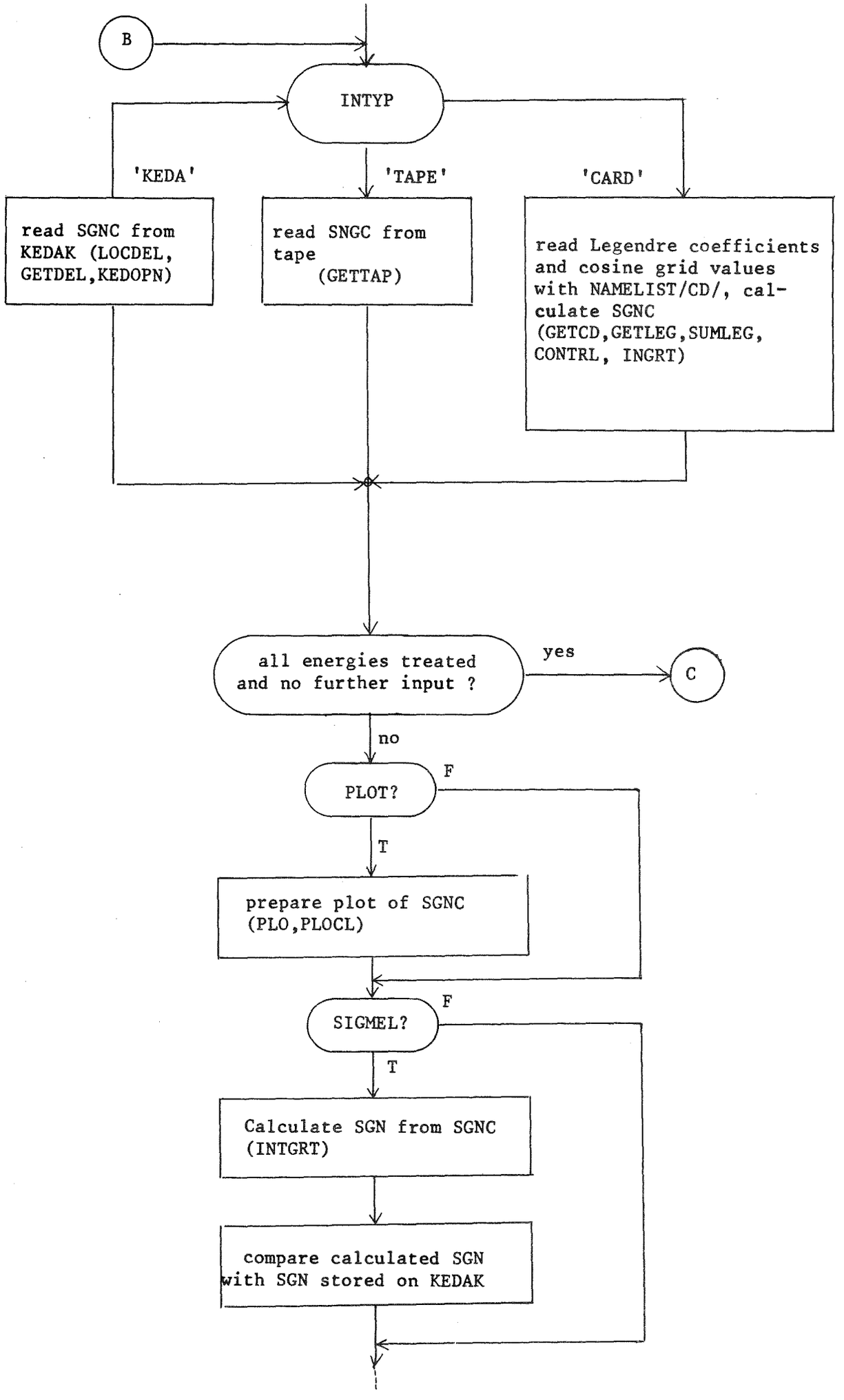
$$\sigma_{tr} = (1-\bar{\mu}_{el})\sigma_n$$

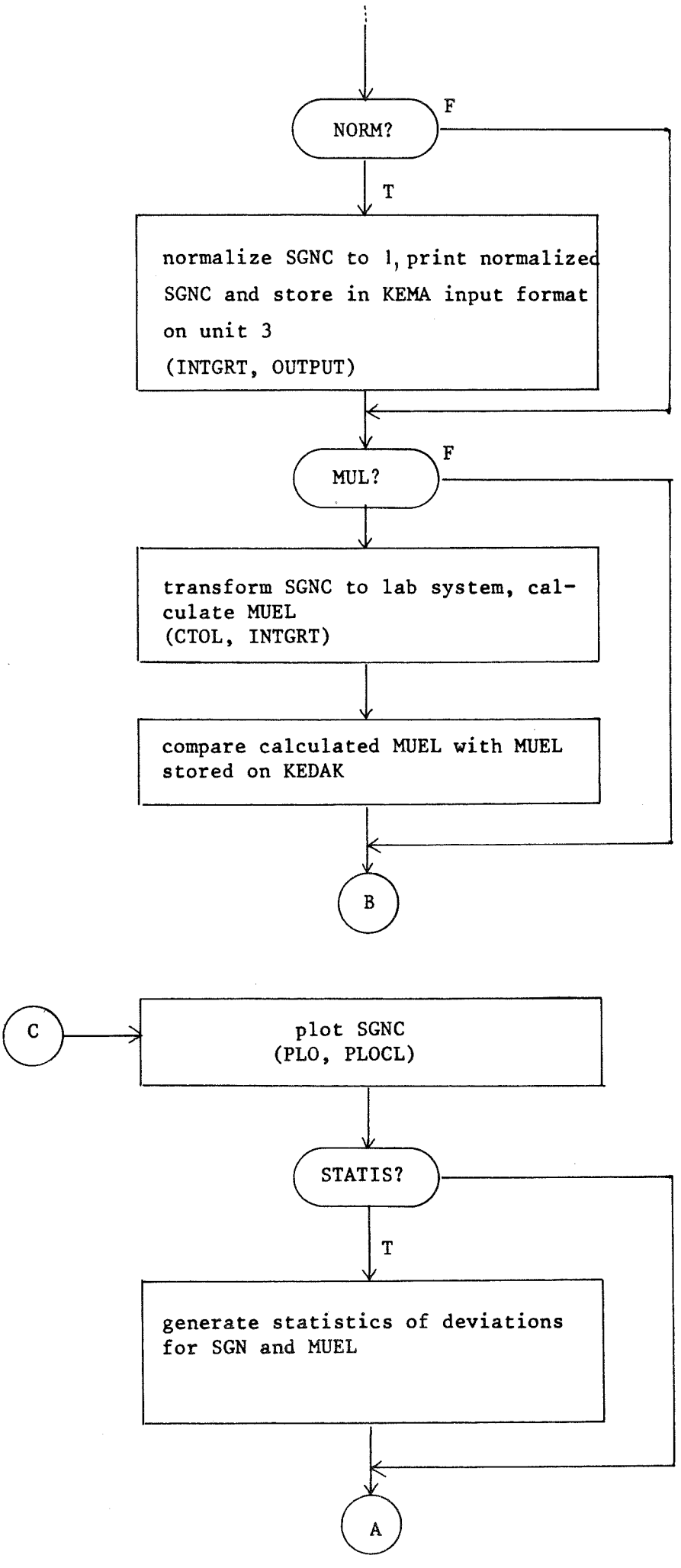
if $\bar{\mu}_L$ is calculated from SGNC rather than retrieved from KEDAK.

2.2.4 Short description of SELPLO main program and subroutines

Flow diagram of SELPLO main program
(T=TRUE, F=FALSE, subroutines called are shown in brackets)




B
INTYP
'KEDA'
'TAPE'
'CARD'
read SGNC from KEDAK (LOCDEL, GETDEL,KEDOPN)
read SNGC from tape (GETTAP)
read Legendre coefficients and cosine grid values with NAMELIST/CD/, calculate SGNC (GETCD,GETLEG,SUMLEG, CONTRL, INGRT)
all energies treated and no further input ?
yes
C
no
PLOT?
F
T
prepare plot of SGNC (PLO,PLOCL)
SIGMEL?
F
T
Calculate SGN from SGNC (INTGRT)
compare calculated SGN with SGN stored on KEDAK


NORM?
F
T
normalize SGNC to 1, print normalized SGNC and store in KEMA input format on unit 3
(INTGRT, OUTPUT)
MUL?
F
transform SGNC to lab system, calculate MUEL
(CTOL, INTGRT)
compare calculated MUEL with MUEL stored on KEDAK
B
C
plot SGNC
(PLO, PLOCL)
STATIS?
T
generate statistics of deviations for SGN and MUEL
A

SELPLO subroutines

CDINPT	reads card input from unit 5 and stores it momentarily on unit 4
KEDOPN	opens nuclear data file KEDAK, calls LDFOPN
FETCH	reads SGN or MUEL from KEDAK, calls LDFLOC, LDFNXT
LOCDEL	reads SGNC from KEDAK (ENTRY GETDEL), calls LDFLOC, LDFNXT
GETTAP	reads SGNC from tape
GETCD	reads SGNC or Legendre coefficients with cosine grid values from cards, calls GETLEG, GETTAB
GETLEG	calculates SGNC from Legendre coefficients for all cosine grid values, checks whether cosine grid is fine enough, calls SUMLEG, CONTRL
SUMLEG	calculates SGNC from Legendre coefficients for given energy
CONTRL	calculates first two Legendre moments of a distribution and compares them with the generating Legendre coefficients
GETTAB	dummy in present version of SELPLO
INTGRT	integrates according to trapezoidal rule
OUTPUT	prints SGNC, stores SGNC in KEMA input format

CTOL	transforms from center-of-mass to lab system
ERMSG	prints error messages
PLO	plots SGNC (ENTRY PLOCL), calls RASTER, GRIDLI, PLOTA
RASTER	plots a grid of horizontal and vertical lines, calls RALIN, RALOG, PLOTA
RALIN	generates grid for linear scale
RALOG	generates grid for logarithmic scale
GRIDLI	generates parameters for linear axis corresponding to size of plot paper
PLOTA	is a Karlsruhe general-purpose plot subroutine /1/
LDF	reads the nuclear data library KEDAK /2/, (ENTRIES LDFOPN, LDFLOC, LDFNXT)
STRING,CONVX	system routines /3/
FREEFO	conversion of input data to binary representation /4/

2.2.5 Storage and time requirements on IBM/360-65 and /370-168

SELPLO uses fixed-dimension arrays. At most 2500 SGN and MUEL values can be treated in the energy range EMIN...EMAX specified in the input. The maximum number of SGNC values is 101.

The source program requires about 196 k bytes. An additional 30-50 k bytes are needed for buffers depending on number and blocking of external units. About 0.05 s of CPU time are needed per processed angular distribution.

2.2.6 External storage, job control

Unit	contents	required
1	KEDAK	if INTYP='KEDA' or SIGMEL=T or MUL=T or if atomic weight is read from KEDAK
2	normalized SGNC values in KEMA input format (for output other than printing)	if NORM=T
3	normalized SGNC values (for printing)	if NORM=T
4	intermediate storage of input read from unit 5	always
5	card input	always } units assigned
6	printed output	always } automatically, if standard procedures are used.
7	punched output for IBM 1130 plotter	if PLOT=1
8	SGNC in binary representation (for reading by unit ITAP)	if INTYP='TAPE'
ITAP	SGNC in KEMA input format	if INTYP='TAPE'
(>8)	(input)	

Utilization of all external units (with ITAP=9) requires the following job control information:

```
//INR901SE JØB(0901,106,P6M1A),GØEL,CLASS=A,
// REGIØN=250k,TIME=1
/*FØRMAT PU,DDNAME=FT07F001,FØRMS=TUSCHE
/*SETUP DEVICE=2314,ID=GFK050
/*SETUP DEVICE=2314,ID=GFK029
// EXEC FGLG
//L.PLIB DD UNIT=2314,VØL=SER=GFK029,
//_DSN=INR.STEIN.LØAD,DISP=SHR
//L.SYSIN DD *
 INCLUDE PLIB(RETPAC,LDFPAC,DEFI,SELPLØ)
 ENTRY MAIN
/*
//G.FT01F001 DD UNIT=2314,VØL=SER=GFK050,
// DSN=KEDAK3,DISP=SHR
//G.FT02F001 DD UNIT=SYSDA,SPACE=(TRK,20),
// DISP=(NEW,DELETE)
//G.FT03F001 DD SYSØUT=A
//G.FT04F001 DD UNIT=SYSDA,SPACE=(TRK,(100,10)),
//_DCB=(RECFM=FB,LRECL=80,BLKSIZE=6400),
//_DISP=(NEW,DELETE)
//G.FT07F001 DD SYSØUT=T
//G.FT08F001 DD UNIT=SYSDA,SPACE=(TRK,10)
//G.FT09F001 DD UNIT=2314,VØL=SER=GFK016,
// DSN=SELPLØ.DATA,DISP=SHR
//G.SYSIN DD *
 &IN_...(input parameters)...,&END
/*
//
```

References

/1/ S. Heine, PLOTA, Programmbeschreibung (1967) and updates, unpublished

/2/ Present Compendium Ch. III.2

/3/ K. Gogg, CONVX, Programmbeschreibung, unpublished
K. Gogg. STRING, Programmbeschreibung, unpublished

/4/ H. Bachmann, (1970) unpublished

Appendix: SELPLØ program listing

```
      LOGICAL*1 SIGMEL,MUL,NORM,PLOT,STATIS,DRUCK                                10
      REAL*4 MU,INTGRL,INTGR,MUB,MUK,INTYP,KEDA/'KEDA'/                          20
      DIMENSION ARG(4),NARG(3),COS(101),DSIG(101),ENEL(2500),SIGEL(2500)         30
     1,ENMU(2500),MU(2500),TXT(20)                                               40
      DIMENSION XZ(1001),YZ(1001),ZZ(1001),NMU(42),NSL(22)                       50
      REAL*8 MAT,NAME(3),SGN/'SGN     '/,                    MUEL/'MUEL        60
     1'/,BLANK/'        '/,ISOT1/'ISOT1   '/,TTEND/'ENDE'/                       70
      COMMON INPUT,KOUT,ITAP,ITAB                                                80
      DATA CARD/'CARD'/,TAPE/'TAPE'/,TIN/' &IN'/,TEND/'ENDE'/,I2/2/              90
      LOGICAL FIRST/T/                                                          100
      DATA TUSCHE/'T'/                                                          110
      NAMELIST/IN/ MAT,EMIN,EMAX,PLOT,SIGMEL,MUL,NORM,STATIS,INTYP,A,           120
     1  DRUCK,NUMKED,ITAP,PLOFOR                                                130
      DATA ATN/1.0086654/,PI/3.14159265/                                        140
      CALL ESPIE                                                                150
      KEKONT=0                                                                  160
      LENX=101                                                                  170
      LAXX=1001                                                                 180
      PLOFOR=TUSCHE                                                             190
      MAXNUM=2500                                                               200
      ITAP=8                                                                    210
      ITAB=2                                                                    220
      INPUT=4                                                                   230
      KIN=5                                                                     240
      KOUT=6                                                                    250
      CALL CDINPT(KIN)                                                          260
      EMIN=0.                                                                   270
      EMAX=20.E+6                                                               280
      PLOT=.FALSE.                                                              290
      SIGMEL=.FALSE.                                                            300
      MUL=.FALSE.                                                               310
      NORM=.FALSE.                                                              320
      STATIS=.FALSE.                                                            330
      DRUCK=.FALSE.                                                             340
      INTYP=KEDA                                                                350
C                                                                               360
  999 WRITE(KOUT,605)                                                           370
  605 FORMAT(1H1)                                                               380
      NUMELD=0                                                                  390
      NPLO=0                                                                    400
      NMU(1)=0                                                                  410
      NSL(1)=0                                                                  420
      CALL STRING(NMU(2),NMU(1),164)                                            430
      CALL STRING(NSL(2),NSL(1),84)                                             440
      A=0.                                                                      450
      NUMKED=0                                                                  460
    6 READ(INPUT,502) TXT                                                       470
  502 FORMAT(20A4)                                                              480
      IF(TXT(1).NE.TIN) GOTO 12                                                 490
      BACKSPACE INPUT                                                           500
      READ(INPUT,IN,END=89)                                                     510
      WRITE(KOUT,600) MAT,EMIN,EMAX,PLOT,PLOFOR,SIGMEL,MUL,NORM,STATIS,D        520
     1RUCK,INTYP,NUMKED                                                         530
  600 FORMAT(' MAT=',A8/                                                        540
     1' EMIN=',E15.5,8X,'EMAX=',E15.5/                                          550
     2' PLOT=',L2,',  PLOFOR=',A2,                                             560
     X',  SIGMEL=',L2,',  MUL=',L2,',  NORM=',L2,',  STATIS=',L2,             570
     X',  DRUCK=',L2/                                                          580
     4' INTYP=',A4,',  NUMKED=',I3)                                            590
      IF(NORM) WRITE(KOUT,622) ITAB                                             600
      IF (INTYP.EQ.TAPE) GO TO 11                                               610
  622 FORMAT(' ITAB=',I2)                                                       620
      NAME(1)=MAT                                                               630
      NAME(3)=BLANK                                                             640
      GOTO 14                                                                   650
C                                                                               660
C FEHLERZWEIG.                                                                  670
   12 IF(TXT(1).EQ.TEND) GOTO 89                                                680
      CALL ERMSG(TIN,TXT)                                                       690
      GOTO 6                                                                    700
C                                                                               710
   11 WRITE(KOUT,631)ITAP                                                       720
  631 FORMAT('   TAPE INPUT UNIT: ITAP=',I3)                                   730
      IF (ITAP.GT.8) GO TO 13                                                   740
      WRITE(KOUT,632)                                                           750
  632 FORMAT( ' UNITS 1 - 8 ARE  INTERNALLY RESERVED USE ANOTHER UNIT           760
     1 FOR TAPE INPUT')                                                         770
        GO TO 999                                                               780
   13  CALL FREE72(ITAP,8,6,MU,MU,MU)                                           790
C                                                                               800
C    ITAP  INPUT UNIT ANGULAR DISTRIBUTIO IN KEMAINPUT FORMAT                   810
C     8    UNIT ON WHICH THE INPUT INFORMATION IS WRITTEN IN                    820
C          BINARY FORM                                                          830
C     6    PROTOCOLE UNIT                                                       840
C     MU   WORKING SPACE                                                        850
C                                                                               860
C SGN WERDEN VON KEDAK NACH (ENEL,SIGEL) GELESEN                                870
   14 IF(.NOT.SIGMEL.OR..NOT.STATIS) GOTO 20                                    880
      NAME(2)=SGN                                                               890
      NARG(1)=2                                                                 900
      IF(KEKONT.EQ.0) CALL KEDOPN(KEKONT,&95)                                   910
      CALL FETCH(NARG,NAME,EMIN,EMAX,ENEL,SIGEL,MAXEL,MAXNUM,&15)               920
      GO TO 20                                                                  930
   15 STATIS=.FALSE.                                                            940
C                                                                               950
C MUEL WERDEN VON KEDAK NACH (ENMU,MU) GELESEN                                  960
   20 IF(.NOT.MUL.OR..NOT.STATIS) GOTO 30                                       970
      NAME(2)=MUEL                                                              980
      NARG(1)=2                                                                 990
      IF(KEKONT.EQ.0) CALL KEDOPN(KEKONT,&95)                                  1000
      CALL FETCH(NARG,NAME,EMIN,EMAX,ENMU,MU,MAXMU,MAXNUM,&25)                 1010
      GOTO 30                                                                  1020
   25 STATIS=.FALSE.                                                           1030
C                                                                              1040
C ISOT1 WIRD GEHOLT.ATOMGEWICHT WIRD EINGELESEN.                               1050
   30 MEL=1                                                                    1060
      MMU=1                                                                    1070
      IF(A.NE.0.) GOTO 28                                                      1080
      NARG(1)=2                                                                1090
      NAME(1)=MAT                                                              1100
```

```
      NAME(2)=ISOT1                                                     1110
      IF(KEKONT.EQ.0) CALL KEDOPN(KEKONT,&95)                           1120
      CALL LOFLOC(NERR,NARG,NAME,ARG)                                   1130
      IF (NERR.NE.0) GO TO 27                                           1140
      WRITE(KOUT,633)NAME(1)                                            1150
 633  FORMAT(' ***** ATIOMIC WEIGHT FOR  ',A8,' NOT AVAILABLE ON KEDAK E 1160
     1NTER IT VIA INPUT PARAMETER A ******')                            1170
       GO TO 999                                                        1180
 27    A=ARG(1)                                                         1190
   28 WRITE(6,650) A                                                    1200
  650 FORMAT('  A=',F12.6)                                             1210
      A=A/ATN                                                           1220
C                                                                       1230
C ELASTISCHE WINKELVERTEILUNGEN WERDEN VON KEDAK GEHOLT (SGNC)          1240
C             BZW VON BAND IN AUFNAHMEFORMAT FUER KEDAK.                1250
      WRITE(KOUT,624)                                                   1260
  624 FORMAT(//' ENERGIEN IN EV,SIGMAS IN BARN'/)                       1270
      IF(.NOT.STATIS.AND.(SIGMEL.OR.MUL)) WRITE(KOUT,623)               1280
  623 FORMAT(3X,'ENERGIE',25X,'SIGMA EL',29X,'MUE'//)                   1290
      IF(STATIS) WRITE(KOUT,621)                                        1300
  621 FORMAT(                                                           1310
     13X,'ENERGIE',15X,'SGN',7X,'AUS SGNC',3X,'DIFF(%)',9X,'MUEL',5X,'AU 1320
     2S SGNC',4X,'DIFF',3X,'D(MU)/(1-MU)'//)                            1330
      EL=EMIN                                                           1340
   31 MAX=NUMKED                                                        1350
      IF(INTYP.EQ.KEDA) GOTO 34                                         1360
      IF(INTYP.EQ.TAPE) GOTO 33                                         1370
      IF(INTYP.EQ.CARD) GOTO 36                                         1380
      WRITE(KOUT,620)                                                   1390
  620 FORMAT('  ALS EINGABEMEDIEN SIND DERZEIT NUR BAND,KEDAK UND KARTEN 1400
     1 ERLAUBT')                                                        1410
      GOTO 999                                                          1420
   33 CALL GETTAP(MAT,EL,COS,DSIG,MAX,EMIN,EMAX,NUMKED,LENX,&80)        1430
      IF(FIRST.AND.NUMKED.EQ.0) NUMKED=MAX                              1440
      IF(FIRST) FIRST=.FALSE.                                           1450
      IF(MAX.NE.NUMKED) GOTO 94                                         1460
      GOTO 35                                                           1470
   34 IF(FIRST) GOTO 341                                                1480
      CALL GETDEL(EL,COS,DSIG,MAX,&80,&80)                              1490
      GOTO 342                                                          1500
  341 IF(KEKONT.EQ.0) CALL KEDOPN(KEKONT,&95)                           1510
      CALL LOCDEL(MAT,EL,COS,DSIG,MAX,&80,&80)                          1520
      IF(FIRST.AND.NUMKED.EQ.0) NUMKED=MAX                              1530
      FIRST=.FALSE.                                                     1540
      GOTO 342                                                          1550
  342 IF(EL.GT.EMAX) GOTO 80                                            1560
      IF(MAX.NE.NUMKED) GOTO 94                                         1570
      GOTO 35                                                           1580
   36 CALL GETCD(COS,DSIG,MAX,EL,ZZ,LENX,DRUCK,&80)                     1590
      IF(EL.GT.EMAX) GOTO 80                                            1600
   35 WRITE(KOUT,607) EL                                                1610
  607 FORMAT(/2X,1PE12.3)                                               1620
      NUMELD=NUMELD+1                                                   1630
C                                                                       1640
C PLOTTEN DER ELASTISCHEN VERTEILUNGEN                                  1650
      IF(.NOT.PLOT) GOTO 38                                             1660
      CALL PLO(COS,DSIG,MAX,EL,MAT,3)                                   1670
      NPLO=NPLO+1                                                       1680
      IF(NPLO.LT.4) GOTO 38                                             1690
      CALL PLOCL(PLOFOR)                                                1700
      NPLO=0                                                            1710
      GOTO 38                                                           1720
C                                                                       1730
C SGNC WIRD UEBER 4PI INTEGRIERT                                        1740
   38 IF(.NOT.SIGMEL) GOTO 47                                           1750
      CALL INTGRT(COS,DSIG,MAX,INTGRL)                                  1760
      SN=2.*PI*INTGRL                                                   1770
C                                                                       1780
C ZUGEHOERIGER ELASTISCHER QUERSCHNITT WIRD GEHOLT                      1790
      IF(.NOT.STATIS) GOTO 43                                           1800
   39 IF(MEL.GT.MAXEL) GOTO 43                                          1810
      IF(ENEL(MEL)-EL) 40,42,44                                         1820
   40 MEL=MEL+1                                                         1830
      GOTO 39                                                           1840
   42 SNK=SIGEL(MEL)                                                    1850
      GOTO 46                                                           1860
   43 WRITE(KOUT,608) SN                                                1870
  608 FORMAT('+',22X,'------',2X,F10.4,3X,'----')                       1880
      GOTO 47                                                           1890
   44 SNK=SIGEL(MEL-1)+(SIGEL(MEL)-SIGEL(MEL-1))*(EL-ENEL(MEL-1))/(ENEL( 1900
     1MEL)-ENEL(MEL-1))                                                 1910
   46 SN=SN*SNK                                                         1920
      DIFF=100.*ABS(SNK-SN)/SNK                                         1930
      WRITE(KOUT,609) SNK,SN,DIFF                                       1940
  609 FORMAT('+',18X,F10.4,2X,F10.4,2X,F5.1)                            1950
      IF(DIFF.GE.1.) GOTO 41                                            1960
      DIFF=DIFF+1.                                                      1970
      GOTO 45                                                           1980
   41 DIFF=DIFF/5.+2.                                                   1990
      IF(DIFF.GT.21.) DIFF=22.1                                         2000
   45 NSL(DIFF)=NSL(DIFF)+1                                             2010
C                                                                       2020
C NORMIEREN DER ELASTISCHEN VERTEILUNGEN AUF 1.                         2030
   47 IF(.NOT.NORM) GOTO 50                                             2040
      IF(SIGMEL) GOTO 49                                                2050
      CALL INTGRT(COS,DSIG,MAX,INTGRL)                                  2060
      SN=2.*PI*INTGRL                                                   2070
   49 FNORM=1./SN                                                       2080
      DO 48 M=1,MAX                                                     2090
   48 DSIG(M)=FNORM*DSIG(M)                                             2100
      CALL OUTPUT(MAT,EL,COS,DSIG,MAX)                                  2110
C                                                                       2120
C BERECHNUNG VON MUEL AUS SGNC                                          2130
   50 IF(.NOT.MUL) GOTO 31                                              2140
      CALL CTOL(COS,DSIG,MAX,XZ,YZ,    LAX,LAXX,A)                    2150
      CALL INTGRT(XZ,YZ,LAX,INTGRL)                                     2160
      DO 52 I=1,LAX                                                     2170
   52 ZZ(I)=XZ(I)*YZ(I)                                                 2180
      CALL INTGRT(XZ,ZZ,LAX,INTGR)                                      2190
      MUB=INTGR/INTGRL                                                  2200
```

```
C                                                                       2210
C KEDAK-MUEL HOLEN                                                      2220
      IF(.NOT.STATIS) GOTO 59                                           2230
   54 IF(MMU.GT.MAXMU) GOTO 59                                          2240
      IF(ENMU(MMU)-EL) 56,58,60                                         2250
   56 MMU=MMU+1                                                         2260
      GOTO 54                                                           2270
   58 MUK=MU(MMU)                                                       2280
      GOTO 62                                                           2290
   59 WRITE(KOUT,610) MUB                                               2300
  610 FORMAT(' +',59X,'------',2X,F8.4,2(4X,'------'))                  2310
      GOTO 31                                                           2320
   60 MUK=MU(MMU-1)+(MU(MMU)-MU(MMU-1))*(EL-ENMU(MMU-1))/(ENMU(MMU)-ENMU 2330
     1(MMU-1))                                                          2340
   62 DIFF=ABS(MUK-MUB)                                                 2350
      DIFF1=DIFF/(1.-MUK)                                               2360
      WRITE(KOUT,611) MUK,MUB,DIFF,DIFF1                                2370
  611 FORMAT(' +',57X,3(F8.4,2X),G12.4)                                 2380
      IF(DIFF.GE.1.E-2) GOTO 63                                         2390
      DIFF=DIFF+1.01                                                    2400
      GOTO 64                                                           2410
   63 DIFF=DIFF*20.+2.                                                  2420
      IF(DIFF.GT.41.) DIFF=42.1                                         2430
   64 NMU(DIFF)=NMU(DIFF)+1                                             2440
      GOTO 31                                                           2450
C                                                                       2460
C STATISTIK                                                             2470
C                                                                       2480
   80 CALL PLOCL(PLOFOR)                                                2490
      FIRST=.TRUE.                                                      2500
      WRITE(KOUT,626) NUMELD                                            2510
  626 FORMAT('1'/' ANZAHL DER BEARBEITETEN ELASTISCHEN VERTEILUNGEN = ', 2520
     1I5)                                                               2530
      IF(.NOT.STATIS) GOTO 999                                          2540
      WRITE(KOUT,618)                                                   2550
  618 FORMAT(//2X,'VERTEILUNG DER ABWEICHUNGEN ZWISCHEN KEDAK TABELLIERT 2560
     1 UND KEDAK GERECHNET AUS DEN DIFFERENTIELLEN VERTEILUNGEN')       2570
      IF(.NOT.SIGMEL) GOTO 85                                           2580
      WRITE(KOUT,619)                                                   2590
  619 FORMAT(//' FUER SGN:'/)                                           2600
      DO 82 M=1,21                                                      2610
      MA=M+1                                                            2620
      DO 81 J=MA,22                                                     2630
   81 NSL(M)=NSL(M)+NSL(J)                                              2640
      MM=5*(M-2)                                                        2650
      IF(M.EQ.2) MM=1                                                   2660
      IF(M.EQ.1) MM=0                                                   2670
      WRITE(KOUT,627) MM,NSL(M)                                         2680
  627 FORMAT(' DIFF>',I4,' %',I6)                                       2690
   82 CONTINUE                                                          2700
      MM=100                                                            2710
      WRITE(KOUT,627) MM,NSL(22)                                        2720
   85 IF(.NOT.MUL) GOTO 999                                             2730
      WRITE(KOUT,628)                                                   2740
  628 FORMAT(//' FUER MUEL:'/)                                          2750
      DO 84 M=1,41                                                      2760
      MA=M+1                                                            2770
      DO 83 J=MA,42                                                     2780
   83 NMU(M)=NMU(M)+NMU(J)                                              2790
      DIFF=.05*FLOAT(M-2)                                               2800
      IF(M.EQ.1) DIFF=0.                                                2810
      IF(M.EQ.2) DIFF=.01                                               2820
      WRITE(KOUT,629) DIFF,NMU(M)                                       2830
  629 FORMAT(' DIFF>',F4.2,I6)                                          2840
   84 CONTINUE                                                          2850
      DIFF=2.                                                           2860
      WRITE(KOUT,629) DIFF,NMU(42)                                      2870
      GOTO 999                                                          2880
   89 WRITE(KOUT,615)                                                   2890
  615 FORMAT('1ENDE DER EINGABE ERREICHT')                              2900
      IF(.NOT.NORM) GOTO 90                                             2910
      WRITE(ITAB) I2,TTEND                                              2920
       WRITE(3,634)I2,TTEND                                             2930
  634  FORMAT(I8,3X,A4)                                                 2940
      ENDFILE ITAB                                                      2950
       WRITE(3,635) ITAB                                                2960
  635  FORMAT(' END OF FILE WRITTEN ON UNIT',I3)                        2970
   90 STOP                                                              2980
C                                                                       2990
C FEHLERZWEIGE                                                          3000
   94 WRITE(KOUT,616) MAT,EL,NUMKED                                     3010
  616 FORMAT(' FUER ',A8,' SGNC EL=',E12.4,' IST DIE ANZAHL DER SAETZE N 3020
     1ICHT = ',I3)                                                      3030
      MAX=NUMKED                                                        3040
      GOTO 35                                                           3050
C                                                                       3060
   95 WRITE(KOUT,617)                                                   3070
  617 FORMAT(' LDFOPN-AUFRUF BLIEB OHNE ERFOLG')                        3080
      GOTO 80                                                           3090
C                                                                       3100
      END                                                               3110
```

```
      SUBROUTINE FETCH(NARG,NAME,EMIN,EMAX,ARG,WERT,MAX,MAXE,*)          10
      COMMON INPUT,KOUT                                                  20
      DIMENSION ARG(2500),WERT(2500),Z(2),NARG(3)                        30
      REAL*8 NAME(3)                                                     40
      I=1                                                                50
      CALL LDFLOC(NERR,NARG,NAME,Z)                                      60
      IF(NERR.EQ.0) GOTO 30                                              70
      IF(Z(1).LT.EMIN) GOTO 20                                           80
      IF(Z(1).GT.EMAX) GOTO 32                                           90
      ARG(I)=Z(1)                                                       100
      WERT(I)=Z(2)                                                      110
      I=I+1                                                             120
   20 CALL LDFNXT(NERR,NARG,NAME,Z)                                     130
      IF(NERR.EQ.0) GOTO 22                                             140
      IF(Z(1).LT.EMIN) GOTO 20                                          150
```

```
      IF(Z(1).GT.EMAX) GOTO 24                                          160
      ARG(I)=Z(1)                                                       170
      WERT(I)=Z(2)                                                      180
      I=I+1                                                             190
      IF(I.GT.MAXE) GOTO 34                                             200
      GOTO 20                                                           210
   22 WRITE(KOUT,605) (NAME(J),J=1,2)                                   220
  605 FORMAT(' EMAX > OBERSTER ENERGIEWERT AUF KEDAK FUER  ',2A8)       230
   24 MAX=I-1                                                           240
      WRITE(KOUT,609) (NAME(J),J=1,2),MAX                               250
      RETURN                                                            260
   30 WRITE(KOUT,606) (NAME(I),I=1,2),EMIN,EMAX                         270
  606 FORMAT(' FUER ',2A8,' WURDEN ZWISCHEN',E12.4,'EV  UND',E12.4,'EV  280
     1AUF KEDAK KEINE DATEN GEFUNDEN')                                  290
      RETURN 1                                                          300
   32 WRITE(KOUT,607) (NAME(I),I=1,2),EMAX,EMIN                         310
  607 FORMAT(' FUER ',2A8,'IST ANGEGEBENES EMAX=',E12.4,'EV KLEINER ALS 320
     11. KEDAKENERGIE ODER ANGEGEBENES EMIN=',E12.4,'EV')               330
      RETURN 1                                                          340
   34 WRITE(KOUT,608) (NAME(J),J=1,2),MAXE,EMIN,ARG(I-1)                350
  608 FORMAT(' ANZAHL DER ENERGIEN AUF KEDAK FUER ',2A8,' GROESSER ALS '360
     1,I5  /' ES WIRD DAS EINLESEN VON KEDAKWERTEN BEI DIESER ZAHL ABGEB370
     2ROCHEN.'/' EINGELESEN WURDEN ALLE WERTE ZWISCHEN',E12.4,'EV UND ',380
     3E12.4,'EV')                                                       390
      MAX=I-1                                                           400
      WRITE(KOUT,609) (NAME(J),J=1,2),MAX                               410
  609 FORMAT(' ANZAHL DER VON KEDAK GEHOLTEN DATENPUNKTE FUER ',2A8,' = 420
     1',I5)                                                             430
      RETURN                                                            440
      END                                                               450
```

```
      SUBROUTINE PLO(X,Y,MAX,EL,MAT,NTI)                                10
      COMMON INPUT,KOUT                                                 20
      REAL*8 NSKB(11)/'-1.0..  ','-).8..  ','-0.6..  ','-0.4..  ','-0.2.30
     1.  ',' 0.0..  ','+0.2..  ','+0.4..  ','+0.6..  ','+0.8..  ','+1.0.40
     2.  '/,   SGNC/'SGNC    '/,    MAT                                 50
      DIMENSION XBS(11),YBS(11),NDIRSB(10),NSCSB(10),XBSDUM(2),YBSDUM(2) 60
      DIMENSION X(2),Y(2)                                               70
      DIMENSION XZ(404),YZ(404)                                         80
      DIMENSION NTEXT(15),XB(2),YB(2),NDIR(2),NSC(2),XT(2),YT(2),NTE(10) 90
      DIMENSION XR(200),YR(200)                                        100
      DIMENSION MAXN(10),ELA(10),NT(10),NP(10)                         110
      LOGICAL XLOG                                                     120
      DATA NUM/0/,ID/0/,YMAX,YMIN/0.,-2./,                             130
     1NLGY/+1/,YA,YE/-1.,+1./,NFX/'E9.2'/,NFY/'F5.1'/,NSCX,NSCY/1,1/,  140
     2 NUOX,NUOY/+1,+1/,NPSX,NPSY/-1,+1/,NX,NY/1,1/,NT/3/,             150
     3NPG/1/,IND/1/,NPA/1/,DX,DY/1.,.2/,ELA/0./,NTE(1)/'EN= '/,        160
     4 MN/0/,NP/0,1,2,3,4,5,6,7,8,9/,SY/2.54E-3/,NDIR/3,3/,NSC/1,1/    170
      DATA NSCSB/10*1/,NDIRSB/10*3/,NTSB/10/,NTSB1/1/                  180
      DATA TUSCHE/'T'/                                                 190
      DATA DYY/.2/                                                     200
C EINSPEICHERN DER ZU PLOTTENDEN DATEN NACH XZ,YZ                      210
```

```
      IF(MN+MAX.GT.404) GOTO 89                                        220
      NUM=NUM+1                                                        230
      DO 101 J=1,MAX                                                   240
      MN=MN+1                                                          250
      XZ(MN)=Y(J)                                                      260
  101 YZ(MN)=X(J)                                                      270
      ELA(NUM)=EL                                                      280
      MAXN(NUM)=MN                                                     290
      NT(NUM)=NTI                                                      300
      GOTO 99                                                          310
C                                                                      320
C ES WERDEN ALLE SEIT DEM LETZTEN PLOCL-AUFRUF DURCH PLO ANGELIEFERTEN 330
C               KURVEN GEZEICHNET.                                     340
C                                                                      350
      ENTRY PLOCL(PLOFOR)                                              360
      IF(NUM.LT.1) GOTO 99                                             370
      SY=2.54E-3                                                       380
      IF(PLOFOR.NE.TUSCHE) SY=2.5E-3                                   390
C                                                                      400
C TITEL UND NUMMER DER ZEICHNUNG                                       410
      CALL STRING( NTEXT(1),'    ')                                    420
      CALL STRING(NTEXT(2),NTEXT(1),56)                                430
      CALL STRING(NTEXT(1),MAT,8)                                      440
      CALL STRING(NTEXT(3),SGNC,8)                                     450
      ID=ID+1                                                          460
C                                                                      470
C PLOTGRENZEN UND PLOTART FESTLEGEN.                                   480
      XMAX=XZ(1)                                                       490
      XMIN=XZ(1)                                                       500
      DO 3 M=2,MN                                                      510
      IF(XZ(M).GT.XMAX) GOTO 2                                         520
      IF(XZ(M).GE.XMIN) GOTO 3                                         530
      XMIN=XZ(M)                                                       540
      GOTO 3                                                           550
    2 XMAX=XZ(M)                                                       560
    3 CONTINUE                                                         570
      IF(XMIN.NE.0) GOTO 4                                             580
      XLOG=.FALSE.                                                     590
      GOTO 6                                                           600
    4 IF(XMAX.GT.20.*XMIN) GOTO 5                                      610
      XLOG=.FALSE.                                                     620
      GOTO 6                                                           630
    5 XLOG=.TRUE.                                                      640
    6 IF(XLOG) GOTO 7                                                  650
      GOTO 30                                                          660
C                                                                      670
C ERSTER PLOT FUER LOGARITHMISCHE SIGMA-ACHSE.                         680
    7 XMAX=ALOG10(XMAX)                                                690
      XMIN=ALOG10(XMIN)                                                700
      XRES=.05*FLOAT(NUM+2)                                            710
      DO 9 M=1,MN                                                      720
      YZ(M)=YZ(M)-1.                                                   730
    9 XZ(M)=ALOG10(XZ(M))                                              740
      MA=INT(XMAX)                                                     750
      IF(FLOAT(MA).LT.XMAX) MA=MA+1                                    760
```

```
      M=INT(XMAX+XRES)
      IF(FLOAT(M).LT.XMAX+XRES) M=M+1
      MI=INT(XMIN)
      IF(FLOAT(MI).GT.XMIN) MI=MI-1
      NRES=INT(XMIN-XRES)
      IF(FLOAT(NRES).GT.XMIN-XRES) NRES=NRES+1
      IF(M.EQ.MA) GOTO 11
      IF(NRES.NE.MI) GOTO 10
      NRES=1
      GOTO 12
   10 MA=M
   11 NRES=2
      GOTO 12
   12 XMIN=FLOAT(MI)
      XMAX=FLOAT(MA)
      INDZ=INT((XMAX-XMIN)/2.5)+1
      XA=10.**MI
      XE=10.**MA
      XALOG=XMIN
      XELOG=XMAX
      SX=2.54E-3
      NLGX=-1
      CALL PLOTA(XZ,YZ,MAXN(1),NT(1),NP(1),NPG,IND,NPA,INDZ,XMAX,XMIN,
     1SX,YMAX,YMIN,SY,NTEXT,ID,
     2NLGX,XA,DX,XE,NFX,NSCX,NUOX,NPSX,NX,
     3NLGY,YA,DY,YE,NFY,NSCY,NUOY,NPSY,NY,0)
      INDZ=0
      CALL RASTER(-1,+1,XMAX,XMIN,SX,YMAX,YMIN,SY,INDZ,XR,YR,DX,DY)
      GOTO 20
C
C ERSTER PLOT IM FALL LINEARER SIGMA-ACHSE
   30 CALL GRIDLI(XMIN,XMAX,RMIN,RMAX,SX)
      IF(PLOFOR.NE.TUSCHE) SX=SX*2.5/2.54
      DX=20.*SX/.254
      IF(PLOFOR.NE.TUSCHE) DX=80.*SX
      XRES=FLOAT(NUM+2)*DX/4.
      NRES=INT(XRES/DX)+1
      XRES=FLOAT(NRES)*DX
      XMAX=RMAX+XRES
      NRES=2
      XMIN=RMIN
      INDZ=INT((XMAX-XMIN)/800./SX)+1
      NLGX=+1
      XA=XMIN
      XE=XMAX
      DO 32 M=1,MN
   32 YZ(M)=YZ(M)-1.
      CALL PLOTA(XZ,YZ,MAXN(1),NT(1),NP(1),NPG,IND,NPA,INDZ,XMAX,XMIN,
     1 SX,YMAX,YMIN,SY,NTEXT,ID,
     2 NLGX,XA,DX,XE,NFX,NSCX,NUOX,NPSX,NX,
     3 NLGY,YA,DY,YE,NFY,NSCY,NUOY,NPSY,NY,0)
      INDZ=0
      CALL RASTER(1,1,XMAX,XMIN,SX,YMAX,YMIN,SY,INDZ,XR,YR,DX,DY)
      GOTO 20
C
C BESCHRIFTUNG DER MUE-ACHSE
   20 XX=XMIN-24.*SX
      YBS(1)=YMIN+16.*SY
      DO 21 I=1,11
   21 XBS(I)=XX
      DO 22 I=2,11
   22 YBS(I)=YBS(I-1)+DYY
      XBSDUM(1)=XMIN-1.E+3*SX
      XBSDUM(2)=XBSDUM(1)
      YBSDUM(1)=YMIN-1.E+3*SY
      YBSDUM(2)=YBSDUM(1)
      CALL PLOTA(XBSDUM,YBSDUM,2,2,1,1,IND,1,INDZ,XMAX,XMIN,SX,YMAX,YMIN
     1,SY,1,ID,0,0,
     2NTSB,XBS,YBS,NDIRSB,NSCSB,NSKB(1),NSKB(2),NSKB(3),NSKB(4),NSKB(5),
     3NSKB(6),NSKB(7),NSKB(8),NSKB(9),NSKB(10))
      CALL PLOTA(XBSDUM,YBSDUM,2,1,1,1,IND,1,INDZ,XMAX,XMIN,SX,YMAX,YMIN
     1,SY,1,ID,0,0,
     2NTSB1,XBS(11),YBS(11),NDIRSB,NSCSB,NSKB(11))
C
C ZEICHENERKLAERUNG FUER DIE ERSTE KURVE
      IF(NRES.EQ.1) XT(1)=XMIN+XRES
      IF(NRES.EQ.2) XT(1)=XMAX-10.*SX
      XT(2)=XT(1)
      YT(1)=YMAX-60.*SY
      YT(2)=YT(1)
      XB(1)=XT(1)-8.*SX
      YB(1)=YT(1)-25.*SY
      CALL STRING(NTE(2),'    @')
      CALL STRING(NTE(3),NTE(2),32)
      CALL STRING(NTE(5),'  EV@')
      CALL STRING(NTE(6),'..  @')
      CALL CONVX(ELA(1),NTE(2),'E12.5')
      CALL PLOTA(XT,YT,2,1,NP(1),NPG,IND,NPA,INDZ,XMAX,XMIN,SX,
     1 YMAX,YMIN,SY,1,ID,0,0,1,
     2 XB,YB,NDIR,NSC,NTE)
C
C JETZT WERDEN DIE UEBRIGEN KURVEN GEPLOTTET ZUSAMMEN MIT
C              ZEICHENERKLAERUNG.
      IF(NUM.LT.2) GOTO 90
      DO 25 N=2,NUM
      M=MAXN(N-1)+1
      LMAX=MAXN(N)-MAXN(N-1)
      CALL PLOTA(XZ(M),YZ(M),LMAX,NT(N),NP(N),NPG,IND,NPA,INDZ,
     1 XMAX,XMIN,SX,YMAX,YMIN,SY,1,ID,0,0,0)
      XT(1)=XT(1)-20.*SX
      XT(2)=XT(1)
      XB(1)=XT(1)-8.*SX
      YB(1)=YT(1)-25.*SY
      NDIR(1)=3
      NSC(1)=1
      CALL STRING(NTE(2),'    @')
      CALL STRING(NTE(3),NTE(2),32)
      CALL STRING(NTE(5),'  EV@')
      CALL STRING(NTE(6),'..@')
      CALL CONVX(ELA(N),NTE(2),'E12.5')
```

```
      CALL PLOTA(XT,YT,2,1,NP(N)     ,NPG,IND,NPA,INDZ,XMAX,XMIN,SX,      1870
     1YMAX,YMIN,SY,     1,ID,     0,0,1,XB,YB,NDIR,NSC,NTF)                 1880
   25 CONTINUE                                                             1890
      GOTO 90                                                              1900
   89 WRITE(KOUT,689)                                                      1910
  689 FORMAT(/' DIE ANGELIEFERTEN DATEN FINDEN IM PLOTFELD KEINEN PLATZ    1920
     1'/)                                                                  1930
      GOTO 99                                                              1940
   90 MN=0                                                                 1950
      NUM=0                                                                1960
      GOTO 99                                                              1970
   99 RETURN                                                               1980
      END                                                                  1990
```

```
      SUBROUTINE GRIDLI(XMIN,XMAX,RMIN,RMAX,SX)                             10
C                                                                           20
C GRIDLI BESTIMMT DIE ACHSEN-KENNGROESSEN FUER LINEARE ACHSEN SO,           30
C              DASS DEREN LAENGE 10-20CM BETRAEGT.                          40
      XLX=ALOG10(XMAX)                                                      50
      LX=INT(XLX)-1                                                         60
      IF(XLX.LT.0.) LX=LX-1                                                 70
      EH=10.**LX                                                            80
      IMAX=INT(XMAX/EH)                                                     90
      IF(XMAX.GT.(FLOAT(IMAX)*EH)) IMAX=IMAX+1                             100
      IMIN=INT(XMIN/EH)                                                    110
    2 LSKAL=IMAX-IMIN                                                      120
      IF(LSKAL.LE.100) GOTO 4                                              130
      IMIN=IMIN/10                                                         140
      IM=IMAX/10                                                           150
      IF((10*IM).LT.IMAX) IM=IM+1                                          160
      IMAX=IM                                                              170
      EH=EH*10.                                                            180
      GOTO 2                                                               190
    4 IF(LSKAL.NE.0) GOTO 10                                               200
      IMAX=IMAX+50                                                         210
      IMIN=IMIN-50                                                         220
   10 IF(LSKAL.LE.50) GOTO 12                                              230
      ISTART=(IMIN/10)*10                                                  240
      IEND=(IMAX/10)*10                                                    250
      IF(IEND.LT.IMAX) IEND=IEND+10                                        260
      SX=.5*EH*.254                                                        270
      GOTO 20                                                              280
   12 IF(LSKAL.LE.40) GOTO 14                                              290
      ISTART=(IMIN/5)*5                                                    300
      IEND=(IMAX/5)*5                                                      310
      IF(IEND.LT.IMAX) IEND=IEND+5                                         320
      SX=.25*EH*.254                                                       330
      GOTO 20                                                              340
   14 IF(LSKAL.LE.20) GOTO 16                                              350
      ISTART=(IMIN/4)*4                                                    360
      IEND=(IMAX/4)*4                                                      370
      IF(IEND.LT.IMAX) IEND=IEND+4                                         380
      SX=.2*EH*.254                                                        390
      GOTO 20                                                              400
   16 IF(LSKAL.LE.10) GOTO 18                                              410
      ISTART=(IMIN/2)*2                                                    420
      IEND=(IMAX/2)*2                                                      430
      IF(IEND.LT.IMAX) IEND=IEND+2                                         440
      SX=.1*EH*.254                                                        450
      GOTO 20                                                              460
   18 IMIN=IMIN*10                                                         470
      IMAX=IMAX*10                                                         480
      EH=EH/10.                                                            490
      LSKAL=IMAX-IMIN                                                      500
      GOTO 10                                                              510
   20 RMIN=FLOAT(ISTART)*EH                                                520
      RMAX=FLOAT(IEND)*EH                                                  530
      RETURN                                                               540
      END                                                                  550
```

```
      SUBROUTINE INTGRT(X,F,MAX,S)                                          10
      DIMENSION X(2),F(2)                                                   20
      S=0.                                                                  30
      DO 10 I=2,MAX                                                         40
   10 S=S+(F(I)+F(I-1))*(X(I)-X(I-1))                                       50
      S=S*.5                                                                60
      RETURN                                                                70
      END                                                                   80
```

```
      SUBROUTINE CTOL(COSC,DSIGC,MAX,COSL,DSIGL,     LMAX,LLMAX,A)          10
      DIMENSION COSC(1),DSIGC(1),COSL(1),DSIGL(1)                           20
      A22=A**2                                                              30
      A2=2.*A                                                               40
      IF(A.LT.2.AND.A.GT.1.) GOTO 80                                        50
      MA=1                                                                  60
      L=1                                                                   70
      MMAX=MAX-1                                                            80
      LD=(LLMAX-1)/MMAX                                                     90
    8 DO 12 M=MA,MMAX                                                      100
      COSL(L)=COSC(M)                                                      110
      DSIGL(L)=DSIGC(M)                                                    120
      DC=(COSC(M+1)-COSC(M))/FLOAT(LD)                                     130
      DS=(DSIGC(M+1)-DSIGC(M))/FLOAT(LD)                                   140
      LA=L+1                                                               150
      LE=L+LD-1                                                            160
      L=L+LD                                                               170
      DO 10 LL=LA,LE                                                       180
      COSL(LL)=COSL(LL-1)+DC                                               190
   10 DSIGL(LL)=DSIGL(LL-1)+DS                                             200
   12 CONTINUE                                                             210
   14 LMAX=LE+1                                                            220
```

```
      COSL(LMAX)=COSC(MAX)                                                    230
      DSIGL(LMAX)=DSIGC(MAX)                                                  240
      IF(A.LE.1.) GOTO 30                                                     250
      W=A-1.                                                                  260
      DC=-1.                                                                  270
      DSIGL(1)=W*DSIGL(1)/(A*(1.-DC/W))                                       280
      COSL(1)=DC                                                              290
      LA=2                                                                    300
   19 DO 20 L=LA,LMAX                                                         310
      DC=COSL(L)                                                              320
      W=SQRT(A22+A2*DC+1.)                                                    330
      DC=(A*DC+1.)/W                                                          340
      DSIGL(L)=W*DSIGL(L)/(A*(1.-DC/W))                                       350
   20 COSL(L)=DC                                                              360
      GOTO 100                                                                370
   30 IF(A.LT.1.) GOTO 40                                                     380
      COSL(1)=0.                                                              390
      DSIGL(1)=0.                                                             400
      LA=2                                                                    410
      GOTO 19                                                                 420
C                                                                             430
C         TO EACH MU(L) THERE EXIST TWO MU(C).                                440
C         PROCEEDING: CALCULATE MU(L) TO GIVEN MU(C).FIND SECOND MU(C)        450
C         CORRESPONDING TO THIS MU(L).SUM CONTRIBUTIONS FROM BOTH.            460
   40 DO 42 L=2,LMAX                                                          470
      IF(COSL(L).LE.-A) GOTO 42                                               480
      LA=L                                                                    490
      GOTO 43                                                                 500
   42 CONTINUE                                                                510
   43 DS=COSL(LA)                                                             520
      DC=COSL(LA-1)                                                           530
C                                                                             540
C         CALCULATE DSIGMA IN L-SYSTEM FOR LOWEST INTERVAL OF MU(L)           550
      DC=(-.5*((DSIGL(LA)-DSIGL(LA-1))/(DS-DC)*(A+DS))+DSIGL(LA))*(A+DS)      560
      W=SQRT(A22+A2*DS+1.)                                                    570
      DS1=(A*DS+1.)/W                                                         580
      W1=SQRT(A22+DS1**2-1.)                                                  590
      W=(DS1**2-DS1*W1-1.)/A                                                  600
      DO 45 L=1,LMAX                                                          610
      IF(COSL(L).LE.W) GOTO 45                                                620
      LE=L                                                                    630
      GOTO 46                                                                 640
   45 CONTINUE                                                                650
   46 DS=COSL(LE)                                                             660
      DS2=COSL(LE-1)                                                          670
      DC=DC+(0.5*((DSIGL(LE)-DSIGL(LE-1))/(DS-DS2)*(W-DS))+DSIGL(LE))*(D      680
     1S-W)                                                                    690
      DS2=SQRT(1.-A22)                                                        700
      COSL(LA)=DS1                                                            710
      DS=DC/(DS1-DS2)                                                         720
      DC=DS2                                                                  730
C                                                                             740
C         DO THE OTHER INTERVALS                                              750
      LE=LE-1                                                                 760
      DSIGL(LA)=(DSIGL(LA)*(2.*DS1+W1+DS1**2/W1)-DSIGL(LE)*(2.*DS1-W1-DS      770
     11**2/W1))/A                                                             780
      LA=LA+1                                                                 790
      IF(LA.GT.LMAX) GOTO 70                                                  800
      DO 60 L=LA,LMAX                                                         810
      DS1=COSL(L)                                                             820
      W=SQRT(A22+A2*DS1+1.)                                                   830
      DS1=(1.+A*DS1)/W                                                        840
      W1=SQRT(A22+DS1**2-1.)                                                  850
      W=(DS1**2-DS1*W1-1.)/A                                                  860
   51 IF(COSL(LE).LE.W) GOTO 52                                               870
      LE=LE-1                                                                 880
      GOTO 51                                                                 890
   52 W=DS1**2/W1                                                             900
      DS2=2.*DS1                                                              910
      DSIGL(L)=(DSIGL(L)*(DS2+W1+W)-DSIGL(LE)*(DS2-W1-W))/A                   920
      COSL(L)=DS1                                                             930
   60 CONTINUE                                                                940
   70 LA=LA-1                                                                 950
      DSIGL(LA)=DS                                                            960
      COSL(LA)=DC                                                             970
      DO 72 L=LA,LMAX                                                         980
      J=L+1-LA                                                                990
      DSIGL(J)=DSIGL(L)                                                      1000
   72 COSL(J)=COSL(L)                                                        1010
      LMAX=LMAX+1-LA                                                         1020
      GOTO 100                                                               1030
   80 W=1./A                                                                 1040
      LE=LLMAX/4                                                             1050
      DC=(1-W)/(LE-1)                                                        1060
      DO 81 L=1,LE                                                           1070
   81 COSL(L)=(L-1)*DC-1.                                                    1080
      LA=2                                                                   1090
      DO 83 L=1,LE                                                           1100
   82 IF(COSL(L).LT.COSC(LA)) GOTO 83                                        1110
      LA=LA+1                                                                1120
      GOTO 82                                                                1130
   83 DSIGL(L)=(DSIGC(LA)-DSIGC(LA-1))/(COSC(LA)-COSC(LA-1))*(COSC(LA)-C     1140
     ZOSL(L))+DSIGC(LA)                                                      1150
      DS2=DSIGC(LA)                                                          1160
      DC2=COSC(LA)                                                           1170
      DS1=DSIGC(LA-1)                                                        1180
      DC1=COSC(LA-1)                                                         1190
      W1=W+DC2                                                               1200
      DC=0.001                                                               1210
   85 MA=W1/DC                                                               1220
      IF(MA.LE.LLMAX/10) GOTO 86                                             1230
      DC=2.*DC                                                               1240
      GOTO 85                                                                1250
   86 L=LE+1                                                                 1260
      DS=COSL(LE)                                                            1270
   87 COSL(L)=DS+(L-LE)*DC                                                   1280
      IF(COSL(L).GE.0.999*DC2) GOTO 89                                       1290
      DSIGL(L)=(DS2-DS1)/(DC2-DC1)*(DC2-COSL(L))+DS2                         1300
      L=L+1                                                                  1310
      GOTO 87                                                                1320
```

```
 89 MA=LA                                                              1330
    MMAX=MAX-1                                                         1340
    IF(MAX.EQ.MA) GOTO 14                                              1350
    LD=(LLMAX-L)/(MAX-MA)                                              1360
    GOTO 8                                                             1370
100 RETURN                                                             1380
    END                                                                1390
```

```
    SUBROUTINE OUTPUT(MAT,EL,X,Y,MAX)                                    10
    COMMON INPUT,KOUT,ITAP,ITAB                                          20
    DIMENSION X(21),Y(21),NM(3)                                          30
    REAL*8 ADD/'ADD     '/,NAME(3)                                       40
    REAL*8 SGNC/'SGNC    '/,MAT                                          50
    DATA NM/3,1,1/                                                       60
    NUM=10+2*MAX                                                         70
    NALP=2                                                               80
    NNAM=NM(1)                                                           90
    NARG=NM(2)                                                          100
    NWERT=NM(3)                                                         110
    NAME(1)=MAT                                                         120
    NAME(2)=SGNC                                                        130
    NAME(3)=DBLE(EL)                                                    140
    WRITE(3,600) ITAB,NUM,ADD,NNAM,(NAME(I),I=1,NALP),EL,NARG,          150
   1NWERT,(X(I),Y(I),I=1,MAX)                                           160
600 FORMAT( '  OUTPUT WRITTEN ON UNIT',I3,/                           170
   1              I3,1X,A8,I2,1X,2A8,E12.4,1X,I2,I2/(7(F8.4)))         180
    WRITE(ITAB) NUM,ADD,NNAM,(NAME(I),I=1,NALP),EL,NARG,NWERT,          190
   1(X(I),Y(I),I=1,MAX)                                                 200
    RETURN                                                              210
    END                                                                 220
```

```
    SUBROUTINE LOCDEL(MAT,ENERGY,X,Y,MAX,*,*)                            10
    REAL*8 NAME(3),                    SGNC/'SGNC    '/, ENGOLD, MAT    20
    DIMENSION NARG(3),ARG(2),X(MAX),Y(MAX)                               30
    COMMON INPUT,KOUT                                                    40
    NARG(1)=3                                                            50
    NAME(1)=MAT                                                          60
    NAME(2)=SGNC                                                         70
    NAME(3)=DBLE(ENERGY)                                                 80
    CALL LDFLOC(NERR,NARG,NAME,ARG)                                      90
    CALL LDFLOC(NERR,NARG,NAME,ARG)                                     100
    IF(NERR.EQ.0) GOTO 91                                               110
  8 X(1)=ARG(1)                                                         120
    Y(1)=ARG(2)                                                         130
    J=2                                                                 140
    ENGOLD=NAME(3)                                                      150
    ENERGY=SNGL(ENGOLD)                                                 160
  9 CALL LDFNXT(NERR,NARG,NAME,ARG)                                     170
    IF(NERR.EQ.0) GOTO 20                                               180
    X(J)=ARG(1)                                                         190
    Y(J)=ARG(2)                                                         200
    J=J+1                                                               210
    GOTO 9                                                              220
    ENTRY GETDEL(ENERGY,X,Y,MAX,*,*)                                    230
    CALL LDFLOC(NERR,NARG,NAME,ARG)                                     240
    IF(NERR.EQ.0) GOTO 92                                               250
    IF(ENGOLD.NE.NAME(3)) GOTO 8                                        260
    WRITE(KOUT,610) MAT,NAME(3)                                         270
610 FORMAT(' LETZTER SGNC-SATZ AUF KEDAK FUER ',A8,' IST',1PE14.5)       280
    RETURN 1                                                            290
 20 MK=J-1                                                              300
    IF(MAX.EQ.0) GOTO 22                                                310
    IF(J-1.EQ.MAX) GOTO 22                                              320
    WRITE(KOUT,620) NAME,MAX,MK                                         330
620 FORMAT(' ANZAHL DER ERWARTETEN SAETZE FUER ',2A8,D12.4/              340
   1' = ',I3,' GEFUNDEN WURDEN ',I3)                                    350
 22 MAX=MK                                                              360
    RETURN                                                              370
 91 WRITE(KOUT,691) NAME                                                380
691 FORMAT(' LDFLOC-AUFRUF BLIEB ERFOLGLOS FUER ',2A8,D12.4)            390
    RETURN 2                                                            400
 92 WRITE(KOUT,692)                                                     410
692 FORMAT(' PROGRAMMIERER RUFEN ')                                     420
    RETURN 2                                                            430
    END                                                                 440
```

```
    SUBROUTINE GETTAP(MAT,ENERGY,X,Y,MJ,EUNT,EOB,NUMKED,LENX,*)          10
    COMMON INPUT,KOUT,ITAP,ITAB                                          20
    DIMENSION X(2),Y(2)                                                  30
    REAL*8 NAME(3),MAT,SGNC/'SGNC    '/,OPT,ADD/'ADD     '/           40
    REAL*8 MATOLD/'         '/                                           50
    LOGICAL PRINT/F/,RD/F/                                               60
    DATA NAME/'         ','         ','         '/                      70
    DATA TEND/'ENDE'/                                                    80
    DIMENSION W(212),M(1),NARG(3)                                        90
    EQUIVALENCE(W(1),M(1))                                              100
C   WRITE(6,700) MAT,EUNT,EOB                                           110
C700 FORMAT(/' GETTAP MAT=',A8,' EUNT=',E13.6,'  EOB=',E13.6)           120
    M1=2*NUMKED+10                                                      130
    IF(MATOLD.EQ.MAT) GOTO 2                                            140
    RD=.FALSE.                                                          150
    MATOLD=MAT                                                          160
    IF(NAME(1).NE.MAT) GOTO 2                                           170
    IF(NAME(2).NE.SGNC) GOTO 2                                          180
    GOTO 4                                                              190
  2 READ(8,END=10) NUM,(W(I),I=1,NUM)                                   200
C   WRITE(6,701) (W(I),I=1,11)                                          210
C701 FORMAT('  NEUES RECORD GELESEN ',11(Z8,1X))                        220
    IF(NUMKED.EQ.0) GOTO 3                                              230
    IF(NUM.NE.M1) GOTO 13                                               240
  3 M2=(NUM-10)/2                                                       250
```

```
      IF(M2.GT.LENX) GOTO 19                                                    260
      NARG(1)=M(3)                                                              270
      J=M(3)+6                                                                  280
      NARG(2)=M(J)                                                              290
      NARG(3)=M(J+1)                                                            300
      CALL STRING(NAME(1),W(4),16)                                              310
    4 CALL STRING(ENERGY,W(8),4)                                                320
      CALL STRING(OPT,W(1),8)                                                   330
      NAME(3)=DBLE(ENERGY)                                                      340
      IF(OPT.NE.ADD) GOTO 12                                                    350
      IF(NAME(1).NE.MAT) GOTO 11                                                360
      IF(NAME(2).NE.SGNC) GOTO 11                                               370
      RD=.TRUE.                                                                 380
      IF(ENERGY.LT.EUNT) GOTO 2                                                 390
      IF(ENERGY.GT.EOB) GOTO 17                                                 400
      J=M(3)+8                                                                  410
      MJ=0                                                                      420
      DO 6 I=J,NUM,2                                                            430
      MJ=MJ+1                                                                   440
      X(MJ)=W(I)                                                                450
    6 Y(MJ)=W(I+1)                                                              460
      IF(PRINT) WRITE(KOUT,600) NARG,NAME,(X(I),Y(I),I=1,MJ)                    470
  600 FORMAT(' NARG=',3I2,'   NAMES=',2A8,D12.4/(5(F10.2,F12.4)))               480
      RETURN                                                                    490
   10 WRITE(KOUT,604) ITAP                                                      500
  604 FORMAT(' ENDE DER DATEN AUF FT',I2,' (8) ERREICHT')                       510
   14 RETURN 1                                                                  520
   11 IF(.NOT.RD) GOTO 2                                                        530
      WRITE(KOUT,601) NAME                                                      540
  601 FORMAT('  NAMEN STIMMEN MIT DEN VORGEGEBENEN NICHT UEBEREIN.'/            550
     1'   SIE LAUTEN:',2A8,D12.4)                                               560
      GOTO 14                                                                   570
   12 WRITE(KOUT,602) OPT                                                       580
  602 FORMAT(' OPTION=',A8,' NICHT GLEICH ADD')                                 590
      GOTO 10                                                                   600
   13 IF(NUM.EQ.2) GOTO 15                                                      610
   16 WRITE(KOUT,603) NUM,M1                                                    620
  603 FORMAT(' ANZAHL DER WORTE AUF DEM BAND =',I3,' UNGLEICH ',I3)             630
      GOTO 10                                                                   640
   15 IF(W(1).NE.TEND) GOTO 16                                                  650
      WRITE(KOUT,605) ITAP                                                      660
  605 FORMAT('1 ENDE-RECORD AUF BAND',I2,' ERREICHT')                           670
      GOTO 14                                                                   680
   17 WRITE(KOUT,606) ITAP,EUNT,EOB                                             690
  606 FORMAT(' ENDE DER DATEN AUF FT',I2,' ZWISCHEN EMIN=',E13.6,'EV            700
     1UND EMAX=',E13.6,'EV')                                                    710
      GOTO 14                                                                   720
   19 WRITE(KOUT,607) NUM,LENX                                                  730
  607 FORMAT('   ANZAHL DER DATENPAARE IM SATZ =',I3,' UEBERSTEIGT DATEN        740
     1FELDLAENGE =',I3)                                                         750
      GOTO 14                                                                   760
      END                                                                       770
```

```
      SUBROUTINE RASTER(KREX,KREY,XMAX,XMIN,SX,YMAX,YMIN,SY,INDZ,X,Y,DX,     10
     1 DY)                                                                      20
      DIMENSION X(2),Y(2)                                                       30
      DATA NRAST/9999/                                                          40
      IF(INDZ.NE.0.AND.INDZ.NE.10) GOTO 50                                      50
      IF(KREY.EQ.-1) GOTO 10                                                    60
      CALL RALIN(YMIN,YMAX,DY,XMIN,XMAX,M,X,Y)                                  70
      GOTO 20                                                                   80
   10 CALL RALOG(YMIN,YMAX,SY,XMIN,XMAX,M,X,Y)                                  90
   20 CALL PLOTA(X    ,Y    ,M,2,0,1,1,1,INDZ,XMAX,XMIN,SX,YMAX,YMIN,SY,1,    100
     1NRAST,0,0,0)                                                             110
      IF(KREX.EQ.-1) GOTO 30                                                    120
      CALL RALIN(XMIN,XMAX,DX,YMIN,YMAX,M,Y,X)                                  130
      GOTO 40                                                                   140
   30 CALL RALOG(XMIN,XMAX,SX,YMIN,YMAX,M,Y,X)                                  150
   40 CALL PLOTA(X    ,Y    ,M,2,0,1,1,1,INDZ,XMAX,XMIN,SX,YMAX,YMIN,SY,1,    160
     1NRAST,0,0,0)                                                             170
   50 RETURN                                                                    180
      END                                                                       190
```

```
      SUBROUTINE RALIN(YMIN,YMAX,DY,XMIN,XMAX,NPOINT,X,Y)                       10
      DIMENSION X(2),Y(2)                                                       20
C                                                                               30
C  DIESE VERSION ERZEUGT GITTERLINIEN IM ABSTAND DY                             40
C                                                                               50
      YL=YMAX-YMIN                                                              60
      NLIN=INT(YL/DY)                                                           70
      IF((YL-FLOAT(NLIN)*DY).GT.DY/2.) NLIN=NLIN+1                              80
      NLIN=NLIN+1                                                               90
      NPOINT=2*NLIN                                                            100
      IF(NPOINT.GT.200) NPOINT=200                                             110
      Y1=YMIN                                                                  120
      Y2=YMIN+DY                                                               130
      DO 10 I=1,NPOINT,4                                                       140
      X(I)=XMIN                                                                150
      X(I+1)=XMAX                                                              160
      X(I+2)=XMAX                                                              170
      X(I+3)=XMIN                                                              180
      Y(I)=Y1                                                                  190
      Y(I+1)=Y1                                                                200
      Y(I+2)=Y2                                                                210
      Y(I+3)=Y2                                                                220
      Y1=Y2+DY                                                                 230
      Y2=Y1+DY                                                                 240
   10 CONTINUE                                                                 250
      RETURN                                                                   260
      END                                                                      270
```

```
      SUBROUTINE RALOG(YMIN,YMAX,SY,XMIN,XMAX,NPOINT,X,Y)                10
      DIMENSION X(2),Y(2),DYL(9)                                         20
      DATA DYL/.0458,.3010,.1761,.1250,.0969,.0792,.0669,.0580,.0511/    30
      YL=YMAX-YMIN                                                       40
      NDEK=INT(YL)                                                       50
      IF((YL-FLOAT(NDEK)).GT.0.8) NDEK=NDEK+1                            60
      IF(NDEK.GT.6) GOTO 25                                              70
      NLIN=9*NDEK+1                                                      80
      NPOINT=2*NLIN                                                      90
      J=0                                                               100
      Y1=YMIN                                                           110
      Y2=YMIN+DYL(2)                                                    120
      DO 20 I=1,NPOINT,4                                                130
      X(I)=XMIN                                                         140
      X(I+1)=XMAX                                                       150
      X(I+2)=XMAX                                                       160
      X(I+3)=XMIN                                                       170
      Y(I)=Y1                                                           180
      Y(I+1)=Y1                                                         190
      Y(I+2)=Y2                                                         200
      Y(I+3)=Y2                                                         210
      J=J+2                                                             220
      Y1=Y2+DYL(MOD(J,9)+1)                                             230
      Y2=Y1+DYL(MOD(J+1,9)+1)                                           240
   20 CONTINUE                                                          250
   25 RETURN                                                            260
      END                                                               270
```

```
      SUBROUTINE GETCD(X,Y,NX,EL,ZZ,LENX,DRUCK,*)                        10
       LOGICAL*1 DRUCK                                                   20
      DIMENSION   X(1),Y(1),ZZ(1)                                        30
      DIMENSION TXT(20),COSTAB(101)                                      40
      COMMON INPUT,KOUT                                                  50
      REAL*8 MODE/'LEG'/,LEG/'LEG'/,TAB/'TAB'/,NEWTAB/'NO'/,FMT(10)      60
      REAL*4 LEGCOF(20)                                                  70
      NAMELIST/CD/ MODE,ENERGY,NUMLC,NEWTAB,NUMCOS,                      80
     1LEGCOF,COSTAB                                                      90
      DATA FMT/'(8F10.2)',9*' '/,TCD/' $CD'/,TIN/' $IN'/,TEND/'ENDE'/   100
   10 READ(INPUT,501) TXT                                               110
      IF(TXT(1).NE.TCD) GOTO 300                                        120
      BACKSPACE INPUT                                                   130
      READ(INPUT,CD)                                                    140
      WRITE(KOUT,601) ENERGY,MODE,NUMLC,NEWTAB,NUMCOS                   150
      EL=ENERGY                                                         160
      IF(MODE.EQ.LEG) GOTO 100                                          170
      IF(MODE.EQ.TAB) GOTO 200                                          180
      WRITE(KOUT,600)                                                   190
      GOTO 999                                                          200
  100 CALL GETLEG(X,Y,NX,NUMCOS,NUMLC,NEWTAB,ZZ,LENX,DRUCK,LEGCOF,      210
     1COSTAB)                                                           220
      GOTO 999                                                          230
  200 CALL GETTAB                                                       240
      GOTO 999                                                          250
  300 IF(TXT(1).EQ.TIN.OR.TXT(1).EQ.TEND) GOTO 310                      260
      CALL ERMSG(TCD,TXT)                                               270
      GOTO 10                                                           280
  310 BACKSPACE INPUT                                                   290
      RETURN 1                                                          300
  999 RETURN                                                            310
  501 FORMAT(20A4)                                                      320
  600 FORMAT(/' ***** MODE CAN ONLY BE TAB OR LEG'/)                    330
  601 FORMAT(/'   ENERGY=',E14.6,'   MODE=',A8,'   NUMLC=',I6,'   NEWTA  340
     1B=',A8,'   NUMCOS=',I6/)                                          350
      END                                                               360
```

```
      SUBROUTINE GETTAB                                                  10
      COMMON INPUT,KOUT                                                  20
      WRITE(KOUT,600)                                                    30
  600 FORMAT(//' ***** NO TABULATED INPUT VIA CARDS ALLOWED'//)          40
      RETURN                                                             50
      END                                                                60
```

```
      SUBROUTINE GETLEG(X,Y,NX,LTAB,ND,NEWTAB,ZZ,LENX,DRUCK,CL,CT)       10
C                                                                        20
C GETLEG CONVERTS LEGENDRE COEFFICIENT INTO   TABULATED DATA             30
C                                                                        40
      DIMENSION X(1),Y(1),ZZ(1),CL(20),CT(1)                             50
      LOGICAL*1 DRUCK                                                    60
      REAL*8 NEWTAB,EQUI/'EQUI'/,RED/'READ'/,NO/'NO'/,FMT(1)             70
      COMMON INPUT,KOUT                                                  80
      MAXCL=20                                                           90
      IF(ND.GT.MAXCL) GOTO 48                                           100
    9 IF(NEWTAB.EQ.NO) GOTO 50                                          110
      IF(LTAB.GT.LENX) GOTO 49                                          120
   10 IF(NEWTAB.EQ.EQUI) GOTO 30                                        130
      IF(NEWTAB.EQ.RED) GOTO 20                                         140
      GOTO 47                                                           150
   20 DO 21 I=1,LTAB                                                    160
   21 X(I)=CT(I)                                                        170
      WRITE(KOUT,500) (X(I),I=1,LTAB)                                   180
  500 FORMAT('NEW COSTAB READ IS',16F5.2)                               190
      NX=LTAB                                                           200
      GOTO 50                                                           210
   30 X(1)=-1.                                                          220
      DX=LTAB-1                                                         230
      DX=2./DX                                                          240
      DO 32 I=2,LTAB                                                    250
   32 X(I)=X(I-1)+DX                                                    260
      X(LTAB)=+1.                                                       270
      NX=LTAB                                                           280
      GOTO 50                                                           290
```

```
   47 WRITE(KOUT,600)
  600 FORMAT(/' ***** NEWTAB MUST BE = EQUI OR = READ'/)
      NX=0
      GOTO 999
   48 WRITE(KOUT,601) MAXCL
  601 FORMAT(/' *****NUMDAT MUST BE LESS THAN  ',I3/)
      ND=MAXCL
      GOTO 9
   49 WRITE(KOUT,602) LENX
  602 FORMAT(/' ***** LENGTH OF TABULATION MUST NOT EXCEED ',I4/)
      NX=LENX
      GOTO 10
   50 CONTINUE
      IF(.NOT.DRUCK) GOTO 58
      WRITE(KOUT,603)
  603 FORMAT(/' LEGENDRE COEFF. IN INPUT'/)
      WRITE(KOUT,604) (I,CL(I),I=1,ND)
  604 FORMAT((5(I9,E14.6)))
   58 DO 60 I=1,NX
   60 CALL SUMLEG(Y(I),X(I),CL,ND-1)
      IF(.NOT.DRUCK) GOTO 68
      WRITE(KOUT,605)
  605 FORMAT(/'  TABULATED DISTRIBUTION:'/)
      WRITE(KOUT,606) (X(I),Y(I),I=1,NX)
  606 FORMAT((5(5X,F6.3,E14.6)))
   68 CALL CONTRL(X,Y,NX,CL,ND,ZZ)
  999 RETURN
      END
```

```
      SUBROUTINE CONTRL(X,Y,NX,CL,ND,ZZ)
C
C        CONTRL CHECKS CONVERTED DISTRIBUTION.
C       IT CALCULATES PO AND P1 AND COMPARES IT WITH GIVEN LEGCOEFF.
C
      DIMENSION X(1),Y(1),CL(1),ZZ(1)
      DATA PI/3.14159265/,KOUT/6/
      P1=0.
      IF(ND.LT.2) GOTO 20
      DO 10 I=1,NX
   10 ZZ(I)=X(I)*Y(I)
      CALL INTGRT(X,ZZ,NX,P1)
   20 CALL INTGRT(X,Y,NX,PO)
      P1=P1/PO
      IF(ABS(2.*CL(1)-PO).LT.1.E-4*PO) GOTO 30
      PO=2.*PI*PO
      POS=4.*PI*CL(1)
      WRITE(KOUT,600) POS,PO
  600 FORMAT(/' ***** FROM LEG SIGMA INTEGRATED=',E14.6,'   FROM TAB SIG
     1MA INTEGRATED=',E14.6)
      WRITE(KOUT,601)
  601 FORMAT(/10X' IF YOU WISH TO IMPROVE AGREEMENT INCREASE NUMBER OF P
     1OINTS IN TABULATION'/)
```

```
   30 IF(ND.LT.2) GOTO 999
      POS=CL(2)/3./CL(1)
      IF(ABS(POS-P1).LT.1.E-4) GOTO 999
      WRITE(KOUT,602) POS,P1
  602 FORMAT(/' ***** FROM LEGCOEFF. MUEL(CM.)=',E14.6,'   FROM TABULATI
     1ON MUEL(CM.)=',E14.6)
      WRITE(KOUT,601)
  999 RETURN
      END
```

```
      SUBROUTINE ERMSG(T1,T2)
C
C
C  PRINTS ERRORMESSAGE.
C
      COMMON INPUT,KOUT
      DIMENSION T2(20)
      WRITE(KOUT,601) T1,T2
  601 FORMAT(/' ***** ',A4,' EXPECTED.FOUND :',4X,'*',20A4,'*'/)
      RETURN
      END
```

```
      SUBROUTINE SUMLEG(Y,X,C,N)
C
C SUMLEG BERECHNET DEN WERT EINER LEGENDRE-ENTWICKLUNG Y AN DER STELLE X
C         C(1) BIS C(N+1) SIND DIE KOEFFIZIENTEN DER ENTWICKLUNG NACH
C         P0 BIS PN.
C
      DIMENSION C(1)
      IF(N.LT.0) GOTO 9
      Y=C(1)
      IF(N.GE.1) GOTO 1
      GOTO 9
    1 Y=Y+C(2)*X
      IF(N.GE.2) GOTO 2
      GOTO 9
    2 Y1=1.
      Y2=X
      DO 3 I=2,N
      Y0=Y1
      Y1=Y2
      Y2=X*Y1
      Y2=Y2-Y0+Y2-(Y2-Y0)/FLOAT(I)
    3 Y=Y+C(I+1)*Y2
    9 RETURN
      END
```

```
      SUBROUTINE KEDOPN(K,*)                                            10
      IF(K.NE.0) GOTO 100                                               20
      K=1                                                               30
      CALL LOFOPN(1,IDATUM,&200)                                        40
  100 RETURN                                                            50
  200 RETURN1                                                           60
      END                                                               70
```

```
      SUBROUTINE CDINPT(KIN)                                            10
      DIMENSION A(20),TEND(20)                                          20
      DATA TTEND/'ENDE'/,BLANK/'    '/                                  30
      COMMON INPUT,KOUT                                                 40
      TEND(1)=TTEND                                                     50
      DO 100 I=2,20                                                     60
  100 TEND(I)=BLANK                                                     70
      N=1                                                               80
 1000 READ(KIN,500,END=99,ERR=9) A                                      90
      WRITE(INPUT,500) A                                               100
  500 FORMAT(20A4)                                                     110
      WRITE(KOUT,501)A                                                 120
  501 FORMAT(10X,20A4)                                                 130
      N=N+1                                                            140
      GOTO 1000                                                        150
    9 WRITE(KOUT,601) N                                                160
  601 FORMAT(/' FEHLER BEIM LESEN DER',I5,'-TEN KARTE'/)               170
      GOTO 1000                                                        180
   99 WRITE(INPUT,500) TEND                                            190
      REWIND INPUT                                                     200
      WRITE(KOUT,602) INPUT,N                                          210
  602 FORMAT(/' DER DATASET AUF G.FT',I2,' BESTEHT AUS',I5,'  RECORDS.'/ 220
     15X,'ALS LETZTES RECORD IST EIN *ENDE*-RECORD HINZUGEFUEGT WORDEN') 230
      RETURN                                                           240
      END                                                              250
```

3. PROGRAMS PERFORMING ARITHMETIC OPERATIONS ON NUCLEAR DATA

In the course of nuclear data evaluation work it is quite frequently necessary to perform simple arithmetic operations on data sets such as

- multiplication with a constant (e.g. renormalisation) or
- summing with or without weighting (e.g. calculation of averages, of best estimates or of cross sections for isotopic mixtures).

Two programs for calculations of this type, CALCUL and the Karlsruhe version of SCØRE, are described in the sequel. Other programs like PRTSTKED, SELPLO, SIGPLO perform also a certain amount of arithmetic calculations, but since this is not their main purpose, they are described under different headings in the present compendium (vol. V 2.1, 2.2, 4.1 respectively)

3.1 Simple Averages (AVERAGE)

Contents:

3.1.1 Purpose of AVERAGE

Calculation of simple averages provides a quick and convenient way to compare data sets (e.g. data from different measurers) or to check modified (e.g. reformatted) data against the original data. The FORTRAN IV program AVERAGE was written to calculate unweighted averages of point data from three types of input sources:

- KEDAK,
- files in NEUDADA "transmission" format,
- card images.

The point data are assumed to be given as abscissae x_i (put by the program in ascending order, if necessary) and ordinates y_i, $i = 1,2,\ldots,n$.

The average $\bar{y}$ in a specified interval $(x_A \ldots x_B)$ is defined as

$$\bar{y} = \frac{1}{x_B - x_A} \int_{x_A}^{x_B} y(x)dx,$$

the function $y(x)$ being obtained from the point data by

$$y(x) = y_i + (y_{i+1} - y_i) \frac{x - x_i}{x_{i+1} - x_i} \qquad \text{if } x_i \leq x \leq x_{i+1};$$

$$y(x) = y_1 \qquad \text{if } x < x_1,$$

$$y(x) = y_n \qquad \text{if } x > x_n,$$

i.e. by linear interpolation within the interval of point data, and by utilisation of the nearest ordinate outside this interval. The average is set equal to zero if the range of point data $(x_1 \ldots x_n)$ and the averaging interval $(x_A \ldots x_B)$ do not overlap.

The full capability of the program can be seen from the following operating description written by the author of AVERAGE in 1973. A program list is appended.

3.1.2 Operating description for AVERAGE

by R. Meyer

Contents

1) Introduction

AVERAGE is a program designed as a convenient means to calculate unweighted averages for neutron cross section data from three types of input sources: KEDAK, NEUDADA-format files, card image.
Its main purpose is to retrieve averages quickly and conveniently for comparison of different data sets and/or checking of derived - e.g. reformatted - data.
Since averaging is done straight forward, AVERAGE is not intended to replace any of the current group constants programs. Also, resultant averages should be used only in the manner described above. The user should not hope to get accurate results for further application purposes by this program and is referred to the respective averaging capabilities of MIGROS or other similar programs to evaluate more accurate results.

2) How to get the program 1),2)

The program is part of the program services for the nuclear data library KEDAK.
At present it is member of the dataset INR.STEIN.LOAD on the set-up dis GFK029 the mounting of which may be requested by job control cards. The name of the member is AVERAGE.
You may use the following job control cards to obtain the program:

```
//   JØB card
/*SETUP_DEVICE=2314,ID=GFK029
//_EXEC_FHLG,PARM.L=MAP
//L.LIB_DD_UNIT=2314,VØL=SER=GFK029,DISP=SHR,
  DSN= INR.STEIN.LOAD
//L.SYSIN_DD_*
_INCLUDE_LIB(AVERAGE,LDFPAC,DEFI,PRIEIN)
_ENTRY_MAIN
/*
 .
 .  JOB control cards for running
 .  the program - see following paragraphs.
 .
//  Job end card
```

Chapter 9) contains a summary of input instructions to operate AVERAGE. It may be used for quick reference.
Experienced programmers may read that section only and use the other paragraphs for reference in cases where chapter 9) is not sufficient.

1) In the following text an underscore (_) will be used to denote blanks.
2) This paragraph is intended for Karlsruhe users only.

3) How the program is controlled by the user

The user controls the program by supplying

program control input.

Program control input may be supplied by punched cards. These punched cards must immediately follow the job control card

```
//G.SYSIN_DD_ *
```

This job control card together with the succeeding program control input may be placed anywhere among the job control cards for running the program (see preceding paragraph).

Program control input is supplied by n a m e l i s t i n p u t and must follow its rules /1/. These rules will not be repeated here. However the user is invited to follow the examples given and look into reference /1/ only in crucial cases.
Only one namelist name is used: IN . Its parameters are explained in detail in the following sections.

Examples:

```
//G.SYSIN_DD_*
_&IN_INPUT='KEDAK',NAMES='U 235','SGG',EMIN=1.24E+3,
_7.86E+3,EMAX37.86E+3,19.07E+3,&END
_&IN INPUT='CARD',UNIT=9,FMT='(2E 14.6)',NUM=1240,
_&END
/*
```

The above example shows the program control input for averaging two data sets, one from KEDAK and one from card image input.

Note the general format of namelist input data
_&IN_, parameterlist, &END
where parameterlist is a list of names and values separated by commas:
Parameterlist $\hat{=}$ PARM1 = $value_1$, PARM2 = $value_{21}$, $value_{22}$,...
Note, that one or several values may be assigned to a named variable. This is explained in more detail later.
You will also notice, that each input line starts in column two. It may be extended up to column 80 and continuation lines are allowed. An input list is terminated by &END.
You also will see, that some data are enclosed within apostrophes. These data will be interpreted as text, whereas all other data will be taken to be numeric.
The line containing /* is optional. If used it terminates the program control input. If missing, program control input will be terminated by the next job control card encountered after // G.SYSIN...
Although not recommended, program control input may be mixed with data input. This will be shown later.

4) Input of energy limits.

Upper and lower energy limits may be defined using the parameters EMIN and EMAX.
Several values may be assigned to each of them.
Averaging will be done within the energy intervals
EMIN (1) - EMAX (1)
EMIN (2) - EMAX (2)
⋮
A maximum of 100 averaging intervals my be used.
Energy limits are assumed to be given in eV.
Example:

```
//G.SYSIN DD *
 &IN INPUT = 'KEDAK', NAMEN= 'FE', 'SGT'.
 EMIN = 1.E6, 2.E6, 3. E6, 10.E6 ,
 EMAX = 2.E6, 3.E6, 4. E6, 12.E6,
 &END
```

```
_ &IN_NAMEN (2) = 'SGG', &END
_ &IN_NAMEN(2)  = 'SGN', &END
/*
```

In this example the first input list requests averaging of KEDAK data for the total cross section of iron in the intervals <1MeV, 2MeV> ,<2MeV, 3MeV>, <3MeV,4MeV>, <10MeV,12MeV>. The second input list requests averaging for the capture cross section of iron. Since no new energy limits are defined, the previous ones will be used.
The same holds for the third line, where averaging for the elastic cross section of the same material will be done.

Note that the intervals need not be continuous. Therefore minimum and maximum energy both are required. A limiting energy need not be contained in the tabulation energies but interpolation will be performed.
The requested data however must cover the specified energy intervals.

Attention:

- linear interpolation will be used always. If linear interpolation is not sufficient, a suitable data set may be prepared using the interpolation capabilities of SIGPLO /2/ or any other interpolating program.
  Note however, that AVERAGE is only intended to give good estimates of averages for quick comparison or check.

- It is not possible to redefine energy limits by the following method:

```
//G.SYSIN_DD_x
_&IN_INPUT='KEDAK',NAMEN='FE','SGT',
_EMIN=1.E6,2.E6,EMAX=2.E6,3.E6,&END
_&IN_TYP='SGG',EMIN(2)=3.E6,EMAX(2)=4.E6,&END
/*
```

In fact, the second input list would not redefine any energy limits. If energy limits are given, all of them must be

specified from the very beginnig.
They will then make up the new set of averaging intervals.
That is a complete set of energy intervals must be input.
It is not possible to just modify the current one.
The correct procedure for the above example would be:

```
//G.SYSIN DD *
_&IN_INPUT='KEDAK',NAMEN='FE','SGT',
_EMIN=1.E6,2.E6,EMAX=2.E6,3.E6,&END
_&IN_TYP='SGG',EMIN=1.E6,3.E6,EMAX=2.E6,4.E6,&END
/*
```

5) Requesting KEDAK data

The preceding paragraphs already gave a few examples of retrieving data from the KEDAK library.
Use the following parameters for this purpose:

```
INPUT, NAMES, NAMZ, PLEH
```

a) With INPUT you specify the kind of input source. In the present case you code INPUT = 'KEDAK'.
If the current value of INPUT is already 'KEDAK' you need not respecify it.

b) PLEH allows you to specify the FORTRAN data set reference number for the KEDAK library to address. Thus it is possible to address two or more KEDAK libraries in one job.
For each KEDAK library addressed you must supply the corresponding DD-card. Do not forget that you will need 8K buffer for each KEDAK library usually.

Example:

If you code PLEH = 1 the KEDAK library is expected on logical unit 1. A DD-card describing the KEDAK library must be present:

```
//G.FT01F001 DD UNIT=2314,VOL=SER=GFK050,DISP=SHR,DSN=KEDAK3
```

This DD-card would address the present released version of KEDAK. You insert this DD-card anywhere among the job control cards for running the program (see paragraph 1). But you must not mix it with your program control data.
PLEH = 1 is the default value: you need not specify PLEH if it is to be 1.

c) You use NAMES to address a specific cross section set on KEDAK, in the order: material name, data type name,

and if applicable also an excitation energy, e.g.:

```
NAMES='FE','SGT'
NAMES='U_238','SGF'
NAMES='PU239','SGIZ',0.
NAMES='FE___EN3','SGG'
```

You will notice, that each non-numeric name must be enclosed in apostrophes. For definition and explanation of KEDAK names you should refer to /3/.

d) NAMZ is a parameter to specify the number of names necessary to address a KEDAK data set.
Its default value is 2.

Example:

```
//G.SYSIN_DD_*
_&IN_NAMES='FE','SGT',EMIN=1.E+6,EMAX=1.E+7,&END
_&IN_NAMES(2)='SGN',&END
_&IN_TYP='SGP',&END
_&IN_MAT='FE___EN3',TYP='SGT',NAMZ=2,PLEH=2,&END
_&IN_TYP='SGN',&END
_&IN_TYP='SGP',&END
_&IN_TYP='SGIZ',EXC=0.,NAMZ=3,&END
_&IN_MAT='FE,TYP='SGX',PLEH=1,NAMZ=2,&END
/*
//G.FT01F001_DD_UNIT=2314,VØL=SER=NUSYSO,DISP=SHR,DSN=KNDF
//G.FT02F001_DD_UNIT=2314,VØL=SER=GFKO29,DISP=SHR,DSN=ENDFB3.CONV
/*SETUP_DEVICE=2314,ID=NUSYSO
```

In this example averages will be calculated for the interval <1MeV, 10Mev> for σ_T, σ_n , σ_p and the excitation cross section of the first inelastic level for two materials: 'FE' and 'FE___EN3', where the former material is obviously to be retrieved from the released version of the KEDAK library

via // G.FT01F001- since no PLEH is given and therefore is one by default - and the second material is some conversion from ENDF/B3 and is taken from a data library that is specified by // G.FT02F001...

. Note, that in the last input line it is necessary to specify PLEH = 1 although this was the default value, since its value had been changed from default one to two by input line 5.
. Also note, that it is necessary to redefine NAMZ = 2 in input lines 5 and 9.
. Note the alternative use of MAT, TYP and EXC for NAMES (1) , NAMES (2), NAMES (3) respectively. Coding NAMES (1) or MAT is completely identical in effect. The same holds for the other alternatives.

Attention:

. A maximum of five KEDAK libraries may be used in one run. The addressed materials must be uniquely defined, i.e. no equally named materials must be contained in two different libraries, or confusion might occur. 8K buffer area will be required by each addressed library normally.
. If a different package of retrieval routines is used for the KEDAK library than given in paragraph 2) PLEH = 1 may be used only.
. Energies are expected to be in eV.

6) How to get NEUDADA-data

The following parameters control retrieval of data from NEUDADA-tapes[x]:

||INPUT, UNIT, NEUFOR, NAMES, SCALE||

a) INPUT

You request data to be retrieved from a data set in NEUDADA-format, if you code INPUT = 'NEUDADA'. If the current value of INPUT already is 'NEUDADA' you need not respecify it. Simply omit parameter INPUT in this case.

b) UNIT

You must specify the Fortran data set reference number which applies to the data to be retrieved, e.g. UNIT = 3 would imply that the data shall be fetched from a data set described by // G.FT03F0001_ DD _ .....
You will observe that each unit number is linked with a DD-card. This DD-card must describe the features of the NEUDADA-format data set. See below for examples.
It is absolutely necessary to specify the unit number, since its default value (UNIT = 5) must not be applied to NEUDADA data. 2K buffer area will be needed for each NEUDADA file normaly.

c) NEUFOR

NEUDADA tapes are distributed by CCDN/Saclay usually in the NEUDADA Transmission Format. On request it is also possible to obtain data in Internal Expanded Format. The latter format reduces the amount of storage and retrieval time needed to keep and access the data.

Code NEUFOR = 1 for data in Internal Expanded Format,
NEUFOR = 2 for data in Transmission Format.
NEUFOR = 2 is the default value. If not previously respecified by the user, the parameter may be omitted if the default value should be applied.

d) NAMES

Data sets in both NEUDADA formats get an identifier of six characters in length. With each NEUDADA tape you will receive a list displaying the contents of the tape and the identifiers given to the various data sets on the tape. If you wish to retrieve a given data set, you should look up its corresponding identifier on that list. Specify this identifier in your program control input, e.g.

NAMES = 'E00001' or
NAMES = 'U00049'

Make sure, that the desired data have received a unique identifier. If not, use the capability of AVERAGE to concatenate these two or more sets. Refer to paragraph 8) for this purpose.

Attention:

The current contents of NAMES are destroyed when requesting NEUDADA input, i.e. although the user actually only fills NAMES (1), the current values of NAMES (2) , ... are destroyed since they will be used by the program when it retrieves the data.
Therefore do not expect in the following example:

```
_&IN_INPUT='KEDAK',NAMES='FE','SGT',&END
_&IN_INPUT='NEUDADA',NEUFOR=2,NAMES='E00022',&END
_&IN_INPUT='KEDAK',NAMES='CR',&END
```

to obtain the total cross section of chromium in the third input line.

e) SCALE

Energies on NEUDADA format data sets are expected to be given in MeV, cross section values in barn. In many time-of-flight measurements however cross sections are given in barns * $\sqrt{E}$. These cases will be indicated on the list you receive from CCDN/Saclay together with the tape. The program AVERAGE optionally permits you to have cross section data converted from this scale to the normal scale (barns) by specifying SCALE = 'SQE'. You may nullify this specification by coding SCALE = '-', which is the default as long as no other value has been specified.

Example:

```
//G.SYSIN DD *
_&IN_INPUT='NEUDADA',UNIT=2,NAMES='E00013',NEUFOR=2,
_EMIN=0.25,1.02,2.66,4.81,
_EMAX=1.02,2.66,4.81,9.67,
_&END
_&IN_NAMES='E00016',&END
_&IN_NAMES='U00027',&END
_&IN_NAMES='E00001',UNIT=3,NEUFOR=1,&END
/*
//G.FT02F001 DD UNIT=TAPE7,VOL=SER=ND0607,LABEL=(2,NL,,IN),
//_DCB=(RECFM=FB,LRECL=132,BLKSIZE=1320,TRTCH=ET,DEN=1)
//_DISP=(OLD,PASS)
/*SETUP DEVICE=TAPE7,ID=(ND0607,NORING,,NL)
//G.FT03F001 DD UNIT=TAPE9,VOL=SER=ND0657,LABEL=(2,NL,,IN),
//_DCB=(RECFM=FB,LRECL=72,BLKSIZE=720),DISP=(OLD,PASS)
/*SETUP DEVICE=TAPE9,ID=(ND0657,NORING,NL)
```

In this example averaging is requested in the four intervals <0.25eV,1.02 eV> , <1.02eV, 2.66eV> , <2.66eV,4.81eV> , <4.81eV,9.67eV> for four data sets, all residing on NEUDADA-tapes.
The first three data sets 'E00013', E00016', '000027' are on the 7-track tape ND0607 in transmission format.
This tape is accessed by // G.FT02F001_ DD _ .. and the respective Setup card.
The fourth data set 'E00001' is on 9-track tape ND0657 in internal expanded format. This tape is accessed by // G.FT03F001 _ DD _ ... and the respective Setup card.

Note the following:

- The above two DD-cards are two typical examples of how the DD-cards for NEUDADA-tapes must look like. If you do not have any experience in coding DD-cards you should take the paper label which is distributed with each tape and contact some more experienced person for coding the DD-statements. Bring the above lines as example with you. The paper label will contain all information to build up the necessary DD-card.
  Since CCDN/ Saclay presently is operating its IBM 360/30 in DOS the difference in tape marks between DOS and OS must be considered usually leading to the result that DOS-files 1,2,3, ... will be found in OS as files 2,4,6, ...

- Be aware of the fact, that energy limits are always specified in eV, no matter what units the energies of the requested data points are. The program will take care of the necessary conversion.
  Conversion always will be to eV.

Program execution may be slowed down appreciably, if data sets are not specified in the order they reside on tape. Backspacing and rewinding however will be provided by the program automatically.
Note that the program expects the tape to be numerically sorted in ascending order, i.e. it expects E00013 to be physically behind E00012 and U00002 behind U00001.
It does not have any expectation concerning the sorting of E and U.

7) Input of card image data

A third input possibility is offered for all those data sets to which neither of the preceding formats apply. Data may also be input from card image data sets with the applicable format being defined by the user.
The following parameters are available for card image input:

||INPUT, UNIT, FMT, NUM,C||

a) INPUT

To request card image input you must code INPUT = 'CARD'. It is not necessary to repeat this specification if the current value of INPUT already is 'CARD'.

b) UNIT

By UNIT = nn you specify the data set reference number to be used for reading the data input, e.g. coding UNIT = 9 will cause the requested data to be read in from // G.FT09F001 _ DD _ ... Do not forget to supply a DD-card for each data set you reference. You may insert this DD-card anywhere among the job control cards for running the job (see paragraph 2). You must not mix this DD-card however with any of your data input or program control input cards.
If you omit the UNIT-specification, data will be read from logical unit five (unless no different specification preceded). Logical unit five is the program control data set. Although it is not recommended it is thus possible to mix data- and program control input.

c) FMT

With FMT you specify the input format to be used for reading the data. Since no default value is available you must specify a format.

Note that input energies are expected to be in eV.
However by using a suitable scaling factor in your format (/1/) you may select different energy units for your input data.

d) NUM

Gives the number of data points to be read. Be sure to use the correct number since an error in NUM may cause the job to fail. For the maximum number of data points permitted see the respective paragraph.

e) C

C may be set .True. or .False. respectively if a comment card precedes the data set to be read or not.
Code C = .TRUE. or C = T if so ,
or C = .FALSE.or C = F else.
It is highly recommended that such a card precedes the data set. It will be read and printed by the program and offers a convenient means to identify the data set.
Output from SIGPLO /2/ will contain such heading identifying cards.
If it was not respecified by the user, C will be .TRUE. by default.

Example:

```
//G.SYSIN DD *
_&IN_INPUT='CARD',UNIT=9,FMT='(2E10.5)',NUM=5,
_EMIN=1.,EMAX=5.,&END
_&IN_UNIT=5,NUM=4,&END
NOTE_THAT_INPUT_READ_FROM_UNIT_5_MUST_IMMIDIATELY_FOLLOW.
     1.0      1.0
     3.0      1.8
     4.0      2.5
     6.2      1.9
              ^
              |_____________________ Column 10
```

```
_&IN_UNIT=9,NUM=2,&END
_&IN_UNIT=10,NUM=4,FMT='(10F8.5)',&END
_&IN_NUM=3,&END
/*
//G.FT10F001_DD_*
COMMENT:THIS_IS_THE_FIRST_INPUT_DATA_SET_FROM_UNIT_10.
   0.5      1.9      2.8      0.97      2.4      1.01      5.08      0.42
COMMENT:THIS_IS_THE_SECOND_INPUT_FROM_UNIT_10.
   6.0      1.87     2.0      1.97      0.99     1.88
/*
//G.FT09F001_DD_*
HEADING_CARD_1
    0.97        1.87
    1.62        0.98
    3.05        1.17
    4.99        1.86
    5.17        2.88
HEADING_CARD_2
    1.0         1.0
    5.0         1.0
/*
```

From this example you may learn:

- data are read in this order E(1), σ(1), E (2), σ(2), ... using the specified format.
- Each card may contain one or more data points. This is exclusively a matter of format.
- If data are to be read from the control unit 5, they must immediately follow the requesting control card, as is the case for input list number two.
- Data need not be ordered. The program will reorder them if necessary.
- It is essential that you observe your format specifications and punch the data in the appropriate columns.

Notes

Although not used in the above example, the data set may of course reside also on tape or disc and its line length may exceed 80 columns. Comment will always however be read in 80 columns length. Therefore no shorter line length should be used or the job will fail.

8) Concatenating data sets.

In some cases it might be necessary to combine two or more sets of data into one before averaging.
This can be done by using the parameter ADD.
If you specify ADD = .TRUE. or ADD = T the next set of data to be read will be added to the one currently read. Averaging will be postponed.

Example:

```
//G.SYSIN_DD_*
_&IN_INPUT='NEUDADA',NAMES='E00014',UNIT=1,ADD=T,
_EMIN=1.67E+4,EMAX=2.36E+4,&END
_&IN_NAMES='E00015',&END
_&IN_NAMES='E00016',&END
_&IN_INPUT='CARD',NUM=1,FMT='(2F10.5)',C=F,ADD=F,&END
    1.67E+4    0.0
_&IN_INPUT='KEDAK',NAMES='FE','SGT',UNIT=2,&END
/*
//G.FT01F001_DD_UNIT=TAPE9,VOL=SER=ND2108,DISP=(OLD,PASS),
//_LABEL=(2,NL,,IN),
//_DCB=(RECFM=FB,LRECL=132,BLUSIZE=1320)
//G.FT02F001_DD_UNIT=2314,VOL=SER=GFK050,DISP=SHR,
//_DSN=KEDAK3
/*SETUP_DEVICE=2314,ID=GFK050
/*SETUP_DEVICE=TAPE9,ID=(ND2108,NORING,,NL)
```

In this example the three data sets 'E0014' , 'E00015', 'E00016' on NEUDADA-tape and the single data point from card input are combined into one set of data. Note that this is possible since the data are being reordered before they are used. Averaging is performed after the last data set to be added has been read in, which is indicated by ADD = F.

Attention:

Do not forget to turn off the ADD-function when requesting input of the last set of data to be added, or unwanted results will be obtained.

9) Quick reference for the experienced user.

The program AVERAGE is controlled by program control input. This input is supplied via // G.SYSIN _ DD _ .... and is assigned logical unit 5. It is in the form of namelist data. The only namelist name being used is IN. The following input parameters may be used:

EMIN, EMAX } to specify averaging intervals

INPUT = 'CARD' / 'KEDAK' / 'NEUDADA' — for requesting card image - , KEDAK - , or NEUDADA input

NAMES, NAMZ, UNIT } to control retrieval from KEDAK

NAMES, UNIT, NEUFOR, SCALE } to control retrieval from NEUDADA files.

UNIT, FMT, NUM, C } to control card image input.

ADD - to combine two or more data sets before averaging.

Averaging intervals are given by entering values for EMIN, EMAX. Up to 100 values may be entered for each. Each pair {EMIN (1), EMAX (1)} sets up one averaging interval . The values must be given in eV. The set of intervals remains in effect until a new set is entered. Only complete sets can be entered at a time. If an EMAX (I) < EMIN (I), the corresponding average will be set 0.

INPUT source is requested by parameter INPUT. Its value remains in effect until a new one is specified.

KEDAK data may be retrieved by specifying the appropriate names of the requested data set by NAMES =, giving the number of names by NAMZ (if different from two) and specifying the logical unit to access the Kedak library by UNIT = nn. For each accessed KEDAK library a DD-card must be available. Up to five KEDAK libraries may be accessed in one job step. Each KEDAK library accessed will eat up 8K of buffer area.
If a retrieval routine package other than LDFPAC is used, the UNIT-number must be 1 and only one library can be accessed within one job step.
Energies are expected to be in eV.

NEUDADA data may be retrieved from both transmission format or internal expanded format files. This is controlled by specifying NEUFOR = 2 or NEUFOR = 1 respectively, where the former value is default. The 6-character data set name is specified by NAMES = '....' . Note that all locations of NAMES are used and therefore previous contents of NAMES (2), NAMES (3), ... are not kept. The program expects the identifier to consist of a leading character followed by five decimal digits, and that within each character group data sets are ordered according to increasing numeric part. Backspacing and Rewind is controlled by the retrieval program on the basis of this assumption.
The logical unit assigned to the NEUDADA file is given by UNIT = nn.
Optionally, the user may switch from cross section values in barn to barn $*\sqrt{E}$ by specifying SCALE = 'SQE'.
The program then will calculate the correct cross section values.
This specification is nullified by SCALE = '_' ,
Energies are expected to be in MeV and are converted to eV accordingly.
For data set attributes and data set identifiers consider the labels and listings accompanying each tape.

Card image input for all other kinds of data requires specification of the input format via FMT. The format specification may not exceed 40 characters. The minimum line length is eighty. An optional heading card for reading and printing only is read to 80 characters length only. If this length is exceeded, the text will be truncated. The heading card will be read if logical variable C is true (default), else it will be assumed that the starting record of the data set already contains numerical data. Scaling of data to eV/barns must be done by format scaling factors, if necessary.
UNIT specifies the logical unit to be used for reading.
If 5 , the data records must immediately follow the namelist requesting them. If not 5, an appropriate DD-card must be available. NUM specifies the number of data points to be read.
Data points need not be sorted.

Combination of data sets is requested by setting logical variable ADD to .True. . If so, the data requested by the current namelist input are not averaged but kept and combined with the data requested by the succeeding namelist input. This is repeated until ADD = F terminates this process. ADD = F must be coded on the last data set to be concatenated and will cause averaging of the combined data. Data sets need not be concatenated in their numerical order.

Defaults if mentioned apply to program start.
Values assigned to any parameter (except NUM) remain in effect until respecified by the user und thus form a set of "current defaults".

10) Notes, Region

- Data will be retrieved from KEDAK/NEUDADA within the energy range defined by the lowest valued EMIN and the maximum value of EMAX. Card image data will be read as specified by NUM.

- Interpolation to limiting energies is done linear in x/linear in y. The same interpolation rule is used for integration.

- Data need not be sorted, but will be sorted by the program if necessary.

- Running times are dificult to estimate since very different storage media may be used. It should be emphasized however, that the averaging procedure usually will require least of the running time.

- Allocation of working area for the cross section data is dynamical. 30 % of the available free region will be reserved for buffers, up to a maximum of 40K buffer area. Jobs requiring more buffer space cannot be run but must be separated to reduce buffer request. Allocation of buffer and working area is displayed to the user, so he may after an initial run easily estimate the actual REGION parameter he must apply provided he knows the maximum number of data points to be averaged. Note that you always may reduce the required amount of core by dividing one set of averaging intervals into several.

- Output is hoped to be self- explanative and is not discussed here.

11) Example:

This last example is designed to give a quick overview of the job control cards and program control input necessary to run the program.

```
//INR613AV_JOB_(0613,101,P0000),MEYER,REGION=240K,TIME=3
//_EXEC_FHLG,PARM.L=MAP
/XSETUP_DEVICE=2314,ID=GFK029
/*SETUP_DEVICE=2314,ID=GFK050
/*SETUP_DEVICE=TAPE7,ID=(ND5321,NORING,,NL)
//L.LIB_DO_UNIT=2314,VOL=SER=GFK029,DISP=SHR,DSN=INR.STEIN.LOAD
//L.SYSIN_DD_*
_INCLUDE_LIB(AVERAGE,LDFPAC,DEFI,PRIEIN)
_ENTRY_Main
/*
//G.FT01F001_DD_UNIT=2314,VOL=SER=GFK050,DISP=SHR,DSN=KEDAK3
//G.FT02F001_DD_UNIT=2314,VOL=SER=GFK029,DISP=SHR,DSN=MEYER.ENDFB3
//G.FT03F001_DD_UNIT=TAPE7,VOL=SER=ND5321,DISP=(OLD,PASS),
//_LABEL=(2,NL,,IN),
//_DCB=(RECFM=FB,LREC=132,BLKSIZE=1320,TRTCH=ET)
//G.FT09F001_DD_UNIT=2314,VOL=SER=GFK029,DISP=SHR,
//_DSN=MEYER.AVERG.CARDIM
//G.SYSIN_DD_*
_&IN_UNIT=1,INPUT='KEDAK',NAMES='NA-23','SGG',
_&EMIN=0.5E+3,1.E+3,2.E+3,2.5E+3,3.E+3,4.E+3,6.E+3,8.E+3,10.E+3
_EMAX=1.E+3,2.E+3,2.5E+3,3.E+3,4.E+3,6.E+3,8.E+3,10.E+3,15.E+3,
_&END
_&IN_UNIT=2,MAT='NA_23EN3',&END
_&IN_UNIT=3,NAMES='E00001',INPUT='NEUDADA',ADD=T,
 &END
_&IN_NAMES='E00014',ADD=F,&END
_&IN_UNIT=9,INPUT='CARD',FMT='(2E14.6)',NUM=1487,&END
/*
//job end card
```

The reader is invited to interpret the above example himself.

References

/1/ IBM 05/360 FORTRAN IV Language manual

/2/ R. Meyer, SIGPLO program description to be published

/3/ B. Krieg, Program services of the evaluated nuclear data library KEDAK, Part I, KFK 1725/I

```
      DIMENSION X(1),EO(100),EU(100),S(100)                                 10
      DIMENSION EMIN(100),EMAX(100),NARG(5)                                 20
      REAL*8 NAMES(3),NAMOLD(3)/3*'        '/,INPUT,CARD/'CARD'/,KEDAK/'    30
     1KEDAK'/,FMT(5)/5*'        '/,TXT(10),NEUDAD/'NEUDADA'/,SQ/'SQE'/,      40
     2SCALE/'  '/,MAT,TYP,EXC                                               50
      COMMON/INOUT/KOUT                                                     60
      LOGICAL C/T/,ADD/F/,ADAT/F/,STANZ/F/                                  70
      INTEGER UNIT,PLEH                                                     80
      LOGICAL*1 KEKONT(99)/99*F/                                            90
      EXTERNAL LDFLOC,LDFNXT,NEULOC,NEUNXT,PRIEIN                          100
      EQUIVALENCE (KE,UNIT),(NAMES(1),MAT),(NAMES(2),TYP),(NAMES(3),EXC)   110
      NAMELIST/IN/NAMES,NAMZ,EMIN,EMAX,INPUT,FMT,KE,NLM,C,NEUFOR,UNIT,     120
     1 SCALE,ADD,MAT,TYP,EXC,STANZ,KSTANZ,DIFV,MAXINT,PLEH                 130
      DATA MAXE/100/,KIN/5/,NNI/0/,NAMZ/2/                                 140
      KOUT=6                                                               150
      KSTANZ=7                                                             160
      MAXINT=25                                                            170
      DIFV=0.0                                                             180
      PLEH=1                                                               190
      CALL PRIEIN(KIN,KOUT,60)                                             200
      NEUFOR=2                                                             210
      KE=KIN                                                               220
      CALL SPACE(X(1),LX,NMAX)                                             230
    5 DO 10 I=1,MAXE                                                       240
      EMIN(I)=0                                                            250
      EMAX(I)=0                                                            260
   10 CONTINUE                                                             270
      READ(KIN,IN,END=1010)                                                280
      MAXINT=MAXO(1,MINO(MAXINT,100))                                      290
      IF(DIFV.LE.0) GOTO 2                                                 300
      EMAX(1)=EMIN(1)+DIFV                                                 310
      IF(MAXINT.LE.1) GOTO 6                                               320
      DO 3 I=2,MAXINT                                                      330
      EMIN(I)=EMAX(I-1)                                                    340
    3 EMAX(I)=EMIN(I)+DIFV                                                 350
    6 II=MAXINT+1                                                          360
      IF(II.GT.100) GOTO 2                                                 370
      DO 7 I=II,100                                                        380
      EMIN(I)=0.0                                                          390
    7 EMAX(I)=0.0                                                          400
    2 IF(.NOT.ADAT) NTOT=0                                                 410
      WRITE(KOUT,605) INPUT                                                420
  605 FORMAT(1H1,'       TYPE OF INPUT REQUESTED=',A8/)                    430
      IF(INPUT.NE.KEDAK) GOTO 12                                           440
      WRITE(KOUT,600) (NAMES(I),I=1,NAMZ)                                  450
  600 FORMAT('      AVERAGING FOR  ',2(A8,2X),1P4E13.5/)                   460
      IF(.NOT.KEKONT(PLEH)) CALL LDFOPN(PLEH,IDAT,A1C1C)                   470
      KEKONT(PLEH)=.TRUE.                                                  480
      GOTO 19                                                              490
   12 IF(INPUT.EQ.NEUDAD) GOTO 14                                          500
      WRITE(KOUT,606) NUM,KE,FMT                                           510
  606 FORMAT('       AVERAGING FOR CARD INPUT REQUESTED:'/                 520
     1'        NUM=',I5,'       KE=',I2,'    FMT=',5A8/)                  530
      GOTO 19                                                              540
   14 WRITE(KOUT,600) NAMES(1)                                             550
      WRITE(KOUT,611) UNIT,NEUFOR                                          560
  611 FORMAT(T10,'UNIT=',I2,'   NEUFOR=',I2/)                              570
   19 IF(ADAT) GOTO 35                                                     580
      DO 20 I=1,MAXE                                                       590
      IF(EMIN(I).NE.0.OR.EMAX(I).NE.0) GOTO 20                             600
      NI=I-1                                                               610
      GOTO 30                                                              620
   20 CONTINUE                                                             630
      NI=MAXE                                                              640
   30 IF(NI.NE.0.OR.NNI.NE.0) GOTO 32                                      650
      WRITE(KOUT,601)                                                      660
  601 FORMAT('       NO ENERGY LIMITS GIVEN')                             670
      GOTO 1000                                                            680
   32 IF(NI.EQ.0) GOTO 35                                                  690
      NNI=NI                                                               700
      DO 34 I=1,NI                                                         710
      EO(I)=EMAX(I)                                                        720
      EU(I)=EMIN(I)                                                        730
   34 CONTINUE                                                             740
      EA=XMIN(NNI,EU)                                                      750
      EB=XMAX(NNI,EO)                                                      760
   35 IF(INPUT.EQ.KEDAK) GOTO 36                                           770
      IF(INPUT.EQ.NEUDAD) GOTO 50                                          780
      IF(.NOT.C) GOTO 31                                                   790
      READ(KE,500) TXT                                                     800
  500 FORMAT(10A8)                                                         810
      WRITE(KOUT,607) TXT                                                  820
  607 FORMAT(1X,'COMMENTCARD=***',10A8,'***'/)                             830
   31 IF(NUM+NTOT.GT.NMAX) GOTO 990                                        840
      NTOX=NTOT+LX-1                                                       850
      NTOY=NTOT+LX+NMAX-1                                                  860
      READ(KE,FMT) (X(NTOX+I),X(NTOY+I),I=1,NUM)                           870
      NUMX=NUM                                                             880
      NAMOLD(1)=CARD                                                       890
      GOTO 39                                                              900
   36 NARG(1)=NAMZ                                                         910
      DO 37 I=1,NAMZ                                                       920
      IF(NAMOLD(I).NE.NAMES(I)) GOTO 38                                    930
   37 CONTINUE                                                             940
      GOTO 39                                                              950
   38 CALL RETXS(NARG,NAMES,EA,EB,X(NTOT+LX),X(NTOT+LX+NMAX),NUMX,         960
     1NMAX-NTOT,NR,LDFLOC,LDFNXT)                                          970
      DO 33 I=1,NAMZ                                                       980
   33 NAMOLD(I)=NAMES(I)                                                   990
      IF(NR.GE.2.AND.NR.NE.10) GOTO 990                                   1000
      GOTO 39                                                             1010
   50 NARG(1)=3                                                           1020
      NARG(2)=1                                                           1030
      NARG(3)=1                                                           1040
      NARG(4)=UNIT                                                        1050
      NARG(5)=NEUFOR                                                      1060
      NAMES(2)=UNIT                                                       1070
      NAMES(3)=EA                                                         1080
      DO 51 I=1,3                                                         1090
      IF(NAMES(I).NE.NAMOLD(I)) GOTO 52                                   1100
```

```
 51 CONTINUE                                                            1110
    GOTO 39                                                             1120
 52 EX=EA*1.E-6                                                         1130
    EY=EB*1.E-6                                                         1140
    CALL RETXS(NARG,NAMES,EX,EY,X(NTOT+LX),X(NTOT+LX+NMAX),NUMX,        1150
   1NMAX-NTOT,NR,NEULOC,NEUNXT)                                         1160
    DO 53 I=1,3                                                         1170
 53 NAMOLD(I)=NAMES(I)                                                  1180
    IF(NR.GE.2.AND.NR.NE.10) GOTO 990                                   1190
    J=NTOT+LX                                                           1200
    DO 54 I=1,NUMX                                                      1210
    X(J)=X(J)*1.E+6                                                     1220
 54 J=J+1                                                               1230
    IF(SCALE.NE.SC) GOTO 56                                             1240
    J=NTOT+LX+NMAX                                                      1250
    DO 55 I=1,NUMX                                                      1260
    X(J)=X(J)/SQRT(X(J))                                                1270
 55 J=J+1                                                               1280
 56 CONTINUE                                                            1290
 39 NTOT=NTOT+NUMX                                                      1300
    IF(.NOT.ADD) GOTO 57                                                1310
    ADAT=.TRUE.                                                         1320
    GOTO 5                                                              1330
 57 ADAT=.FALSE.                                                        1340
    CALL ORDNEN(NTOT,X(LX),X(LX+NMAX))                                  1350
    CALL EQUEN(NTOT,X(LX),X(LX+NMAX))                                   1360
    WRITE(KOUT,602)                                                     1370
602 FORMAT(//6X,'FROM',T24,'TO',T36,'AVERAGE'/)                         1380
    DO 40 I=1,NNI                                                       1390
    CALL AVERG(X(LX),X(LX+NMAX),NTOT,EU(I),EO(I),S(I))                  1400
    WRITE(KOUT,610) I,EU(I),EO(I),S(I)                                  1410
610 FORMAT('0',I5,2X,1P3E15.5)                                          1420
    IF(.NOT.STANZ) GOTO 40                                              1430
    EUO=0.5*(EU(I)+EO(I))                                               1440
    WRITE(KSTANZ,612) EUO,S(I)                                          1450
612 FORMAT(2E14.6)                                                      1460
 40 CONTINUE                                                            1470
    GOTO 1000                                                           1480
990 IF(NR.NE.2) GOTO 992                                                1490
    WRITE(KOUT,603)                                                     1500
603 FORMAT('     NOT ENOUGH INCORE STORAGE FOR THIS DATA-TYPE.'/)       1510
    GOTO 1000                                                           1520
992 WRITE(KOUT,604)                                                     1530
604 FORMAT('     NO DATA IN REQUESTED ENERGY INTERVAL(S).'/)            1540
    GOTO 1000                                                           1550
1000 ADAT=.FALSE.                                                       1560
    IF(ADD) ADAT=.TRUE.                                                 1570
    GOTO 5                                                              1580
1010 STOP                                                               1590
    END                                                                 1600
```

```
      FUNCTION XMIN(N,X)                                                10
      DIMENSION X(1)                                                    20
      XMIN=X(1)                                                         30
      IF(N.LT.2) GOTO 100                                               40
      DO 10 I=2,N                                                       50
      IF(X(I).LT.XMIN) XMIN=X(I)                                        60
   10 CONTINUE                                                          70
  100 RETURN                                                            80
      END                                                               90
```

```
      FUNCTION XMAX(N,X)                                                10
      DIMENSION X(1)                                                    20
      XMAX=X(1)                                                         30
      IF(N.LT.2) GOTO 100                                               40
      DO 10 I=2,N                                                       50
      IF(X(I).GT.XMAX) XMAX=X(I)                                        60
   10 CONTINUE                                                          70
  100 RETURN                                                            80
      END                                                               90
```

```
      SUBROUTINE AVERG(X,Y,N,XA,XB,YAV)                                 10
C                                                                       20
C         CALCULATION OF UNWEIGHTED AVERAGE IN A GIVEN RANGE WHERE      30
C         A FUNCTION Y(X) IS GIVEN BY A DISCRETE NUMBER OF POINTS.      40
C         LINEAR INTERPOLATION IS USED BETWEEN THESE POINTS.            50
C                                                                       60
C         X(K): ABSCISSAE                                               70
C         Y(K): ORDINATES                                               80
C         N   : NUMBER OF CO-ORDINATE PAIRS                             90
C         XA  : LOWER LIMIT OF AVERAGING INTERVAL                       100
C         XB  : UPPER   "   "      "         "                        110
C         YAV : UNWEIGHTED AVERAGE                                      120
C                                                                       130
      DIMENSION X(1),Y(1)                                               140
C                                                                       150
      YAV=0.                                                            160
C         RETURN UNLESS THE RANGES XA...XB AND X(1)...X(N) OVERLAP      170
      IF(XA.GE.XB.OR.XA.GE.X(N).OR.XB.LE.X(1))RETURN                    180
C         RETURN IF NOT ALL X(K) ARE DIFFERENT AND IN ASCENDING ORDER   190
      IF(N.EQ.1) GO TO 2                                                200
      DO 1 K=2,N                                                        210
      IF(X(K).LE.X(K-1))RETURN                                          220
    1 CONTINUE                                                          230
C         FIND SUBSCRIPT KA OF LARGEST X(K) NOT EXCEEDING XA            240
    2 DO 3 K=1,N                                                        250
      KA=K-1                                                            260
      IF(X(K).GT.XA)GO TO 4                                             270
    3 CONTINUE                                                          280
C         FIND SUBSCRIPT KB OF LARGEST X(K) NOT EXCEEDING XB            290
```

```
    4 DO 5 K=1,N                                                          300
      KB=N+1-K                                                            310
      IF(X(KB).LT.XB)GO TO 6                                              320
    5 CONTINUE                                                            330
C         CALCULATE ABSCISSAE YA,YB FOR ORDINATES XA,XB                   340
    6 IF(KA.EQ.0)YA=Y(1)                                                  350
      IF(KA.GT.0)YA=Y(KA)+(Y(KA+1)-Y(KA))*(XA-X(KA))/(X(KA+1)-X(KA))      360
      IF(KB.EQ.N)YB=Y(N)                                                  370
      IF(KB.LT.N)YB=Y(KB)+(Y(KB+1)-Y(KB))*(XB-X(KB))/(X(KB+1)-X(KB))      380
C         CALCULATE AVERAGE                                               390
      IF(KA.LT.KB)GO TO 7                                                 400
C         (1) NO MESH POINTS BETWEEN XA AND XB                            410
      YAV=0.5*(YA+YB)                                                     420
      RETURN                                                              430
C         (2) AT LEAST ONE MESH POINT BETWEEN XA AND XB                   440
    7 S=(Y(KA+1)+YA)*(X(KA+1)-XA)+(YB+Y(KB))*(XB-X(KB))                   450
      IF(KB.EQ.KA+1)GO TO 9                                               460
      K1=KA+1                                                             470
      K2=KB-1                                                             480
      DO 8 K=K1,K2                                                        490
    8 S=S+(Y(K+1)+Y(K))*(X(K+1)-X(K))                                     500
    9 YAV=0.5*S/(XB-XA)                                                   510
      RETURN                                                              520
      END                                                                 530
```

```
      SUBROUTINE NEULCC(NR,NARG,NAMES,X)                                   10
      COMMON/INOUT/KOUT                                                    20
      REAL*8 NAMES(1),NAM,FMT(2)/'(42A1,I5',' ',FMT2/',7A4)'/,FMT3/',7F    30
     1 12.5)'/,FMTF(2)                                                     40
      DIMENSION NARG(1),X(1),XX(7),XXA(7)                                  50
      LOGICAL*1 TXT(41),LNAM(1),LR                                         60
      INTEGER*2 AR,BL/'  '/,AREQ                                           70
      EQUIVALENCE(NAM,LNAM(1)),(AR,LR),(FMTF(1),FMT2),(FMTF(2),FMT3)       80
      NREW=0                                                               90
      NR=1                                                                100
      NAM=NAMES(1)                                                        110
      AR=BL                                                               120
      LR=LNAM(1)                                                          130
      CALL CONVY(LNAM(2),IR,3HI5 ,1HI)                                    140
      WRITE(6,699) IR,AR                                                  150
  699 FORMAT(/T10,'IR=',I5,'    AR=',A2)                                  160
      IFIL=NARG(4)                                                        170
      IFOR=NARG(5)                                                        180
      WRITE(6,698) IFIL,IFOR                                              190
  698 FORMAT(T10,'IFIL=',I2,'    IFOR=',I2)                               200
      BACKSPACE IFIL                                                      210
      IF(NARG(1).GE.3) BACKSPACE IFIL                                     220
      FMT(2)=FMTF(IFOR)                                                   230
      READ(IFIL,FMT,END=10) TXT,AREQ,IREQ,XX                              240
      IF(AR.NE.AREQ.OR.IR.NE.IREQ) GOTO 15                                250
      IF(NARG(1).LT.3.OR.XX(1).GT.NAMES(3)) GOTO 10                       260
      GOTO 16                                                             270
   10 IF(NREW.NE.0) GOTO 1001                                             280
      REWIND IFIL                                                         290
      NREW=NREW+1                                                         300
   15 READ(IFIL,500,END=10) AREQ,IREQ                                     310
  500 FORMAT(T42,A1,I5)                                                   320
      IF(AR.NE.AREQ.OR.IR.NE.IREQ) GOTO 15                                330
      BACKSPACE IFIL                                                      340
      READ(IFIL,FMT) TXT,AREQ,IREQ,XX                                     350
   16 WRITE(KOUT,600) TXT,AREQ,IREQ,IFIL                                  360
  600 FORMAT(//10X,42A1,I5,'  FOUND ON UNIT ',I2//)                       370
      X(1)=XX(1)                                                          380
      J=2                                                                 390
      IF(NARG(2).NE.2) GOTO 22                                            400
      X(2)=XX(2)                                                          410
      J=3                                                                 420
   22 X(J)=XX(6)                                                          430
      IF(NARG(3).NE.2) GOTO 24                                            440
      J=J+1                                                               450
      X(J)=XX(7)                                                          460
   24 RETURN                                                              470
 1001 CONTINUE                                                            480
      WRITE(KOUT,601) NAMES(1),IFIL                                       490
  601 FORMAT('      ',A8,' NOT FOUND ON UNIT=',I2,' OR')                   500
      NR=0                                                                510
      RETURN                                                              520
      END                                                                 530
```

```
      SUBROUTINE NEUNXT(NR,NARG,NAMES,X)                                   10
      REAL*8 NAMES(1),REQ,FMT1(2)/'(T42,A6,',',7A4)    '/,                 20
     1FMT2(2)/'(T42,A6,',',7E12.5) '/                                      30
      DIMENSION NARG(1),X(1),XX(7)                                         40
      IFIL=NARG(4)                                                         50
      NR=1                                                                 60
      IF(NARG(5).EQ.2) GOTO 5                                              70
      READ(IFIL,FMT1,END=1001) REQ,XX                                      80
      GOTO 10                                                              90
    5 READ(IFIL,FMT2,END=1001) REQ,XX                                     100
   10 IF(REQ.NE.NAMES(1)) GOTO 1002                                       110
      X(1)=XX(1)                                                          120
      J=2                                                                 130
      IF(NARG(2).NE.2) GOTO 22                                            140
      X(2)=XX(2)                                                          150
      J=3                                                                 160
   22 X(J)=XX(6)                                                          170
      IF(NARG(3).NE.2) GOTO 24                                            180
      J=J+1                                                               190
      X(J)=XX(7)                                                          200
   24 RETURN                                                              210
 1001 CONTINUE                                                            220
 1002 CONTINUE                                                            230
      NR=0                                                                240
      RETURN                                                              250
      END                                                                 260
```

```
      SUBROUTINE RETXS(NARG,NAMES,EMIN,EMAX,X,Y,NUMX,MAXNUM,NR,LDFLOC,LD
     1FNXT)
      COMMON/SIGSAV/Z
      DIMENSION X(1),Y(1),Z(2),NARG(1),W(2)
      REAL*8 NAMES(1),NAMSV(4)
C
C         RETXS RETRIEVES KEDAK-DATA.
C
      ASSIGN 20 TO NST
      I=0
      CALL LDFLOC(NERR,NARG,NAMES,Z)
      IF(NERR.EQ.0) GOTO 30
      NAMZ=NARG(1)
      IF(NAMZ.LE.2) GOTO 3
      DO 2 J=3,NAMZ
    2 NAMSV(J-2)=NAMES(J)
    3 IF(Z(1).LE.EMIN) GOTO 5
      IF(Z(1).GE.EMAX) GOTO 32
      GOTO 21
    5 CALL LDFNXT(NERR,NARG,NAMES,W)
      IF(NERR.EQ.0) GOTO 23
      IF(W(1).LE.EMIN) GOTO 10
      IF(W(1).GE.EMAX) GOTO 36
    7 I=I+1
      X(I)=Z(1)
      Y(I)=Z(2)
      I=I+1
      X(I)=W(1)
      Y(I)=W(2)
      GOTO NST,(20,200)
   10 CALL LDFNXT(NERR,NARG,NAMES,Z)
      IF(NERR.EQ.0) GOTO 26
      IF(Z(1).LE.EMIN) GOTO 5
      IF(Z(1).GE.EMAX) GOTO38
   11 I=I+1
      X(I)=W(1)
      Y(I)=W(2)
   12 I=I+1
      X(I)=Z(1)
      Y(I)=Z(2)
      GOTO NST,(20,200)
   20 CALL LDFNXT(NERR,NARG,NAMES,Z)
      IF(NERR.EQ.0) GOTO 22
      IF(Z(1).GE.EMAX) GOTO 24
      IF(I.EQ.MAXNUM) GOTO 34
   21 I=I+1
      X(I)=Z(1)
      Y(I)=Z(2)
      GOTO 20
C
      ENTRY REPXS(NARG,NAMES,EMIN,EMAX,X,Y,NUMX,MAXNUM,NR)
      I=0
      NAMZ=NARG(1)
      IF(NAMZ.LE.2) GOTO 21
      DO 19 J=3,NAMZ
   19 NAMSV(J-2)=NAMES(J)
      GOTO 21
C
   22 IF(I.LT.1) GOTO 23
      NR=1
      GOTO 198
   23 NR=5
      I=1
      X(I)=Z(1)
      Y(I)=Z(2)
      GOTO 198
C
   24 IF(I.EQ.MAXNUM) GOTO 34
      I=I+1
      X(I)=Z(1)
      Y(I)=Z(2)
      NR=10
      GOTO 200
   26 I=1
      X(I)=W(1)
      Y(I)=W(2)
      NR=5
      GOTO 198
C
   30 NR=3
      GOTO 200
C
   32 NR=4
      I=1
      X(1)=Z(1)
      Y(1)=Z(2)
      GOTO 200
C
   34 NR=2
      GOTO 200
   36 ASSIGN 200 TO NST
      NR=10
      GOTO 7
   38 ASSIGN 200 TO NST
      NR=10
      GOTO 11
  198 IF(NAMZ.LE.2) GOTO 200
      DO 199 J=3,NAMZ
  199 NAMES(J)=NAMSV(J-2)
  200 NUMX=I
      RETURN
      END
```

```
      SUBROUTINE LOCXS(NARG,NAMES,X,Y,NUMX,MAXNUM,NR,LDFLOC,LDFNXT)
      COMMON/SIGSAV/ Z
      DIMENSION X(1),Y(1),Z(2),NARG(1)
      REAL*8 NAMES(1),NAMSV(4)
C
C         LOCXS : ARE RETRIEVAL SUBROUTINES
C         NXTXS : FOR KEDAK DATA.
C
      I=0
      CALL LDFLOC(NERR,NARG,NAMES,Z)
      IF(NERR.EQ.0) GOTO 30
   19 NAMZ=NARG(1)
      IF(NAMZ.LE.2) GOTO 21
      DO 10 J=3,NAMZ
   10 NAMSV(J-2)=NAMES(J)
      GOTO 21
   20 CALL LDFNXT(NERR,NARG,NAMES,Z)
      IF(NERR.EQ.0) GOTO 22
      IF(I.EQ.MAXNUM) GOTO 34
   21 I=I+1
      X(I)=Z(1)
      Y(I)=Z(2)
      GOTO 20
C
      ENTRY NXTXS(NARG,NAMES,X,Y,NUMX,MAXNUM,NR)
      I=0
      GOTO 19
C
   22 NR=1
      IF(NAMZ.LE.2) GOTO 200
      DO 23 J=3,NAMZ
   23 NAMES(J)=NAMSV(J-2)
      GOTO 200
   30 NR=3
      GOTO 200
   34 NR=2
      GOTO 200
  200 NUMX=I
      RETURN
      END
```

```
      SUBROUTINE ORDNEN(KMAX,FELD,WERT)
      DIMENSION FELD(KMAX),WERT(KMAX)
C
C ORDNEN ORDNET (FELD,WERT) NACH WACHSENDEN ARGUMENTEN VON FELD
C
      IF(KMAX.LT.2) GOTO 100
      DO 99 K=2,KMAX
      M=K-1
      IF(FELD(K).GE.FELD(M)) GOTO 99
      R=FELD(K)
      S=WERT(K)
   10 M=M-1
      IF(M.EQ.0) GOTO 20
      IF(R.LT.FELD(M)) GOTO 10
   20 IA=M+1
      IE=K-1
      J=IE
      DO 30 I=IA,IE
      FELD(J+1)=FELD(J)
      WERT(J+1)=WERT(J)
   30 J=J-1
      FELD(IA)=R
      WERT(IA)=S
   99 CONTINUE
  100 RETURN
      END
```

```
      SUBROUTINE EQUEN(IMAX,E,S)
      DIMENSION E(2),S(2)
      DATA KOUT/6/
C
C EQUEN BEHAELT VON MEHRFACH VORKOMMENDEN E-WERTEN NUR DEN JEWEILS
C          LETZTEN.E MUSS GEORDNET SEIN.
C
      IF(IMAX.LT.2) GOTO 20
      M=0
      DO 10 I=2,IMAX
      IF(E(I).NE.E(I-1)) GOTO 10
      WRITE(KOUT,600) E(I),E(I-1),S(I-1)
  600 FORMAT(20X,'ZWEI GLEICHE ENERGIEWERTE:',E12.5,' ERSTES WERTEPAAR W
     1IRD AUSGESONDERT:',2E12.5)
      M=1
      E(I-1)=E(I)
      S(I-1)=S(I)
      L=I+1
      GOTO 11
   10 CONTINUE
      GOTO 20
   11 DO 15 I=L,IMAX
      IF(E(I).NE.E(I-1)) GOTO 12
      WRITE(KOUT,600) E(I),E(I-1),S(I-1)
      M=M+1
   12 E(I-M)=E(I)
      S(I-M)=S(I)
   15 CONTINUE
      IMAX=IMAX-M
   20 RETURN
      END
```

```
      SUBROUTINE SPACE(X,LX,NMAX)                                          10
      DIMENSION X(1)                                                       20
C---GET WORKING AREA FOR AVERAGE.                                          30
      REAL*8 DATE,ZEIT                                                     40
      COMMON/INOUT/ KOUT                                                   50
      CALL FREESP(KB)                                                      60
C===RESERVE 30 % BUFFER AREA, AT MAXIMUM 40KBYTE.                          70
      LBUF=(3*KB)/10                                                       80
      IF(LBUF.GT.40) LBUF=40                                               90
      KF=KB-LBUF                                                          100
C===EACH KBYTE CAN ACCOMODATE FOR 128 DATA POINTS.                        110
      NMAX=KF*128                                                         120
      NBYTE=KF*1024                                                       130
      CALL XTAREA(K1,NBYTE,K3,X)                                          140
      IF(NBYTE.EQ.0) GOTO 100                                             150
      LX=(K1-K3)/4+1                                                      160
      CALL DATUM(DATE,ZEIT)                                               170
      WRITE(KOUT,601) DATE,ZEIT,LBUF,NMAX,K3                              180
  601 FORMAT(1H1//////////2(/10X,70('*'))/                                190
     120X,'YOU ARE USING PROGRAMME AVERAGE'//                             200
     220X,'DATE:',A8/20X,'TIME:',A8/20X,'YOUR BUFFER AREA IS ',I3,        210
     3' KBYTES'/20X,'YOUR WORKING AREA ',I6,' DATAPOINTS, ALLOCATED AT',  220
     4Z9/2(/10X,70('*')))                                                 230
      RETURN                                                              240
  100 WRITE(KOUT,602)                                                     250
  602 FORMAT(//' WHEN ATTEMTING STORAGE ALLOCATION'/                      260
     1' AN ERROR OCCURED IN ROUTINE "SPACE".')                            270
      STOP                                                                280
      END                                                                 290
```

3.2 Calculation of composite cross sections (CALCUL)

Contents:

3.2.1 Short introduction

Composite cross sections are to be understood here as quantities obtained by performing arithmetic operations (addition, subtraction, division etc.) on particular partial cross sections. Important examples are

- the total cross section, i.e. the sum of all partial cross sections.

$$\sigma_T = \sum_x \sigma_x,$$

- the absorption cross section

$$\sigma_a = \sigma_\gamma + \sigma_f + \sigma_p + \sigma_\alpha,$$

- the nonelastic cross section

$$\sigma_{nonel} = \sigma_T - \sigma_n,$$

- the capture-to-fission ratio

$$\alpha = \sigma_\gamma : \sigma_f.$$

The program CALCUL was written as a tool for the calculation of these and other composite cross sections from angle-integrated partial cross sections. The following Computer Abstract describes its main features. A detailed description will be published as a separate KFK report.

COMPUTER PROGRAM ABSTRACT 6/76 CALCUL

1. NAME OR DESIGNATION OF PROGRAMME: CALCUL is a program system to calculate cross section quantities. To insert into the KEDAK library neutron cross sections data evaluated elsewhere it is necessary to (re-)calculate them and arrange them according to KEDAK conventions.

2. COMPUTER FOR WHICH THE PROGRAMME IS DESIGNED AND OTHERS UPON WHICH IT IS OPERABLE: IBM/370-168 with TSO, MVT

3. NATURE OF PHYSICAL PROBLEM SOLVED: CALCUL processes data from an external source and/or data from the KEDAK library. A control input in form of a command language controls in which manner the data are processed. The output is a data set consisting of ADD records and DROP records that tell the KEDAK Management program which data are to be added to the library and which data are to be deleted from the KEDAK library.

4. METHOD OF SOLUTION: CALCUL assists the user in the general problem of exercising the basic mathematical operations upon functions depending on one variable (energy). Thus CALCUL simulates a desk calculator operating on functions instead of single numerical values; if necessary interpolation is performed.
   The operations:

   1. operation "+" ADD command
      R = R + y
      R - the current result, is equal to zero at the start of the calculation
      R,y are data arrays

   2. operation "-" SUBTRACT command
      R = R - y

   3. operation "." MULTIPLY command
      R = R * y

   4. operation "/" DIVIDE command
      R = R / y

   5. operation ETA calculation
      R = 1/(1+R)

COMPUTER PROGRAM ABSTRACT 6/76 CALCUL

Furthermore the additional facility to perform the calculation of composed cross sections is provided in CALCUL. The composed cross sections are calculated with the aid of the single operations but the control in this case is performed by a subprogram instead of the control input for the single commands.
The commands and formulas for calculating of the composed cross section values:

ALPHA1	$\alpha = \sigma_\gamma/\sigma_f$
ALPHA2	$\alpha = (\bar{\nu}/\eta)-1$
ETA1	$\eta = \bar{\nu}/(1+\alpha)$
ETA2	$\eta = \bar{\nu}*\sigma_f/(\sigma_f+\sigma_\gamma)$
SGA1	$\sigma_a = \sigma_\gamma+\sigma_f+\sigma_p+\sigma_\alpha$
SGG1	$\sigma_\gamma = \sigma_f*\alpha$
SGG2	$\sigma_\gamma = \sigma_f*((\bar{\nu}/\eta)-1)$
SGG3	$\sigma_\gamma = \sigma_a-\sigma_f-\sigma_p-\sigma_\alpha$
SGI1	$\sigma'_n = \sigma_x-\sigma_\gamma-\sigma_p-\sigma_{2n}-\sigma_\alpha-\sigma_{3n}-\sigma_f$
SGN1	$\sigma_n = \sigma_T-\sigma_x$
SGT1	$\sigma_T = \sigma_n+\sigma_x$
SGTR1	$\sigma_{trans} = \sigma_T-\sigma_n*\bar{\mu}_L$
SGX1	$\sigma_x = \sigma_T-\sigma_n$
SGX2	$\sigma_x = \sigma_{n'}+\sigma_\gamma+\sigma_f+\sigma_p+\sigma_\alpha+\sigma_{2n}+\sigma_{3n}$

5. RESTRICTIONS ON THE COMPLEXITY OF THE PROBLEM: KEDAK library format is prerequisite. The data from the external source are expected in formatted records, each including one energy value and one cross section value. Storage allocation is handled dynamically with the aid of the XTAREA subroutine (see reference 2). No restriction on number of energy mesh points. If the core storage given is not sufficient, CALCUL has the possibility to evade on disk storage (must be made available to the program by the user through DD-statements in the job control language).

6. TYPICAL RUNNING TIME: The running time is dependent on the number of data points (energy points) processed and on the region made available to the program, usually about 1 ms/data point on IBM/370-168.

COMPUTER PROGRAM ABSTRACT 6/76 CALCUL

7. UNUSUAL FEATURES OF THE PROGRAMME: CALCUL could be processed as a background job in batch processing or as a foreground job on a terminal in TSO.

8. RELATED AND AUXILIARY PROGRAMMES: CALCUL is written in modular form with many subroutines to facilitate the addition of new options and to permit the generation of a compact overlay structure.
   The subroutines of CALCUL are comprised in the following modules:
   1. Control module
   2. Operation code definition package - OPDEF
   3. Control input processing package - PROCINP
   4. Calculation package - CALPAC
   5. Cross section formula calculation package - CROSSEC
   6. Data management of the auxiliary direct access data set - DATAMAN
   7. Output editing package (KEMA-input-format data set) - OUTPUT
   8. Processing control input in card image format - CARDIM

   The retrieval packages RETPAC and LDFPAC are used to read the data from the KEDAK library. The following subroutines are external references:
   XTAREA and FREESP - to handle the dynamic storage allocation
   CONVY and STRING - to convert floating point and integer data to alphameric representation and v.v.
   DEFI and DINF - to make the DEFINE FILE statement dynamical
   DDTEST - to test for DD-cards

9. STATUS: In use

10. REFERENCES:

   1. I. Langner, R. Meyer
      CALCUL - a program system to calculate cross section quantities and arrange data for input to KEMA.
      KFK, to be published.
   2. I. Langner, R. Meyer
      IDFPAC/LDFPAC - two retrieval packages for the Karlsruhe Evaluated Nuclear Data Library, KFK, to be published
   3. W. Höbel
      XTAREA, REXTAR - dynamische Dimensionierung von FORTRAN-Feldern, KFK, to be published

4. G. Arnecke, H. Bachmann
DEFI, DINF dynamisches DEFINE FILE, KFK, to be published

5. G. Arnecke
DDTEST - Benutzte Dateien, KFK, to be published

6. B. Krieg
Handling and service programs for the Karlsruhe Nuclear Data File KEDAK, KFK 1725, June 1973

7. R. Meyer
RETPAC - A user oriented retrieval package for use with the Evaluated Nuclear Data Library KEDAK, GfK, Internal Report

8. H. Blesene
CONVY - FORTRAN-Unterprogramm für die IBM/360 zur Umwandlung von in maschineninterner bzw. in alphanumerischer Darstellung vorliegenden Fest- und Gleitkommazahlen in alphanumerische bzw. maschineninterne Darstellung, Internal Report, GfK, 1971

11. MACHINE REQUIREMENTS: Disk storage on a direct-access device; TSO terminal. Without overlay structure the program requires 240 K core storage.

12. PROGRAMMING LANGUAGE USED: FORTRAN IV

13. OPERATING SYSTEM OR MONITOR UNDER WHICH PROGRAMME IS EXECUTED:
OS 360/370 MVT and TSO

14. ANY OTHER PROGRAMMING OR OPERATING INFORMATIONS OR RESTRICTIONS:
With the overlay structure CALCUL requires 130 K bytes core storage. The storage in the GO-step depends on the number of data points processed. If not enough core storage is available for the temporary data sets, the program tries to evade on auxiliary data sets on disk.

15. NAME AND ESTABLISHMENT OF AUTHOR:
I. Langner, R. Meyer
Gesellschaft für Kernforschung mbH
Institut für Neutronenphysik und Reaktortechnik
Postfach 3640
D-7500 Karlsruhe 1

16. MATERIAL AVAILABLE: Source deck. List of available commands, and of conventions, and syntax of the control input list for a command.

4. PROGRAMS FOR WEIGHTING AND SMOOTHING OF CROSS SECTION DATA

4.1 Program SIGPLO

Contents:

4.1 Program SIGPLO

The following program description is based on an unfinished internal report by R. Meyer, the author of the program /1/.

4.1.1 General characteristics

Smoothing and fitting, interpolation, tabulation and plotting of numeric data are constantly recurring tasks in nuclear data evaluation. The very large data sets typical for modern neutron data work make utilization of computer methods mandatory. The program SIGPLO was developped in response to this need. The options offered by SIGPLO will be discussed in the following order:

(1) data retrieval
(2) processing of retrieved data (interpolation, smoothing)
(3) output (printing, plotting, punching,...)
(4) special features

The various options and their default values are summarized for quick reference in Table 1 below.

4.1.2 Data retrieval

The SIGPLO user may address three different sources of data:

(1) data in KEDAK format,
(2) NEUDADA "transmission" tapes,
(3) other formatted (FORTRAN-readable) data.

Each set of data thus retrieved forms the "current data set" on which SIGPLO will operate. The current set may be extended by merging it with data from the same or other sources to form a new "current data set". Due to the use of fixed dimensions the size of the "current data set" must not exceed 1000 data points. Data need not be sorted - they are sorted after retrieval, if necessary. Every time a data set is read it either replaces the "current set" or, if merging is requested, it is merged with the "current set". In both cases the old "current set" is lost and cannot be referenced but by retrieving it again.

The examples given in Sect. 4.1.2 are valid for IBM computer installations working under OS and ASP.

4.1.2.1 Retrieval from files in KEDAK format

Data in KEDAK format may be retrieved from up to five different KEDAK-type data libraries, so that for instance several versions of KEDAK can be compared with each other or with data translated from other libraries, e.g. ENDF/B. Retrieval from the KEDAK file is performed with the program package LDFPAC /2/.

In order to address a KEDAK-type library the user must specify the name of the library and the KEDAK name (key) of the data (cf. /3/). He must provide

(1) a SETUP card for the device that contains the library, as required by IBM-ASP, if necessary;

(2) DD control card(s) describing the library data set, as specified in the IBM-OS job control language manual.

Examples (underscores denote blanks):

```
//G.FT01F001_DD_UNIT=2313,VØL=SER=GFK050,DISP=SHR
//              DSN=KEDAK3
```

will address the library KEDAK3 on the disk pack named GFK050. The data will be available to SIGPLO via the ft-number PLEH=1, cf. (3) below;

```
//G.FT02F001_DD_UNIT=3330,VØL=SER=NUSICE,DISP=SHR,
//              DSN=STRUMAT.ENDFB3
```

will address the data library STRUMAT.ENDFB3 on the disc pack NUSICE; data will be available to SIGPLO through PLEH=2.

(3) program control input describing names and energy range of the data that are to be retrieved:

```
_&KEDAK_PLEH=...,NAMZ=...,NAMEN=...,TØ=...,FRØM=...,
_data-processing and output options,&END
```

where PLEH is the input device number (ft-number),

NAMZ is the number of KEDAK names necessary to address the requested data (default option: NAMZ=2), cf. /3/,

NAMEN are these KEDAK names, e.g. for σ_T of Na-23 one would write NAMEN='NA23', 'SGT' or alternatively NAMEN(1)='NA23', NAMEN(2)='SGT'. The KEDAK names must be written in the same sequence as in /3/;

FRØM minimum energy,

TØ maximum energy.

The data-processing and output options are described in Sects. 4.1.3 - 4.1.5 below.

4.1.2.2 Retrieval from NEUDADA tapes

NEUDADA tapes contain experimental data from the common data base of the "four-centre network" of neutron data centres (National Neutron Cross Section Center, Brookhaven, USA; Centre de Compilation de Données Nucléaires of NEA/OECD, Saclay, France; Nuclear Data Section of IAEA, Vienna, Austria; Centr po Jadernym Dannym, Obninsk, USSR). These (7 or 9 track) tapes are sent by CCDN to users in non-American OECD countries upon request.

In the NEUDADA formats (neutron data under direct access) each data point is coded in one record together with its uncertainty, other related data and a data set identifier. These identifiers (accession numbers) are used by SIGPLO to distinguish the various data sets that may occur on one tape. The user must

- know on which file his data actually reside (tape marks written at CCDN are not consistent with all IBM operating systems) and whether they are written in CCDN "transmission" format or in CCDN internal format;

- check whether his data have got unique accession numbers or whether they are distributed among several accession numbers, which frequently may happen. In view of this possibility SIGPLO offers a special option to combine several sets of data;

- make sure the requested data are complete. If this is not the case SIGPLO allows to add data e.g. on punched cards. At present SIGPLO offers no possibility to delete data points read. The only way to correct faulty data points is to copy-correct the respective data set on the tape.

In order to retrieve data from NEUDADA tapes the user must provide

(1) a SETUP card as required by IBM-ASP;

(2) a DD control card for each file to be addressed, as specified in the IBM-OS job control language manual.

Examples:

```
//G.FT10F001_DD_UNIT=TAPE9,VØL=SER=NDØ501,
//              LABEL=(2,NL,,IN),DISP=(ØLD,PASS),
//              DCB=(BLKSIZE=1320,LRECL=132)
```

will address data on file 2 of the 9-track tape named NDØ501. Data will be available to SIGPLO via ft-number BDEH=10, cf. (3) below;

```
//G.FT04F001_DD_UNIT=TAPE7,VØL=SER=ND2354,
//              LABEL=(2,NL,,IN),DISP=(ØLD,PASS),
//              DCB=(BLKSIZE=1320,LRECL=132,TRTCH=ET)
```

will address data on file 2 of the 7-track tape named ND2354. Data will be available to SIGPLO via BDEH=4, cf. (3) below. The data on these two tapes are available to SIGPLO simultaneously, two tape drives are required.

(3) Program control input:

```
_&NEUDAD_BDEH=...,LTR=...,ACCNR=...,SEQN=...,FØDATA=...,
_FRØM=...,TØ=...,data-processing and output options,&END
```

where

BDEH gives the FT number of the tape unit

LTR specifies the letter, if any, that precedes the accession number supplied by CCDN, e.g. LTR='E' if data set E00012 is desired. The default option is LTR='0', hence LTR need not be specified for data set 000012.

ACCNR is the (numeric) accession number supplied by CCDN, e.g. ACCNR=12 for data set E00012.

SEQN specifies whether the accession number is in line with the expected ascending order on tape (SEQN=T) or not (SEQN=F), so that the tape can be rewound where necessary.

Example

Suppose the sequence of data sets on tape is

...U00003,E00002,...E00013,E00014,...U00015,...

whereas they are to be retrieved in the order

E00013,E00014,U00015,E00002,U00003.

In this case all sets can be retrieved with SEQN=T expect U00003 for which SEQN=F signals that it occurs on tape not in the expected order (i.e. after E00002).

FØDATA indicates whether the NEUDADA tape is written in CCDN "transmission" format (FODATA=T, default option) or in CCDN internal format (FODATA=F).

FRØM,TØ are minimum and maximum energy.

Other data-processing and output options are explained below.

4.1.2.3 Retrieval of other formatted data

Formatted input data in the present context means FORTRAN-readable data (as explained e.g. in the IBM FORTRAN IV manuals) with the following restrictions:

- Each set of numeric data records (data set) must be preceded by a comment record with arbitrary free text. This text is read and printed so that the user can check the input sequence.

- Each data point must be completely contained in one record. On the other hand one record may contain several data points.

- The co-ordinates of each data point must be represented in the order (1) energy, (2) cross section (or other energy-dependent quantity) (3) uncertainty (optional). Other information may be interspersed with these data. The user must specify the reading format accordingly, e.g. by spacing over characters not to be read as data.

Records may be cards or card images on tape or other rigidly structured character sequences on any input device.

For retrieval of a formatted data set the user must provide

(1) a setup card for the device from which the data are to be read, if necessary,

(2) a DD card describing the device and the data set.

Examples

```
//G.FT16F001_DD_UNIT=2314,VØL=SER=NUSYSO,
//              DISP=SHR,DSN=FØRM,INP1
```

would provide formatted data with unit reference number (ft-number) 16. Data are read from a disc named NUSYSO and data set FØRM.INP1. Data set characteristics need not be given, they are provided by the operating system.

```
//G.FT11F001_DD_UNIT=TAPE9,VØL=SER=TAPE72,
//              DISP=(ØLD,PASS),LABEL=(1,NL),
//              DCB=(RECFM=FB,LRECL=30,BLKSIZE=3000)
```

specifies file 1 of the unlabelled 9-track tape named TAPE72 as input. Since the tape has no label, data set characteristics must be supplied.

(3) program control input describing device, format, energy range and other characteristics of the data set:

```
_&KARTEN_KARTEH=...,FMT=...,ANZ=...,
_FRØM=...,TØ=...,EINHTX=...,EINHTY=...,
_REWD=...,data-processing and output options,&END
```

where

KARTEH is the input device reference number. If punched cards are supplied together with control input, KARTEH=9 must be specified (not KARTEH=5) and the cards (preceded by the comment card) must immediately follow the respective control card.

FMT gives the reading format, e.g.

FMT='(15X,E15.5,10X,E15.5)' or
FMT='(8F10.5)'.

ANZ determines the number of data points to be read (as distinct from points actually utilized) as follows:

The requested data set is searched for the first record beginning with an energy ≥ FRØM*0.999999. If this is the first record of the whole data set, reading starts there, otherwise reading starts at the preceding record (because that may contain, in addition to the first energy, other unchecked energies that possibly exceed the limit FRØM*0.999999). Reading stops as soon as ANZ data points are read or an END OF FILE condition occurs. Subsequently SIGPLO searches for the first data point read with energy exceeding TØ*1.000001 and deletes it as well as all subsequent points. Then all data points with energies less than FRØM*0.999999 are discarded. Finally the remaining data points are rearranged in order of ascending energy. Note that ANZ determines the number of data points read, the actual number of data points utilized by SIGPLO can be smaller due to the deletion of points with energies outside the requested range. Since an exact determination of the number of data points to be read may be inconvenient, the author suggests that the user

either supplies only one data point per record, in which case exactly ANZ cards (= data points) are read starting with the card preceding the first data point with energy ≥ FRØM, counted from the current position within the data set,

or matches ANZ exactly with the number of data to be read, if the data set contains only points within the given energy limits,

or uses a different unit reference number for each data set, utilizing the END OF FILE condition as a natural delimiter.

The example below illustrates this.

FRØM,TØ have their usual meaning of minimum and maximum energy.

EINHTX = { 'EV' / 'KEV' / 'MEV' } specifies energy units,

EINHTY = { 'MB' / 'BARN' } specifies cross section units.

Data will always be converted to eV or b internally. Of course, conversion could be obviated by proper scale factors in the FMT input format specification.

REWD may be specified as T (or F) which causes rewinding (or not) before retrieval starts (with the exception of the control input device for which REWD is ignored).

Warning: - ANZ should not exceed 1000.

- Although input data need not be sorted, the user should be aware of the fact that they are sorted only after they have been read. Hence data after the first energy $\geq$ TØ will not be transmitted even though the respective card(s) might have been read.

Example:

```
//G.SYSIN_DD_*
RT__TEST_INPUT
_&KARTEN_KARTEH=9,FMT='(10F10.2)',ANZ=2,
_FROM=0.,TO=100.,&END
C  INPUT FRØM CØNTRØL UNIT
1.       0.37    24.2      0.81
_&KARTEN_KARTEH=14,FMT='(2F10.2)',ANZ=50,
_FRØM=24.6,TØ=82.9,&END
_&KARTEN_KARTEH=11,FMT='(2F10.2)',ANZ=50,
_FRØM=24.6,TØ=82.9,&END
//G.FT11F001_DD_*
C   INPUT FROM UNIT 11
13.6       0.37
17.77      1.44
21.88       .              (first card read)
24.92       .
23.11       .
41.88      etc.
35.97       .
46.1        .
17.15       .
53.2        .
57.2        .
69.44       .
91.85       .
82.1        .              (last card read)
//G.FT14F001_DD_*
C   INPUT FROM UNIT 14
25.99       .              (1st card read)
44.12       .
57.31       .
80.50       .              (last card read)
//
```

Explanation

Let us for the moment disregard the card RT__... which will be explained in Sect. 4.1.3.2 below. The first input data set is read from unit 9. The user took care that only one data card is read by specifying with ANZ=2 the exact number of data to be read. Thus two data points are transferred to SIGPLO:(1.,0.37) and (24.2,0.81).

The second data set is read from unit 11. Reading will start with energy 21.88 eV since this is the last one less than FRØM. Reading will stop after energy 82.1 eV caused by the END OF FILE condition. The points with energies 21.88 eV and 23.11 eV are then deleted, these energies being less than FRØM. Likewise the point with energy 91.85 eV (the first energy exceeding TØ) and all subsequent points are deleted. Thus the points available to SIGPLO after sorting are those with energies 24.92, 35.97, 41.88, 46.1, 53.2, 57.2, and 69.44 eV.

The third data set is transferred from unit 14 to SIGPLO without deletions.

Note that FMT,FRØM and TØ could have been omitted from the last control cards since the previous values are the same and would be kept by default. This, however, does not hold for ANZ. (See Table 1, footnote). Since SIGPLO assumes the units eV and barn by default, EINHTX and EINHTY can be omitted.

The energy-sorting feature allows for easy addition of other data to a punched data set. For instance, if a point at 29.7 eV is to be added to the third set of data one could insert the corresponding card simply at the end of (or some other place within) the card deck containing the third data set.

Merging of data sets is possible with the option

DATADD = { T merge "current" set and newly retrieved set,
         { F replace "current" set by newly retrieved set (default option)

This option can be used in the control input lists _&NEUDAD_... or _&KARTEN_... specifying the newly retrieved set.

4.1.3 Processing of retrieved data

SIGPLO offers two possibilities to process retrieved data: interpolation and smooth-curve fitting (smoothing). The results can be plotted, printed and punched on cards (or written onto some other output medium).

4.1.3.1 Linear and spline interpolation

The aim of interpolation is to run a curve through a given set of points known as "nodes" by specifying an interpolation law. The interpolation options of SIGPLO are most useful for drawing a continuous line through a set of data points on a plot or for retabulation on a different set of nodes ("new grid"). Thus initial data set (nodes), interpolation law and new grid must be specified:

(1) The initial data set is defined by an appropriate retrieval as described in Sect. 4.1.2.

(2) The interpolation law is defined by the key-word KURVE that may be employed in any of the three input lists &KEDAK..., &NEUDAD..., &KEDAK... (cf. Sect. 4.1.2). The options are

$$\text{KURVE} = \begin{cases} \text{'NEIN'} & \text{no interpolation (default option),} \\ \text{'LINE'} & \text{linear interpolation,} \\ \text{'SPLI'} & \text{cubic spline interpolation.} \end{cases}$$

Linear interpolation means that between adjacent nodes (x_i,y_i) and (x_{i+1},y_{i+1}) interpolated values (x,y) are calculated according to the interpolation law

$$\frac{y-y_i}{x-x_i} = \frac{y_{i+1}-y_i}{x_{i+1}-x_i} \quad .$$

Cubic spline interpolation means that the interpolation law $y = f(x)$ between each pair of adjacent nodes is a cubic, the different cubics being chosen such that f, f' and f'' are continuous at the matching points x_i $(i=2,3,\ldots n-1)$ and $f'' = 0$ at the end points (x_1,x_n).

It also means that the minimum number of nodes is 3.

(3) Output options are as follows:

PLØKU = { T plot curve through original points (default option),
        { F do not plot,

DRUKU = { T print table of interpolated set, on new grid,
        { F do not print (default option),

STAKU = { T "punch" interpolated set, on new grid,
        { F do not "punch" (default option)
             where punch means generation of any
             FORTRAN-readable output (on cards, tape, disc,...)

For the option PLOKU the new grid is automatically generated by SIGPLO. For DRUKU and STAKU, however, the new grid as well as the printing or "punching" format must be supplied by the user as explained in Sect. 4.1.3.3 below.

Notes:

- PLØKU=T causes plotting of only the initial set and belongs, strictly speaking, to Sect. 4.1.4 below. If the interpolated set would have to be plotted, it would have to be written onto some rereadable computer medium, reread with another "control input" list and plotted.

- If plotted output is desired the plot must be opened first by definition of its characteristics with a suitable "control input" list (cf. Sect. 4.1.4 below).

- The new grid must be defined before interpolation starts, i.e. the control input list defining the new grid must precede the control input list that requests interpolation.

- Similarly the control input list specifying the format for printing or other formatted-output production must precede the control input list that requests the output.

- If x_i are the initial grid points and x'_j the new grid points $x_1 \leq x'_1$ and $x_{last} \geq x'_{last}$ must hold.

- The linear interpolation and spline options are useful if cross sections are to be retabulated on a new or denser grid, e.g. for KEDAK. Splines, however, are sometimes quite sensitive to the exact position of the nodes and a smooth second derivative (curvature) may be difficult to achieve. For further details of spline fitting see /5/.

- If KEDAK cross sections are plotted, the user usually wants a curve without special marking of the nodes. This is achieved by the following combination of options:
  _&KEDAK NAMEN='material','type',FRØM=xx,TØ=xx,
  _KURVE='LINE',PLOKU=T,PLO=F,&END
  (PLOKU=T requests the curve, PLO=F suppresses special marking of the nodes, cf. Sect. 4.1.4).

4.1.3.2 Curve fitting (smoothing)

Curve fitting is possible with SIGPLO in the following restricted sense:
A curve $y=f(x)$ is drawn through a given set of points (x_i,y_i), $i=1,2,\ldots,n$, which is "as smooth as possible". The user sets a limit χ^2 which the weighted sum of squared deviations must not exceed:

$$\sum_{i=1}^{n} w_i\left[y_i-f(x_i)\right]^2 \leq \chi^2 \qquad (1)$$

Within this limit the fitted curve is determined such that

$$\int_{x_1}^{x_n} f''(x)^2 dx = \min \qquad (2)$$

Obviously this kind of fit (smoothing, ref. /6/) requires caution. Of course, one an produce a smooth curve through violently fluctuating data points, but if the fluctuations are genuine the resultant smooth curve is quite unrealistic. Smoothing of discrepant data sets from different experiments can be very difficult. The result will often not be better than an eyeguide curve, the only advantage being that one need not read off numerical values which the computer produces easily. This must be kept in mind before smoothing is requested.

Since frequently a smooth curve is to be drawn through data from different measurements SIGPLO allows merging several input data sets into one set for smoothing. If, for instance, the data sets S_1, S_2,... are to be merged for joint smoothing, one has to indicate this by putting

$$SM\emptyset\emptyset=\chi_j^2$$

into the control input list of S_j, where the χ_j^2 are real numbers chosen in such a way that their sum is just the upper limit of the weighted sum of squared deviations in Eq. (1), $\chi^2 = \sum_j \chi_j^2$. The joint set $S = S_1 + S_2 + \ldots$ is then retained for final smoothing. Merging a retrieved set into this "retained set" can be prevented by using SMØØ='NEIN' (default option) in its control input list. Smoothing is possible only for control input lists of type &NEUDAD and &KARTEN, but not for &KEDAK (KEDAK point data are linearly interpolable and thus smooth by definition).

Smoothing starts as soon as

- the end of control input is reached or
- a card containing RT__ plus free text is found or
- a &SMOWDH control input list is encountered.

An RT card will empty the retained set after smoothing, a &SMOWDH list (explained in more detail below) will not.

The output options for the results of smoothing are as follows:

PLØSM = { T plot the fitted smooth curve (default option*)
F do not plot;

DRUSM = { T print points of the fitted smooth curve,
F do not print (default option*);

STASM = { T "punch" points of the fitted smooth curve,
F do not punch (default option*)
where "punch" means generation of any
FORTRAN-readable output (on cards, tape, disc,...)

*) Note: The default options are in effect only if the options were not already defined in a preceding control input list!

These output options are not needed until smoothing actually starts. This means they could be coded e.g. within the last control input list that is read before smoothing begins. If plotting is requested (PLØSM=T) the plot must also be opened before smoothing starts. The sum total (χ^2, Eq.(1)) of all SMØØ values will be displayed or not as specified by

$$\text{PLØFQU} = \begin{cases} \text{T} & \text{display the squared-error sum } \chi^2, \\ \text{F} & \text{do not display } \chi^2. \end{cases}$$

The weights w_i in Eq. (1) are specified as follows:

(1) All points of the "retained" set get unit weight by default, if weights are not defined explicitly.

(2) Weights are defined explicitly by retrieval of the experimental uncertainties δy_i associated with the data points (x_i, y_i). The weights are thus calculated as $w_i = (\delta y_i)^{-2}$. The experimental uncertainties can be read together with the data, e.g. from the same tape, or separately. In any case reading of experimental errors must be requested in the same input list as reading of the data. One writes SMGEW=T together with specifications of the medium from which errors are to be read, and error units. The options are as follows:

$$\text{\_\&NEUDAD\_ ... SMGEW=T,MEDIUM} = \begin{cases} \text{'TAPE'} \\ \text{'CARD'} \end{cases} \text{,DEINHT} = \begin{cases} \text{'PERC'} \\ \text{'B'} \\ \text{'MB'} \end{cases} \text{, ...}$$

or

$$\text{\_\&KARTEN\_ ... SMGEW=T,EINHTD} = \begin{cases} \text{'PERC'} \\ \text{'B'} \\ \text{'MB'} \end{cases} \text{, ...}$$

In case MEDIUM='TAPE' the errors are read from tape together with the (x_i, y_i) values and the units are converted to barns according to the NEUDADA specifications on tape. Note that DEINHT and EINHTY (see Sect. 4.1.2.2) must then specify the same units. If these specifications are missing, the

user can override the default option by specifying the error units (PERC for percent errors, B or MB for errors in b or mb).

In case MEDIUM='CARD' one must define input unit, input format and number of data to be read by

KARTEH=...,FMT=...,ZAHL=...

as explained in Sect. 4.1.2.3. If ZAHL is less than the number of data points read from tape the last error value read will be applied to all remaining data points. This allows, for example, to give only one error bar if the same error is to be assigned to all data points. In case data points and errors are read from the same medium (same deck of cards, same tape etc.) they must be in the order $\ldots,x_i,y_i,\delta y_i\ldots$

Reading of data errors can be suppressed by SMGEW=F.

A user who wishes to modify weights differently in different energy regions can do so with the control input list

_&RELGEW_ZAHLXR=...,XR=...,PR=...,EINHTX=...,&END

where ZAHLXR is an integer $n \leq 20$ giving the number of modification factors (energy regions);

XR is a list of n real numbers, separated by commas, giving the lower limits of the energy regions;

PR is a list of n real numbers, separated by commas, giving the modification factors;

EINHTX = { 'EV' / 'KEV' / 'MEV' } indicates in which units the XR values are given.

Relative weights are then applied as follows: Weights of all data points in the "retained data set" are multiplied

- by 1 if $x < XR(1)$,
- by PR(i) if $XR(i) < x < XR(i+1)$, $i = 1,2,\ldots n$.

The first attempt to fit a given set of data is often not very successful, smoothing being very much a trial and error method. Usually a number of fits with varying parameters must be performed before the result is satisfactory. SIGPLO offers a convenient possibility to repeat smoothing, varying only selected parameters given in the following control input list:

```
_&SMØWDH_FQU=...,ZAHLXR=...,XR=...,PR=...,EINHTX=...,NEUZEI=...,
_PLØFQU=...,&END
```

where FQU is a real number establishing a new total limit χ^2 for the weighted squared deviations,

ZAHLXR,XR,PR,EINHTX may be used to define new weight modification factors as explained before,

NEUZEI = { T start a new plot with the current plot size parameters,
F draw fitted curve on current plot (if plotting is requested)

PLØFQU = { T plot the FQU value (χ^2) also,
F do not plot the FQU value. (PLØFQU=T can be used only if not more than one smooth curve is drewn onto one plot. Otherwise the numbers would overprint each other).

Summary

The following control input options are related to smoothing:

(1) current data set options

(&NEUDAD,&KARTEN)	SMØØ,PLØSM,DRUSM,STASM,SMGEW
(&NEUDAD)	MEDIUM,DEINHT,KARTEN,FMT,ZAHL
(&KARTEN)	EINHTD

(2) weight modification

(&RELGEW) ZAHLXR,XR,PR,EINHTX

(3) repetition of smoothing

(&SMØWDH) FQU,ZAHLXR,XR,PR,EINHTX,NEUZEI,PLØFQU

Note: - The maximum number of data points in the "retained set" is 1000.

- Smoothing should be used only if the point scatter is expected to be statistical (random). Do not smooth if fluctuations are genuine, e.g. in the resonance range.

- If the point scatter is statistical one expects for a good fit

$$\sum_i w_i (f(x_i)-y_i)^2 = \chi^2 \simeq N,$$

where N is the degree of freedom of the problem, i.e. the number of points minus the number of fit parameters. Hence N is recommended as a starting value for smoothing.

Example

```
RT__SMØØTH ØF TWØ SETS ØN TAPE
_&NEUDAD_BDEH=2,ACCNR=1,LTR='E',SMØØ=1.2,SMGEW=T,&END
_&PLØTGR_XMIN=0.25,XMAX=1.25,XLG=40,YMIN=0.1,YMAX=0.6,YLG=20.,&END
_&NEUDAD_ACCNR=2,SMØØ=2,PLØFQU=T,&END
_&RELGEW_ZAHLXR=1,PR=0.86,&END
_&SMØWDM_FQU=1.,NEUZEI=T,&END
_&SMØWDM_FQU=0.1,&END
_&SMØWDM_FQU=0.04,&END
RT__SMØØTH ANØTHER SET,PLØT ØUTPUT
_&PLØTGR_XMIN=2.,XMAX=18.,XLG=34.,YMIN=1,YMAX=7,YLG=24.,&END
_&NEUDAD_BDEH=2,ACCNR=5,LTR='U',SMØØ=1.22,SMGEW=T,DEINHT='PERC',
_MEDIUM='CARD',ZAHL=1,FMT='(F5.2)',KARTEH=5,STASM=T,&END
2.6
&STAFØR_STAFMT='(1P6E12.5)',&END
&ØUTINT_ZXINT='KEDA',NAMEN='U235','SGT',FRØM=2.,TØ=18.,&END
/*
```

Explanation

The NEUDADA sets E00001 and E00002 are merged and all weights are multiplied by 0.86. The card _&PLØTGR_... defines the dimensions of a plot as explained in Sect. 4.1.4.2 below. Smoothing begins with χ^2 = 1.2+2.0=3.2 as soon as the first _&SMØWDM card is encountered (before its content is read). The result is plotted by default. Then smoothing and plotting is repeated with χ^2 = 1., 0.1 and 0.04. For each χ^2-value the "retained set" and the resulting smooth curve are plotted onto a new plot (with the initially given plot size specifications). The second RT card causes the retained set to be emptied, whereupon a new plot size is specified and a new data set, U00005, is read in. All error bars are then set equal to 2.6 %. A punching format is given (_&STAFØR_...) and the new grid for the punched output is specified (_&ØUTINT_...) as explained in Sect. 4.1.5.2 below. Smoothing of the last data set (U00005) begins when the end of the file is reached (/* card).

4.1.3.3 Fission spectrum average

The SIGPLO program averages cross sections over a standard fission spectrum if the option FISS=T appears in the control input lists _&KEDAK_..., _&NEUDAD_... or _&KARTEN_... (FISS=F is the default option.) The spectrum is taken as

$$\chi(E)dE = 0.4527\ \text{MeV}^{-1}\ \exp\left(-\frac{E}{0.9650\ \text{MeV}}\right)\ \sinh\sqrt{\frac{E}{0.4367\ \text{MeV}}}\,dE$$

which corresponds to ^{235}U fission at low incident energies. By changing a single statement in subroutine QFISS other spectra could be used. The result,

$$\langle\sigma\rangle_{\chi} = \int \sigma(E)\chi(E)dE$$

is printed as well as the mesh points used for the numerical integration.

4.1.4 Plot production

4.1.4.1 Plotter hardware and software employed by SIGPLO

SIGPLO utilizes the following plotting facilities which are available at Karlsruhe:

- an IBM 1130 plotter producing plots 10 in. high and
- a CALCOMP plotter producing plots 24 in. high.

Access to both devices is controlled by PLØTA /7/, a FORTRAN subroutine extensively used at Karlsruhe.

4.1.4.2 Plot size and scale specification

The user should keep in mind the following rules:

- Only one plot can be produced at a time. This means that output cannot be routed to more than one plot at any given time. If two or more plots are required they must be produced sequentially.

- Before output can be routed to a plot, the size, scaling and other characteristics of the plot must have been established.

These rules can be pictured as the following temporal sequence:

```
define plot no. 1
route output to plot no. 1
define plot no. 2
route output to plot no. 2
  :
  : etc.
```

Each plot must be opened by specification of its characteristics in a program control input list as follows:

```
_&PLØTGR_XMIN=...,XMAX=...,XLG=...,YMIN=...,YMAX=...,YLG=...,
_RAST=...,...,EINHTX=...,EINHTY=...,CALP=...,&END
```

XMIN,XMAX — define minimum and maximum value on the x-axis in the units given by EINHTX.

XLG — is the length of the x-axis in units of 0.4 in.=1.016 cm.

YMIN,YMAX — define minimum and maximum value on the y-axis in the units given by EINHTY.

YLG — is the length of the y-axis in units of 0.4 in.=1.016 cm.

RAST = { 'NEIN' / 'LINE' / 'XLØG' } , { 'NEIN' / 'LINE' / 'YLØG' } specifies whether a grid is to be drawn and if so, for which kind of scale. Linear scaling (LINE) produces parallel lines 0.8 in.=2.032 cm apart, while log scaling produces a line at each integer multiple of the current power of ten.

EINHTX = { 'EV' / 'KEV' / 'MEV' } gives the x units,

EINHTY = { 'BARN' / 'MB' } gives the y units.

The plot will be drawn in units of eV and b unless the user does not request a conversion by stating EINHTX and EINHTY explicitly.

CALP = { T / F } specifies whether a CALCOMP plot is requested (T) or not (F) - in the latter case the IBM1130 plotter is used.

Note that

- 2.*(XMAX-XMIN)/XLG and 2.*(YMAX-YMIN)/YLG should be smooth numbers (integers) to produce convenient scales;

- XLG and YLG should be even integers;

- it is not possible to produce logarithmic scales without grid, i.e. RAST='NEIN',... or RAST='...','NEIN' is automatically interpreted as meaning a linear scale.

4.1.4.3 Plot_options

Having opened a plot by specifying its characteristics in a _&PLØTGR control input list the user can route output to it by using one of the following options:

PLØ = { T / F }, PLØKU = { T / F }, PLØSM = { T / F }.

PLØ=T yields a plot of the "current set" of points,

PLØKU=T " " " " an interpolated curve and

PLØKU=T " " " " a fitted smooth curve.

These options may occur in the input lists _&KEDAK_..., _&NEUDAD_..., _&KARTEN_... with the restriction that PLØSM is not allowed for _&KEDAK_. If data points (as distinct from curves) are to be plotted ten different point symbols are available. These symbols are linked to the integers 0,1,...9 as follows:

n	0	1	2	3	4	5	6	7	8	9
IBM 1130 symbol	○	□	◇	+	×	⧖	⋈	△	▽	·
CALCOMP symbol										

SIGPLO uses the following sequence by default:

9,0,3,4,5,6,7,1,8,2, then repeatedly 0,1,2,7,8.

Up to forty point sets can be plotted onto one plot. The user may overrule the default sequence with

PLOZEI = n, where n is the integer associated with the desired symbol, and
PLOZEI = -1 causes return to the default sequence.

PLOZEI is permitted in any one of the three control input lists _&KEDAK, _&NEUDAD, _&KARTEN.

Example

```
RT__FE-PLØTS
_&PLØTGR_YMIN=1.,YMAX=2.,YLG=10.,XMIN=1.E6,XMAX=5.E6,
_XLG=20.,RAST='LINE','LINE',&END
_&KEDAK_NAMEN='FE','SGN',FRØM=0.998E6,TØ=5.002E6,
_PLØ=F,KURVE='LINE',&END
_&KARTEN_KARTEH=12,ANZ=12,KURVE='NEIN',PLØ=T,
_PLØZEI=5,EINHTX='MEV',&END
_&NEUDAD_BDEH=4,PLØZEI=-1,ACCNR=4,LTR='E',&END
_&PLØTGR_YMIN=2.,YMAX=4.,YLG=20.,&END
_&KARTEN_KARTEH=13,ANZ=108,&END
_&NEUDAD_ACCNR=6,PLØZEI=8,&END
_&NEUDAD_ACCNR=8,PLØZEI=-1,&END
_&NEUDAD_ACCNR=7,LTR='U',&END
_&KEDAK_KURVE='LINE',PLØ=F,NAMEN='FE','SGT',&END
```

Explanation

Convenient scaling is achieved for both plots by the use of (2 cm ≃) 0,8 in. ≙ 0.2b and ≙ 0.4 MeV, respectively. The first plot will contain one curve and two sets of points. The point symbol number 5 is explicitly requested for the first set, symbol number 0 is then assigned to the second set by default. The second plot will contain 4 sets of data points and one curve. The default sequence of point symbols is used except for the first set of points. Note that PLØ=F was used for all KEDAK data so that curves will be shown but not the nodes (the data points through which the curves are drawn). After the first occurrence of PLØ=F marking of points is switched on again by PLØ=T in the next program control input list. Extensive use is made of the default options: CALP=F is omitted; XMIN,XMAX,XLG,RAST are not specified for the second plot so the values of the first plot remain in effect; TØ,FRØM are specified only at the beginning and remain unchanged throughout the job.

4.1.5 Output

A number of output options were mentioned already. The following section complements and summarizes the various possibilities offered by SIGPLO.

4.1.5.1 Output_options_

The "current data set" is listed or not with

DRUCK = { T : print
        { F : do not print (default option),

it is punched (or written on tape or disc etc.) or not with

STANZ = { T : punch
        { F : do not punch

and it is plotted or not with

PLØ = { T : plot
      { F : do not plot.

DRUCK, STANZ and PLØ can occur in any one of the control input lists _&KEDAK..., _&NEUDAD..., _&KARTEN_....Standard formats are used for printing and punching. In the latter case the user can overrule the default option (2E14.6) by means of the special program control input

_&STAFOR='(format)',&END

The options DRUKU,STAKU and PLØKU were already introduced in Sect. 4.1.3.1 (Interpolation), the options DRUSM,STASM,PLOSM in Sect. 4.1.3.2 (Curve fitting, smoothing). These latter six options refer to processed data and in general require specification of a "new energy grid" for the output.

4.1.5.2 Specification_of_new_grid_for_output_

As explained before the options DRUKU,STAKU,DRUSM,STASM permit tabulation of curve points representing the interpolation or smoothing results on various output media. The new grid, i.e. the energy values for the tabulation, will in

general differ from the initial grid of the original data. The user must specify the new grid by the following program control input:

```
_&ØUTINT_FMTØUT='(format)',ZXINT=...,XINT=...,EINHTX=...,
_FRØM=...,TØ=...,ADD=...,additional information,&END
```

where FMTØUT is the format that is used to print curve points of the smooth curve requested by DRUKU or DRUSM.

ZXINT specifies the medium from which the new grid energies (array XINT) is to be retrieved. There are five possibilities:

ZXINT = n, with integer n, means that in the same program control input list the new grid energies are coded in the form XINT=E_1,E_2,...E_n, i.e. the array XINT is given as a list of n energy values (real numbers) separated by commas.

ZXINT='KEDAK' means that XINT is to be retrieved from a KEDAK-type library. This requires the following additional information in the &ØUTINT control input list (unless already specified by default):
PLEH=...,NAMZ=...,NAMEN=...,FRØM=...,TØ=...
(cf. Sect. 4.1.2.1) or an immediately following &KEDAK control input list containing XINTPØ=T.

ZXINT='NEUD' means XINT is to be retrieved from a NEUDADA "transmission" tape. This requires the following additional information in the control input list (unless already specified by default):
BDEH=...,ACCNR=...,LTR=...,SEQN=...,FRØM=...,TØ=...,
(cf. Sect. 4.1.2.2)

ZXINT='CARD' means XINT is to be read from cards. This requires the following additional information in the &ØUTINT control input list (unless already specified by default):
KARTEH=...,FMT=...,ANZ=...,FRØM=...,TØ=...
(cf. Sect. 4.1.2.3) or an immediately following &KEDAK control input list containing XINTPØ=T.

ZXINT='DATA' means that the energies of the initial ("current") data set are to be used as "new grid".

EINHTX = { 'EV' / 'KEV' / 'MEV' } specifies the unit for XINT

ADD = { T causes merging with an already existing XINT array, / F suppresses merging. }

If FORTRAN-readable output is to be produced its FORMAT can be specified by the program control input _&STAFOR_STAFMT='(format)',&END

4.1.6 Summary table

The options available in the various control input lists are shown schematically in Table 1 together with the default options. It should be added that the default option for _&STAFØR_STAFMT=...,&END is (2E14.6).

Table 1 SIGPLO Options

Option	optional in program control input list							default value	explanatory remarks
	&KEDAK	&NEUDAD	&KARTEN	&RELGEW	&SMØWDH	&ØUTINT	&PLØTGR		
Retrieval									
PLEH	+					+		1	FT number of input device
BDEH		+				+			
KARTEH		+	+			+			
NAMZ	+					+		2	number of KEDAK names
NAMEN	+					+			KEDAK names
LTR		+				+		'O'	letter preceding ACCNR
ACCNR		+				+			NEUDADA accession number
ANZ			+			+		O[+)]	number of data points read
FRØM	+	+	+			+		O.	minimum energy
TØ						+		1.E+20	maximum energy
EINHTX			+	+	+		+	'EV'	abscissa units
EINHTY			+				+	'BARN'	ordinate units
EINHTD			+					'BARN'	error units
XEINHT		+						'MEV'	abscissa units
YEINHT		+						'BARN'	ordinate units
DEINHT		+						'BARN'	error units
FMT		+	+			+		'(2E14.6)'	input format
SEQN		+						T	accession number in sequence?
REWD			+			+		F	rewinding?
SMGEW			+					F	weighting?
MEDIUM		+						'_____'	medium for uncertainties
ZAHL		+						0	number of uncertainties
ZAHLXR				+	+			0	number of relative weights
DATADD		+	+					F	merge with "current" set?
ADD						+		F	merge with current "new grid"?
ZXINT						+		0	number of "new grid" points
XINTPØ	+	+						F	are these "new grid" points?

[+)] reset for each program control input list

Table 1 (cont.)

Option	optional in program control input list							default value	explanatory remarks
	&KEDAK	&NEUDAD	&KARTEN	&RELGEW	&SMØWDH	&ØUTINT	&PLØTGR		
Data processing									
KURVE	+	+	+					'NEIN'	interpolation type
SMØØ		+	+					0.+)	maximum chi-squared
FQU					+			0.	new maximum chi-squared?
FISS	+	+	+					F	fission spectrum average?
Plots									
PLØ	+	+	+					T	plot of "current" set?
PLØKU	+	+	+					T	plot of interpolated curve?
PLØSM	+	+	+					T	plot of smooth curve?
PLØZEI	+	+	+					-1	which point symbol?
PLØFQU					+			T	chi-squared displayed?
NEUZEI					+			F	new plot?
XMIN							+	-1.E-60	minimum x-value
XMAX							+	-1.E-60	maximum x-value
XLG							+	-1.E-60	length of x-axis (cm)
YMIN							+	-1.E-60	minimum y-value
YMAX							+	-1.E-60	maximum y-value
YLG							+	-1.E-60	length of y-axis (cm)
RAST							+	'NEIN','____'	grid? log scale(s)?
CALP							+	F	CALCOMP plot?

+) reset for each program control input list

Table 1 (cont.)

Option	optional in program control input list &KEDAK	&NEUDAD	&KARTEN	&RELGEW	&SMØWDH	&ØUTINT	default value	explanatory remarks
Output								
DRUCK	+	+	+				F	"current" set on listing?
STANZ	+	+	+				F	"current" set on cards, tape,...?
DRUKU	+	+	+				F	interpolation results on listing?
STAKU	+	+	+				F	interpolation results on cards, tape,...?
DRUSM		+	+				F	smoothing results on listing?
STASM		+	+				F	smoothing results on cards, tape,...?
FMTØUT						+	'(2E14.6)'	printing format

4.1.7 References

/1/ R. Meyer, User's Guide to SIGPLO, unpublished (1975)

/2/ Present Compendium, Ch. III.2

/3/ Present Compendium, Ch. II.2.3.1.2

/4/ see CCDN Newsletters for more information on neutron data dissemination by CCDN

/5/ H. Späth, SPLINE, (1968), unpublished

/6/ H. Späth, SMØØTH, (1968),unpublished

/7/ S. Heine, PLØTA, Programmbeschreibung (1967) and updates, unpublished

Appendix: SIGPLO Program List

```
C----------SIGPLO.                                                          10
C----------DATE: 7.4.1973.                                                 20
C----------PROGRAM TO RETRIEVE,EDIT,PLOT,PRINT,INTERPOLATE,               30
C----------AND FIT DATASETS THAT ARE CONSIDERED AS EXPERIMENTAL            40
C----------OR EVALUATED. DATA SOURCES MAY BE ANY FORMATTED INPUT,          50
C----------KEDAK LIBRARY FORMAT DATA SETS OR DATA SETS WRITTEN IN          60
C----------NEUDADA TRANSMISSION FORMAT.                                    70
      REAL*8 NAMEN(3)                                                      80
      REAL NEIN,LINE,KURVE,MEDIUM,LTR,MOD                                  90
      INTEGER ACCNR,BDEH                                                  100
      INTEGER ANZ,PLEH,ZAHLXR,ZXINT,WNR,ZAHL,PLOZEI                       110
      LOGICAL DRUCK,STANZ,PLO,PLOKU,DRUKU,STAKU,PLOSM,DRUSM,STASM,SMGEW,  120
     1NEUZEI,XINTPO,PLOFQU,FISS,SEQN,ADD,CALP,CALPJ,DATADD,TST,REWD,KDK   130
     2,TSTS                                                               140
      DIMENSION NTEXT(15),XPH(2),YPH(2)                                   150
      DIMENSION RTITEL(20),STEUER(20),NNP(40),TA(2,7),TB(2)               160
      DIMENSION XP(1000),YP(1000),T(1000),S(1000),S2(1000)                170
      DIMENSION Z1(1002),Z2(1002),Z3(1002),Z4(1002),Z5(1002),Z6(1002),Z7  180
     1(1002)                                                              190
      DIMENSION A(1000),B(1000),C(1000),D(1000)                           200
      DIMENSION XS(1000),YS(1000)                                         210
      DIMENSION XINT(1000)                                                220
      DIMENSION XR(20),PR(20)                                             230
      DIMENSION P(1000),DELTA(1000),PS(1000)                              240
      DIMENSION XL(1000),YL(1000),XSK(200),YSK(200)                       250
      REAL*8 BSK(200)                                                     260
      EQUIVALENCE (FROM,VON,EUNT,E1,EMIN),(TO,BIS,EOB,E2,EMAX)            270
      EQUIVALENCE (S2(1),Z1(1),YL(1)),(XP(1),Z3(1)),(YP(1),Z4(1)),(DELTA  280
     1(1),Z5(1)),(T(1),Z6(1)),(S(1),Z7(1))                                290
      EQUIVALENCE (Z2(1),XL(1),XSK(1)),(XL(201),YSK(1)),(XL(401),BSK(1))  300
      COMMON XP,YP,ANZ                                                    310
      COMMON/PAR/XMAX,SX,YMAX,SY,XMIN,YMIN,XMAL,XMIL,YMAL,YMIL,NLGGX,     320
     1 NLGGY,NX,NY,INDZ,JNDZ,XMILJ,YMILJ,JP                               330
      COMMON/CLP/CALP                                                     340
      COMMON/PARM1/ MAXOPT,PLEH,PLOZEI,ZXINT,IDPLOT,CALPJ                 350
      COMMON/PARM2/ IPLOGR,ISTAFO,ISMO,IFMOUT,ISMNP,DISP,EINHTX,EINHTY,   360
     1 EINHTD,XEINHT,YEINHT,DEINHT,LTR,KARTEH,FMTOUT(20),STAFMT(20),      370
     2 FMT(20),DRUCK,STANZ,PLO,KURVE,PLOKU,DRUKU,STAKU,PLOSM,DRUSM,       380
     3 STASM,SMGEW,NEUZEI,XINTPO,PLOFQU,FISS,SEQN,ADD,DATADD,TST,         390
     4 REWD,NANZ,NADD,ZAHL,MEDIUM,WNR,                                    400
     5 RAST(2),ISMAX,FQU,NAMZ,ZAHLXR,TSTS                                410
      COMMON/PARM3/ SMOO                                                  420
      COMMON/PARM4/FROM,TO                                                430
      COMMON/INOUT/KOUT,KIN,KPU,KINC                                      440
      NAMELIST/KARTEN/KARTEH,FMT,ANZ,DRUCK,STANZ,PLO,KURVE,PLOKU,DRUKU,S  450
     1TAKU,SMOO,PLOSM,DRUSM,STASM,SMGEW,EINHTX,EINHTY,EINHTD,FISS,PLOZEI  460
     2,DATADD,TST,E1,E2,REWD,DISP,TSTS,FROM,VON,EUNT,EMIN,TO,BIS,EOB,     470
     3 EMAX                                                               480
      NAMELIST/KEDAK/PLEH,NAMZ,NAMEN,EUNT,EOB,DRUCK,STANZ,PLO,KURVE,PLOK  490
     1U,DRUKU,STAKU,XINTPO,FISS,PLOZEI,TST,FROM,VON,E1,EMIN,              500
     2 EMAX,TO,BIS,E2                                                     510
      NAMELIST/STAFOR/STAFMT                                              520
      NAMELIST/RELGEW/ZAHLXR,XR,PR,EINHTX                                 530
      NAMELIST/OUTINT/FMTOUT,ZXINT,XINT,EINHTX,ADD,PLEH,NAMEN,NAMZ,       540
     1BDEH,ACCNR,LTR,SEQN,VON,BIS,ANZ,FMT,KARTEH,REWD,                    550
```

```
     2 FROM,EUNT,E1,EMIN,TO,ECB,E2,EMAX                                 560
      NAMELIST/SMOWDH/FQU,ZAHLXR,XR,PR,NEUZEI,EINHTX,FLOFQU,TST,TSTS    570
      DATA NEIN/'NEIN'/,LINE/'LINE'/,SPLI/'SPLI'/,XLOG/'XLOG'/,YLOG/'YLO 580
     1G'/,MOD/'MOD'/                                                    590
      DATA TEND/'ENDE'/,NEUD/'NEUD'/,KEDA/'KEDA'/,KART/'CARD'/          600
      DATA NNP/9,0,3,4,5,6,7,1,8,2,0,1,2,7,8,0,1,2,7,8,0,1,2,7,8,0,1,2,7 610
     1,8,0,1,2,7,8,0,1,2,7,8/                                           620
      DATA RT/'RT  '/,PERC/'PERC'/                                      630
      DATA TA/' &KA','RTEN',' &NE','UDAD',' &KE','DAK ',' &PL','OTGR',' 640
     1&ST','AFOR',' &OU','TINT',' &RE','LGEW'/                          650
      DATA TB/' &SM','OWDH'/                                            660
      DATA NDAT/'DATA'/                                                 670
      DATA NTEXT/'FQU=',14*'    '/                                      680
C----------INITIALIZATION ONCE FOR WHOLE JOB.                          690
      CALL INITL                                                        700
C----------PRINT INPUT FROM KINC.                                      710
      CALL PRIEIN(KINC,KOUT,60)                                         720
C----------PUT CONTROL INPUT ONTO UNIT FT05.                           730
      CALL CDINPT(RTITEL,STEUER)                                        740
      WRITE(KOUT,638)                                                   750
C----------RT  CARD?                                                   760
 1000 READ(KIN,500,END=901,ERR=801) RTITEL                              770
      IF(RTITEL(1).EQ.TEND) GOTO 810                                    780
      IF(RTITEL(1).NE.RT) GOTO 802                                      790
C----------PRINT RT  CARD AND  PERFORM RT-INITIALIZATION.              800
      WRITE(KOUT,600) RTITEL                                            810
 1003 CALL INIT2                                                        820
      DO 1004 I=1,1000                                                  830
 1004 PS(I)=1.                                                          840
  100 ANZ=0                                                             850
      SMOO=NEIN                                                         860
      KDK=.FALSE.                                                       870
C----------TEST COMMAND (=NAMELIST NAME.)                              880
      READ(KIN,500,END=902,ERR=803) STEUER                              890
      DO 101 J=1,MAXOPT                                                 900
      CALL ASSIGN(STEUER,TA,J,NR,MAXOPT)                                910
      IF(NR.EQ.0) GOTO 101                                              920
      BACKSPACE KIN                                                     930
      GOTO (110,120,130,140,150,160,170),J                              940
  101 CONTINUE                                                          950
      GOTO 804                                                          960
C----------FORMATTED INPUT IS REQUESTED.                               970
  110 READ(KIN,KARTEN)                                                  980
      WRITE(KOUT,636)                                                   990
      WRITE(KOUT,612) KARTEH,FMT,ANZ,DATADD,DRUCK,STANZ,PLO,PLOZEI,    1000
     1KURVE,PLOKU,DRUKU,STAKU,DISP,EINHTX,EINHTY,EINHTD                 1010
      WRITE(KOUT,653) E1,E2,REWD                                        1020
      IF(SMOO.NE.NEIN) GOTO 111                                         1030
      WRITE(KOUT,613) SMOO,PLOSM,DRUSM,STASM,SMGEW,FISS                 1040
      GOTO 112                                                          1050
  111 WRITE(KOUT,614) SMOO,PLOSM,DRUSM,STASM,SMGEW,FISS                 1060
      ISMO=1                                                            1070
  112 CALL LESEN1(KARTEH,FMT,SMGEW,DELTA,E1,E2,REWD,&1C0)               1080
C----------CONVERT DATA TO INTERNAL PHYSICAL UNITS,  IF REQUIRED.      1090
      CALL ECONV(ANZ,XP,EINHTX)                                         1100
      CALL ECONV(ANZ,YP,EINHTY)                                         1110
      IF(.NOT.SMGEW) GOTO 128                                           1120
C----------CONVERT ERROR BARS  ALSO.                                   1130
      IF(EINHTD.EQ.PERC) GOTO 113                                       1140
      CALL ECONV(ANZ,DELTA,EINHTD)                                      1150
      GOTO 128                                                          1160
  113 CALL PERCNT(ANZ,YP,DELTA)                                         1170
      GOTO 128                                                          1180
C----------NEUDADA INPUT IS REQUESTED.                                 1190
  120 CALL NEUDA(XINT,DELTA,&100)                                       1200
C----------ADF APPENDS DATA TO CURRENT SET IF REQUESTED.               1210
  128 CALL ADF(XL,YL,Z6,NADD,NANZ,ANZ,XP,YP,DELTA,DATADD,&100)          1220
      GOTO 200                                                          1230
C----------KEDAK DATA ARE TO BE FETCHED.                               1240
  130 READ(KIN,KEDAK)                                                   1250
      KDK=.TRUE.                                                        1260
      WRITE(KOUT,636)                                                   1270
      WRITE(KOUT,619) PLEH,NAMZ,(NAMEN(I),I=1,2)                        1280
      IF(NAMZ.GE.3) WRITE(KOUT   ,656) NAMEN(3)                         1290
      WRITE(KOUT,657) EUNT,EOB,DRUCK,STANZ,PLO,PLOZEI,KURVE,PLOKU,DRUKU, 1300
     1 STAKU,DISP,XINTPO,FISS                                           1310
      CALL LDFOPN(PLEH,IDATUM,&903)                                     1320
  131 CALL KEDSUB(NR,NAMZ,NAMEN,EUNT,EOB,XINTPO,XINT,ZXINT,IFMCUT,&100) 1330
      IF(ANZ.NE.0) GOTO 200                                             1340
      IF(NR.NE.0) GOTO 132                                              1350
      WRITE(KOUT,620)                                                   1360
      GOTO 100                                                          1370
  132 WRITE(KOUT,621)                                                   1380
      GOTO 100                                                          1390
C----------PLOT SIZE  DEFINITION.                                      1400
C----------COMPLETE PREVIOUS  PLOT, IF IT WAS CALCOMP.                 1410
  145 XPH(1)=XMILJ                                                      1420
      XPH(2)=XPH(1)                                                     1430
      YPH(1)=YMILJ                                                      1440
      YPH(2)=YPH(1)                                                     1450
      CALL PLOTA(XPH,YPH,-2,2,9,1,1,1,10,XMAX,XMIN,SX,YMAX,YMIN,SY,1,   1460
     1IDPLOT,0,0,0)                                                     1470
      WRITE(KOUT,645)                                                   1480
      STOP                                                              1490
  140 CALL PREPLO                                                       1500
      GOTO 100                                                          1510
C----------FORMATTED OUTPUT IS TO BE PRODUCED AND OUTPUT               1520
C----------PARAMETERS MUST BE DEFINED.                                 1530
  150 READ(KIN,STAFOR)                                                  1540
      WRITE(KOUT,636)                                                   1550
      WRITE(KOUT,622) STAFMT                                            1560
      WRITE(KOUT,636)                                                   1570
      ISTAFO=1                                                          1580
      GOTO 100                                                          1590
C----------INTERPOLATION REQUESTED. SAVE CURRENT INTERPOLATION         1600
C----------POINTS.                                                     1610
  160 CALL STRING(Z3(1),XINT(1),4*ZXINT)                                1620
      NZXINT=ZXINT                                                      1630
      READ(KIN,OUTINT)                                                  1640
      WRITE(KOUT,636)                                                   1650
```

```
C----------BRANCH TO CORRECT READING ROUTINE FOR INTERPOL. POINTS.
      IF(ZXINT.EQ.NDAT) GOTO 161
      IF(ZXINT.EQ.KEDA) GOTO 167
      IF(ZXINT.EQ.NEUD) GOTO 169
      IF(ZXINT.EQ.KART) GOTO 1670
      CALL ECONV(ZXINT,XINT,EINHTX)
      WRITE(KOUT,623) FMTOUT,ZXINT,EINHTX,ADD,(XINT(I),I=1,ZXINT)
      GOTO 162
  161 WRITE(KOUT,641) FMTOUT,ZXINT,EINHTX,ADD
      GOTO 165
C----------APPEND DATA, IF REQUESTED.
  162 IF(.NOT.ADD) GOTO 165
      IF(IFMOUT.NE.1) GOTO 166
      WRITE(KOUT,642)
      IF(ZXINT+NZXINT.LE.1000) GOTO 163
      WRITE(KOUT,643)
      NZXINT=1000-ZXINT
  163 I=ZXINT+1
      DO 164 J=1,NZXINT
      XINT(I)=Z3(J)
  164 I=I+1
      ZXINT=ZXINT+NZXINT
  165 IFMOUT=1
      WRITE(KOUT,636)
      GOTO 100
  166 WRITE(KOUT,644)
      GOTO 165
C----------RETRIEVE INTERPOLATION ENERGIES FROM KEDAK.
  167 WRITE(KOUT,647) FMTOUT,ZXINT,PLEH,NAMZ,(NAMEN(I),I=1,NAMZ),VON,BIS
      CALL LDFOPN(PLEH,IDATUM,&903)
  168 CALL KEDEN(NAMZ,NAMEN,VON,BIS,XINT,ZXINT,&1697)
      WRITE(KOUT,652) ZXINT,(XINT(I),I=1,ZXINT)
      GOTO 162
C----------RETRIEVE INTERPOLATION ENERGIES FROM NEUDADA  TAPE.
  169 CALL NEUINT(XINT,BDEH,ACCNR,&100)
      GOTO 162
 1697 IFMOUT=0
      GOTO 100
 1670 WRITE(KOUT,654) ZXINT,KARTEH,ANZ,VON,BIS,RWD,FMT,FMTOUT
      CALL LESEN(KARTEH,FMT,REWD,ZXINT,ANZ,VON,BIS,XINT,&1697)
      CALL ECONV(ZXINT,XINT,EINHTX)
      WRITE(KOUT,651) EINHTX,ZXINT,(XINT(I),I=1,ZXINT)
      GOTO 162
C----------RELATIVE WEIGHTS FOR SMOOTH ARE REQUESTED.
  170 READ(KIN,RELGEW)
      CALL ECONV(ZAHLXR,XR,EINHTX)
      WRITE(KOUT,636)
      WRITE(KOUT,624) ZAHLXR,EINHTX,(XR(I),PR(I),I=1,ZAHLXR)
      WRITE(KOUT,636)
      GOTO 100
C-------------------------------------------------------------
C-----------------------------------------------------
C----------DO THE REQUESTED WORK.----------------S
C----------SORT DATA.
  200 CALL ORDNEN(ANZ,XP,YP,DELTA,Z1,Z2,Z6)
      DO 2001 I=1,ANZ
      XP(I)=Z1(I)
      YP(I)=Z2(I)
 2001 DELTA(I)=Z6(I)
      IF(.NOT.DRUCK) GOTO 201
C----------PRINT DATA POINTS.
      WRITE(KOUT,630)
      IF(KDK) GOTO 2004
      IF(SMGEW) GOTO 2002
 2004 CALL DRUCKX(XP,YP,ANZ)
      GOTO 201
 2002 CALL DRUCK3(ANZ,XP,YP,DELTA)
  201 IF(.NOT.STANZ) GOTO 203
C----------FORMATTED OUTPUT OF DATA POINTS.
      IF(ISTAFO.EQ.1) GOTO 202
      WRITE(KOUT,602)
  202 IF(DISP.EQ.MOD) GOTO 2020
      WRITE(KPU,500) RTITEL
 2020 IF(SMGEW) GOTO 2021
      WRITE(KPU,STAFMT) (XP(I),YP(I),I=1,ANZ)
      GOTO 203
 2021 WRITE(KPU,STAFMT) (XP(I),YP(I),DELTA(I),I=1,ANZ)
  203 IF(.NOT.PLO) GOTO 206
C----------PLOT DATA POINTS.
      IF(IPLOGR.EQ.1) GOTO 204
      WRITE(KOUT,603)
      GOTO 206
  204 IZ=0
      IF(ANZ.NE.1) GOTO 205
      IZ=1
      CALL XCORR
C----------A SINGLE DATA POINTS CANNOT BE HANDLED BY  PLOTA.
  205 JP=JP+1
      NP=NNP(JP)
      IF(PLOZEI.NE.-1) NP=PLOZEI
C----------FOR PLOTS WITH ONE AXIS LOGARITHMIC, USE PLOGXY.
      IF(RAST(1).EQ.XLOG.OR.RAST(2).EQ.YLOG) GOTO 2003
      CALL PLOTA(XP,YP,ANZ,1,NP,1,1,1,INDZ,XMAX,XMIN,SX,YMAX,YMIN,SY,1,I
     1DPLOT,0,0,0)
      WRITE(KOUT,604) NP
      CALL SUB1(INDZ,RAST,NEIN,XLOG,YLOG,KREX,KREY,&211)
C----------PLOT SCALE ALSO, IF REQUESTED.
      CALL RASTER(KREX,KREY,XMAX,XMIN,SX,YMAX,YMIN,SY,INDZ,XSK,YSK)
      GOTO 211
 2003 CALL PLOGXY(XP,YP,ANZ,1,NP,1,INDZ,1,
     1 XL,YL,RAST,XMAL,XMIL,XMAX,XMIN,SX,YMAL,YMIL,YMAX,YMIN,SY,
     2 IDPLOT,NLGGX,NLGGY,NX,NY,XSK,YSK,BSK)
      WRITE(KOUT,604) NP
      CALL SUB1(INDZ,RAST,NEIN,XLOG,YLOG,KREX,KREY,&211)
      CALL RASTER(KREX,KREY,XMAL,XMIL,SX,YMAL,YMIL,SY,INDZ,XSK,YSK)
      GOTO 211
  211 IF(IZ.NE.1) GOTO 206
      ANZ=1
C----------IS INTERPOLATION  OF DATA POINTS TO PRODUCE A
C----------CURVE REQUESTED?
```

```
C----------TWO MODES OF INTERPOLATION ARE SUPPLIED:
C----------LINEAR AND SPLINE.
  206 IF(KURVE.EQ.NEIN) GOTO 220
      IF(KURVE.EQ.LINE) IKU=1
      IF(KURVE.EQ.SPLI) IKU=2
      IF(.NOT.PLOKU) GOTO 208
      IF(IPLOGR.EQ.1) GOTO 207
      WRITE(KOUT,603)
      GOTO 208
  207 CALL KURVE1(ANZ,XP,YP,M,T,S,S2,A,B,C,D,IKU,&243)
      IF(.NOT.TST) GOTO 241
      WRITE(KOUT,650) (T(I),S(I),I=1,M)
  241 IF(.NOT.FISS) GOTO 240
C----------FISSION SPECTRUM AVERAGE.
      CALL QFISS(M,T,S,A)
C----------PLOT CURVE  WITH REQUESTED INTERPOLATION.
  240 IF(RAST(1).EQ.XLOG.OR.RAST(2).EQ.YLOG) GOTO 2041
      CALL PLOTA(T,S,M,2,0,1,1,1,INDZ,XMAX,XMIN,SX,YMAX,YMIN,SY,1,IDPLOT
     1,0,0,0)
      CALL SUB1(INDZ,RAST,NEIN,XLOG,YLOG,KREX,KREY,&243)
      CALL RASTER(KREX,KREY,XMAX,XMIN,SX,YMAX,YMIN,SY,INDZ,XSK,YSK)
      GOTO 243
 2041 CALL PLOGXY(T,S,M,2,0,1,INDZ,1,
     1 XL,YL,RAST,XMAL,XMIL,XMAX,XMIN,SX,YMAL,YMIL,YMAX,YMIN,SY,
     2 IDPLOT,NLGGX,NLGGY,NX,NY,XSK,YSK,BSK)
      CALL SUB1(INDZ,RAST,NEIN,XLOG,YLOG,KREX,KREY,&243)
      CALL RASTER(KREX,KREY,XMAL,XMIL,SX,YMAL,YMIL,SY,INDZ,XSK,YSK)
      GOTO 243
  208 IF(.NOT.FISS) GOTO 243
      CALL KURVE1(ANZ,XP,YP,M,T,S,S2,A,B,C,D,IKU,&243)
      CALL QFISS(M,T,S,A)
C----------PRINT CURVE ACCORDING TO USER SPECIFICATIONS.
  243 IF(.NOT.DRUKU) GOTO 209
      IF(IFMOUT.NE.0) GOTO 212
      WRITE(KOUT,627)
      GOTO 220
  212 IF(ZXINT.NE.NDAT) GOTO 216
      ZXINT=ANZ
      DO 217 I=1,ZXINT
  217 XINT(I)=XP(I)
  216 CALL ORDNEN(ZXINT,XINT,Z2,Z2,Z1,Z2,Z2)
      DO 213 I=1,ZXINT
  213 XINT(I)=Z1(I)
      CALL KURVE4(ANZ,XP,YP,ZXINT,XINT,S,S2,A,B,C,D,NA1,NA2,IKU,&209)
      WRITE(KOUT,628)
      WRITE(KOUT,FMTOUT) (XINT(I),S(I),I=NA1,NA2)
C----------FORMATTED OUTPUT OF CURVE ACCORDING USER SPECIFICATIONS.
  209 IF(.NOT.STAKU) GOTO 220
      IF(ISTAFO.EQ.1) GOTO 210
      WRITE(KOUT,602)
  210 IF(IFMOUT.NE.0) GOTO 214
      WRITE(KOUT,627)
      GOTO 220
  214 CALL ORDNEN(ZXINT,XINT,Z2,Z2,Z1,Z2,Z2)
      DO 215 I=1,ZXINT
  215 XINT(I)=Z1(I)
      CALL KURVE4(ANZ,XP,YP,ZXINT,XINT,S,S2,A,B,C,D,NA1,NA2,IKU,&220)
      IF(DISP.EQ.MOD) GOTO 2150
      WRITE(KPU,500) RTITEL
 2150 WRITE(KPU,STAFMT) (XINT(I),S(I),I=NA1,NA2)
      GOTO 220
C----------SAVE DATA FOR SMOOTH  FIT IF REQUESTED.
  220 IF(SMOO.EQ.NEIN) GOTO 230
      IF((ANZ+ISMAX).LE.1000) GOTO 224
      ANZ=1000-ISMAX
      WRITE(KOUT,658) ANZ
  658 FORMAT(//5X,'FEHLERNACHRICHT'/8X,'ZAHL DER PUNKTE FUER SMOOTH-FIT'
     *,' WUERDE GROESSER ALS 1000. ES WERDEN VON DEN SOEBEN GELESENEN ',
     *'DATEN NUR DIE ERSTEN',I5,' PUNKTE GENOMMEN.')
      IF(ANZ.LE.0) GOTO 230
  224 DO 221 I=1,ANZ
      XS(I+ISMAX)=XP(I)
  221 YS(I+ISMAX)=YP(I)
      FQU=FQU+SMOO
      IF(.NOT.SMGEW) GOTO 223
      DO 222 I=1,ANZ
  222 PS(I+ISMAX)=DELTA(I)
  223 ISMAX=ISMAX+ANZ
      GOTO 230
  230 GOTO 100
C-XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C-XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C----------SMOTH FIT SECTION.
  300 IF(ISMAX.LT.3) GOTO 809
C----------SORT DATA.
      CALL ORDNEN(ISMAX,XS,YS,PS,Z1,Z2,Z3)
      DO 301 I=1,ISMAX
      XS(I)=Z1(I)
      YS(I)=Z2(I)
  301 PS(I)=Z3(I)
      CALL EQUEN(ISMAX,XS,YS,PS)
      IF(ZAHLXR.EQ.0) GOTO 310
C----------SORT WEIGHTS.
  309 CALL ORDNEN(ZAHLXR,XR,PR,Z3,Z1,Z2,Z3)
C----------SUPPLY RELATIVE WEIGHTING.
      DO 302 I=1,ZAHLXR
      XR(I)=Z1(I)
  302 PR(I)=Z2(I)
      ASSIGN 305 TO NST
      K=1
      DO 327 I=1,ISMAX
  303 IF(XS(I).LT.XR(K)) GOTO NST,(305,304)
      IF(K.EQ.ZAHLXR) GOTO 306
      K=K+1
      ASSIGN 304 TO NST
      GOTO 303
  306 IANF=I
      GOTO 307
  304 P(I)=PS(I)*PR(K-1)
      GOTO 327
```

```
  305 P(I)=PS(I)
  327 CONTINUE
      GOTO 310
  307 DO 308 I=IANF,ISMAX
  308 P(I)=PS(I)*PR(ZAHLXR)
      GOTO 3102
  310 DO 3100 I=1,ISMAX
 3100 P(I)=PS(I)
C----------PRINT IDETIFICATION AND PROVIDE PRINTOUT OF DATA.
 3102 WRITE(KOUT,631) FQU
      CALL DRUCK3(ISMAX,XS,YS,P)
C----------SMOOTH. USE PROGRAM OF H.SPAETH. (SEE INR-REPORT)
      CALL SMOOTH(ISMAX,XS,YS,P,FQU,A,B,C,D,Z1,Z2,Z3,Z4,Z5,Z6,Z7)
C----------PLOT SMOOTHED CURVE, IF REQUESTED.
      IF(.NOT.PLOSM) GOTO 313
      IF(IPLOGR.EQ.1) GOTO 312
      WRITE(KOUT,603)
      GOTO 313
  312 CALL KURVE2(ISMAX,XS,YS,M,T,S,A,B,C,D,&343)
C----------PROVIDE TESTPRINTOUT OF SMOOTHED CURVE, IF REQUESTED.
      IF(TSTS) CALL TSTPRO(ISMAX,XS,YS,P,A,FQU)
      IF(.NOT.TST) GOTO 3124
      WRITE(KOUT,655)
      CALL DRUCKX(T,S,M)
 3124 IF(.NOT.FISS) GOTO 340
C----------FISSION SPECTRUM AVERAGE.
      CALL QFISS(M,T,S,Z1)
C---------PLOT INTO NEW OR OLD PLOT?
C----------OLD PLOT.
  340 IF(NEUZEI) GOTO 320
C----------INCREMENT PLOT IDENTIFIER.
      ISMNP=ISMNP+1
      NP=NNP(ISMNP)
      IF(ISMNP.EQ.1) GOTO 321
      NPA=M-1
      IF(RAST(1).EQ.XLOG.OR.RAST(2).EQ.YLOG) GOTO 304C
      CALL PLOTA(T,S,M,3,NP,1,1,NPA,INDZ,XMAX,XMIN,SX,YMAX,YMIN,SY,1,
     1IDPLOT,0,0,0)
      WRITE(KOUT,633) NP
  322 CALL SUB1(INDZ,RAST,NEIN,XLOG,YLOG,KREX,KREY,&343)
  319 CALL RASTER(KREX,KREY,XMAX,XMIN,SX,YMAX,YMIN,SY,INDZ,XSK,YSK)
      GOTO 343
 3040 CALL PLOGXY(T,S,M,3,NP,NPA,INDZ,1,
     1 XL,YL,RAST,XMAL,XMIL,XMAX,XMIN,SX,YMAL,YMIL,YMAX,YMIN,SY,
     2 IDPLOT,NLGGX,NLGGY,NX,NY,XSK,YSK,BSK)
      WRITE(KOUT,633) NP
 3220 CALL SUB1(INDZ,RAST,NEIN,XLOG,YLOG,KREX,KREY,&343)
 3190 CALL RASTER(KREX,KREY,XMAL,XMIL,SX,YMAL,YMIL,SY,INDZ,XSK,YSK)
      GOTO 343
  313 IF(.NOT.FISS) GOTO 343
      CALL KURVE2(ISMAX,XS,YS,M,T,S,A,B,C,D,&343)
      IF(.NOT.TST) GOTO 3134
      WRITE(KOUT,655)
      CALL DRUCKX(T,S,M)
 3134 CALL QFISS(M,T,S,Z1)
  343 IF(.NOT.DRUSM) GOTO 316
C----------PRINT SMOOTHED CURVE ACCORDING USER SPECIFICATIONS.
      IF(IFMOUT.NE.0) GOTO 314
      WRITE(KOUT,610)
      GOTO 330
  314 IF(ZXINT.NE.NDAT) GOTO 324
      ZXINT=ISMAX
      DO 323 I=1,ZXINT
  323 XINT(I)=XS(I)
  324 CALL ORDNEN(ZXINT,XINT,Z2,Z2,Z1,Z2,Z2)
      DO 315 I=1,ZXINT
  315 XINT(I)=Z1(I)
      CALL KURVE3(ISMAX,XS,YS,ZXINT,XINT,S,A,B,C,D,NA1,NA2,&330)
      WRITE(KOUT,629)
      WRITE(KOUT,FMTOUT) (XINT(I),S(I),I=NA1,NA2)
  316 IF(.NOT.STASM) GOTO 330
C----------PROVIDE FORMATTED OUTPUT, IF REQUESTED.
      IF(ISTAFO.EQ.1) GOTO 311
      WRITE(KOUT,602)
  311 IF(IFMOUT.NE.0) GOTO 317
      WRITE(KOUT,610)
      GOTO 330
  317 CALL ORDNEN(ZXINT,XINT,Z2,Z2,Z1,Z2,Z2)
      DO 318 I=1,ZXINT
  318 XINT(I)=Z1(I)
      CALL KURVE3(ISMAX,XS,YS,ZXINT,XINT,S,A,B,C,D,NA1,NA2,&330)
      IF(DISP.EQ.MOD) GOTO 3180
      WRITE(KPU,500) RTITEL
 3180 WRITE(KPU,STAFMT) (XINT(I),S(I),I=NA1,NA2)
      GOTO 330
C----------NEW PLOT WAS REQUESTED FOR SMOOTH.
  320 IDPLOT=IDPLOT+1
      IF(PLOFQU) GOTO 325
      IF(RAST(1).EQ.XLOG.OR.RAST(2).EQ.YLOG) GOTO 3200
      CALL PLOTA(XS,YS,ISMAX,1,4,1,1,1,JNDZ,XMAX,XMIN,SX,YMAX,YMIN,SY,1,
     1IDPLOT,0,0,0)
      GOTO 326
 3200 CALL PLOGXY(XS,YS,ISMAX,1,4,1,JNDZ,1,
     1 XL,YL,RAST,XMAL,XMIL,XMAX,XMIN,SX,YMAL,YMIL,YMAX,YMIN,SY,
     2 IDPLOT,NLGGX,NLGGY,NX,NY,XSK,YSK,BSK)
      GOTO 3260
  325 CALL CONVX(FQU,NTEXT(2),5HE12.4)
      IF(RAST(1).EQ.XLOG.OR.RAST(2).EQ.YLOG) GOTO 3250
      CALL PLOTA(XS,YS,ISMAX,1,4,1,2,1,JNDZ,XMAX,XMIN,SX,YMAX,YMIN,SY,
     1NTEXT,IDPLOT,0,0,0)
      GOTO 326
 3250 CALL PLOGXY(XS,YS,ISMAX,1,4,1,JNDZ,NTEXT,
     1 XL,YL,RAST,XMAL,XMIL,XMAX,XMIN,SX,YMAL,YMIL,YMAX,YMIN,SY,
     2 IDPLOT,NLGGX,NLGGY,NX,NY,XSK,YSK,BSK)
      GOTO 3260
  326 JJ=0
      IF(JNDZ.GE.10) JJ=10
      CALL PLOTA(T,S,M,2,0,1,1,1,JJ,XMAX,XMIN,SX,YMAX,YMIN,SY,1,IDPLOT,
     10,0,0)
      J=1
```

```
      CALL SUB1(J,RAST,NEIN,XLOG,YLOG,KREX,KREY,&313)
      WRITE(KOUT,634)
      GOTO 319
 3260 JJ=0
      IF(JNDZ.GE.10) JJ=10
      CALL PLOGXY(T,S,M,2,0,1,JJ,1,
     1 XL,YL,RAST,XMAL,XMIL,XMAX,XMIN,SX,YMAL,YMIL,YMAX,YMIN,SY,
     2 IDPLOT,NLGGX,NLGGY,NX,NY,XSK,YSK,BSK)
      J=1
      CALL SUB1(J,RAST,NEIN,XLOG,YLOG,KREX,KREY,&313)
      GOTO 3190
  321 IF(RAST(1).EQ.XLOG.OR.RAST(2).EQ.YLOG) GOTO 3210
      CALL PLOTA(T,S,M,2,NP,1,1,1,INDZ,XMAX,XMIN,SX,YMAX,YMIN,SY,1,
     1ICPLOT,0,0,0)
      GOTO 322
 3210 CALL PLOGXY(T,S,M,2,NP,1,INDZ,1,
     1 XL,YL,RAST,XMAL,XMIL,XMAX,XMIN,SX,YMAL,YMIL,YMAX,YMIN,SY,
     2 IDPLOT,NLGGX,NLGGY,NX,NY,XSK,YSK,BSK)
      GOTO 3220
C----------END OF SMOOTH. IS ANOTHER ONE REQUIRED?
  330 ISMO=0
      ZAHLXR=0
      GOTO 100
C*****************************************************
C*****************************************************
C  ********** FEHLERROUTINEN **********
  801 WRITE(KOUT,605)
      GOTO 100
  802 WRITE(KOUT,606) RTITEL
      BACKSPACE KIN
      GOTO 1003
  803 WRITE(KOUT,607)
      GOTO 100
  804 IF(ISMO.NE.1) GOTO 805
      BACKSPACE KIN
      WRITE(KOUT,636)
      GOTO 300
  805 IF(STEUER(1).NE.TB(1)) GOTO 807
      IF(STEUER(2).NE.TB(2)) GOTO 807
      BACKSPACE KIN
      ZAHLXR=0
      READ(KIN,SMOWDH)
      WRITE(KOUT,636)
      WRITE(KOUT,639) ZAHLXR,NEUZEI,PLOFQU
      IF(ISMAX.LT.3) GOTO 809
      IF(ZAHLXR.EQ.0) GOTO 310
      WRITE(KOUT,640) EINHTX
      CALL ECONV(ZAHLXR,XR,EINHTX)
      WRITE(KOUT,626) (XR(I),PR(I),I=1,ZAHLXR)
      GOTO 309
  807 IF(STEUER(1).EQ.TEND) GOTO 810
      IF(STEUER(1).NE.RT) GOTO 808
      BACKSPACE KIN
      WRITE(KOUT,638)
      GOTO 1000
  808 WRITE(KOUT,601) STEUER
      GOTO 100
  809 WRITE(KOUT,632)
      ISMO=0
      GOTO 100
  810 WRITE(KOUT,637)
      IF(CALPJ) GOTO 145
      STOP
C  ********** ENDROUTINEN **********
  901 WRITE(KOUT,611)
      STOP
  902 WRITE(KOUT,611)
      IF(ISMO.EQ.1) GOTO 300
      STOP
  903 WRITE(KOUT,608)
      GOTO 100
C  ********** FORMATE **********
  500 FORMAT(20A4)
  600 FORMAT(1H0,7X,84('*')/8X,'*',82X,'*'/8X,'*',1X,20A4,1X,'*'/8X,'*',
     182X,'*'/8X,84('*')///)
  601 FORMAT(//5X,'FEHLERNACHRICHT'/8X,'EINGELESENE KARTE:'/8X,'***',20A
     14,'***'/8X,'WURDE NICHT ALS STEUERKARTE ERKANNT,OBWOHL SOLCHE ERWA
     2RTET WURDE'/5X,'AKTION:UEBERLESEN BIS ERKENNEN EINER STEUER-,ENDE-
     3ODER RT-KARTE'//)
  602 FORMAT(//5X,'NACHRICHT:'/8X,'STANZEN WURDE GEWUENSCHT.DA STAFORKAR
     1TE FEHLT,WIRD (2E12.4) ALS AUSGABEFORMAT ANGENOMMEN')
  603 FORMAT(//5X,'FEHLERNACHRICHT:'/8X,'PLOT WURDE GEWUENSCHT.PLOTGR-KA
     1RTE FEHLT'//)
  604 FORMAT('  PLOTNACHRICHT: NP =',I3)
  605 FORMAT(//5X,'FEHLERNACHRICHT'/8X,'LESEFEHLER BEIM VERSUCH,TITEL-KA
     1RTE ZU LESEN'/5X,'AKTION:KARTE WIRD UEBERGANGEN.ES WIRD VERSUCHT,N
     2AECHSTE KARTE ALS STEUERKARTE ZU LESEN'//)
  606 FORMAT(//5X,'FEHLERNACHRICHT'/8X,'ALS RT-KARTE EINGELESENES RECORD
     1 TRUG KEINE RT-KENNUNG'/8X,'RECORDTEXT:*****',20A4,'*****'/5X,'AKT
     2ION:RECORD WIRD UEBERLESEN.NAECHSTES RECORD WIRD ALS STEUERKARTE E
     3INGELESEN'//)
  607 FORMAT(//5X,'FEHLERNACHRICHT'/8X,'LESEFEHLER BEIM LESEN EINER STEU
     1ERKARTE'/5X,'AKTION:KARTE WIRD UEBERLESEN.ES WIRD VERSUCHT NAECHST
     2E KARTE ALS STEUERKARTE ZU LESEN'//)
  608 FORMAT(//5X,'FEHLERNACHRICHT'/8X,'LDFOPN-AUFRUF BLIEB ERFOLGLOS'/5
     1X,'AKTION:KEDAK-OPERATIONEN UNTERBLEIBEN.NAECHSTE STEUERKARTE WIRD
     2 GESUCHT'//)
  610 FORMAT(//5X,'FEHLERNACHRICHT'/8X,'AUSGABE VON SMOOTH ANDERS ALS DU
     1RCH PLOT WURDE GEFORDERT.ES WURDE JEDOCH KEIN &CUTINT /XINTPO GEFU
     2NDEN'/5X,'AKTION:DIESE AUSGABE ERFOLGT NICHT'//)
  611 FORMAT(//5X,'FEHLERNACHRICHT'/8X,'BEIM VERSUCH,EINE KARTE EINZULES
     1EN WURDE DAS ENDE DER KARTENEINGABE ERREICHT'/5X,'AKTION:GEWUENSCH
     2TE UND NOCH NICHT AUSGEFUEHRTE SMOOTHOPERATIONEN WERDEN DURCHGEFUE
     3HRT'//)
  612 FORMAT(///' &KARTEN KARTEH=',I3,', FMT=',20A4/
     15X,'ANZ=',I5,',  DATADD=',L2,',  CRUCK=',L2,',  STANZ=',L2,',  PL
     XO=',L2,',  PLOZEI=',I2/
     25X,'KURVE=',A4,',  PLOKU=',L2,',  CRUKU=',L2,',  STAKU=',L2,',  DI
     XSP=',A4/
     35X,'EINHTX=',A4,',  EINHTY=',A4,',  EINHTO=',A4)
```

```
613 FORMAT(5X,'SMOO=',A4,', PLOSM=',L2,', DRUSM=',L2,', STASM=',L2,',
   1SMGEW=',L2,', FISS=',L2)
614 FORMAT(5X,'SMOO=',E12.3,', PLOSM=',L2,', DRUSM=',L2,', STASM=',L2,
   1', SMGEW=',L2,', FISS=',L2)
619 FORMAT(///' &KEDAK PLEH=',I3,'  NAMZ=',I3,'  NAMEN=',A8,2X,A8)
620 FORMAT(//5X,'FEHLERNACHRICHT'/8X,'ZAHL DER KEDAKENERGIEN ZWISCHEN
   1EU UND EO = 0'/5X,'AKTION:KEIN PLOT VON KEDAKWERTEN'//)
621 FORMAT(//5X,'FEHLERNACHRICHT'/8X,'ZAHL DER IM KEDAKAUFRUF ERHALTEN
   1EN WERTE = 0'/5X,'AKTION: KEIN PLOT VON KEDAKWERTEN'//)
622 FORMAT(/' &STAFOR STAFMT=',20A4)
623 FORMAT(/' &OUTINT FMTOUT=',20A4/5X,'ZXINT=',I5,', EINHTX=',A4,', A
   1DD=',L2,', XINT='/(10(1X,E12.5)))
624 FORMAT(/' &RELGEW ZAHLXR=',I3,', EINHTX=',A4,', XR,PR='/(10(1X,E12
   1.5)))
625 FORMAT(/' &SMOWDH FQU=',E13.5,', ZAHLXR=',I3,', EINHTX=',A4,', NEU
   1ZEI=',L2)
626 FORMAT(5X,'XR,PR='/(10(1X,E12.5)))
627 FORMAT(//5X,'FEHLERNACHRICHT'/8X,'AUSGABE DER  KURVE ANDERS ALS DU
   1RCH PLOT WURDE GEFORDERT.ES WURDE JEDOCH KEIN &OUTINT /XINTPO GEFU
   2NDEN'/5X,'AKTION:DIESE AUSGABE ERFOLGT NICHT'//)
628 FORMAT(5X,'AUSDRUCK DER AUF XINT INTERPOLIERTEN KURVE')
629 FORMAT(5X,'AUSDRUCK DES AUF XINT INTERPOLIERTEN SMOOTH')
630 FORMAT(5X,'EINGABEDATEN X/Y  BZW  X/Y/P :')
631 FORMAT(5X,'SMOOTH WIRD MIT FOLGENDEN WERTEN ANGELAUFEN:' /8X,'FQU=
   1',E12.5/8X,'X/Y/P ='/)
632 FORMAT(//5X,'NACHRICHT'/8X,'DA DIE ANZAHL DER PUNKTE FUER SMOOTH <
   13 IST,WIRD SMOOTH NICHT AUSGEFUEHRT'//)
633 FORMAT(5X,'NACHRICHT'/8X,'SMOOTH WIRD MIT DEM ZI NP=',I2,' GEHOERE
   1NDEM ZEICHEN GEPLOTTET.DIESE ZEICHEN BEFINDEN SICH AM ANFANG UND A
   2M ENDE DER KURVE ')
634 FORMAT(5X,'NACHRICHT'/8X,'SMOOTH WIRD AUF NEUES BLATT GEPLOTTET.DI
   1E DEM FIT ZUGRUNDE LIEGENDEN PUNKTE WERDEN  MIT NP=4 GEZEICHNET')
636 FORMAT('----------------------------------------------------------
   1----------------------------------------------------------------
   2-------')
637 FORMAT(5X,'NACHRICHT'/8X,'*ENDE*-RECORD WURDE ERREICHT.PROGRAMMAUS
   1FUEHRUNG WIRD BEENDET')
638 FORMAT(1H1)
639 FORMAT(///' &SMOWDH ZAHLXR=',I3,', NEUZEI=',L2,', PLOFQU=',L2)
640 FORMAT(5X,'EINHTX=',A4)
641 FORMAT(/' &OUTINT FMTOUT=',20A4/5X,'ZXINT=',A4,',  EINHTX=',A4,
   1',   ADD=',L2)
642 FORMAT(8X,'DIESE PUNKTE WERDEN ZU DEN SCHON BESTEHENDEN INTERPOLAT
   1IONSPUNKTEN HINZUGEFUEGT')
643 FORMAT(8X,'MIT DEN NEU HINZUGEKOMMENEN PUNKTEN WAERE DEREN ZAHL >
   11000.DIE P(N) MIT N>1000 WERDEN WEGGELASSEN')
644 FORMAT(//5X,'FEHLERNACHRICHT'/8X,'ADD WURDE IN &OUTINT SPEZIFIZIER
   1T.ES LIEGEN BIS JETZT JEDOCH NOCH KEINE INTERPOLATIONSPUNKTE VOR'/
   28X,'AKTION:DIE PUNKTE WERDEN OHNE AENDERUNG UEBERNOMMEN')
645 FORMAT(5X,'CALCOMPPLOT WURDE ABGESCHLOSSEN')
647 FORMAT(/' &OUTINT FMTOUT=',20A4/
   1 5X,'ZXINT=',A4,',  PLEH=',I2,',  NAMZ=',I2,',  NAMEN=',2A8/
   15X,'VON=',E14.6,'EV',5X,'BIS=',E14.6,'EV'/)
650 FORMAT(//'  KONTROLLAUSDRUCK DER GEPLOTTETEN KURVE:'//
   1(10E13.6))
651 FORMAT(//5X,'EINHTX=',A4,'  ZXINT=',I5,'   XINT='/(10E13.6))
652 FORMAT(//5X,'ZXINT=',I5,'   XINT='/(10E13.6))
653 FORMAT(8X,'  FROM=',E13.6,',  TO=',E13.6,'  IN UNITS OF EINHTX.',3
   XX,'REWD=',L2)
654 FORMAT(//5X,'&OUTINT ZXINT=',A4,'   KARTEH=',I2,'   ANZ=',I6,'   F
   XROM=',E13.6,'   TO=',E13.6,'   REWD=',L2/
   1 8X,'FMT=',20A4/
   1 8X,'FMTOUT=',20A4/)
655 FORMAT(///'       TESTAUSDRUCK DER SMOOTHKURVE FUER PLOT ODER SIGMA(
   1FISS).'//)
656 FORMAT('+',54X,E14.6)
657 FORMAT(8X,'FROM=',E14.6,',   TO=',E14.6/
   15X,'DRUCK=',L2,', STANZ=',L2,', PLC=',L2,',  PLCZEI=',I2/
   25X,'KURVE=',A4,',  PLOKU=',L2,',  DRUKU=',L2,',  STAKU=',L2,',  DI
   XSP=',A4/
   35X,'XINTPO=',L2,',  FISS=',L2)
    END
```

```
      SUBROUTINE NEUDA(XINT,DELTA,*)
C----------READ DATA POINTS FROM NEUDADA TRANSMISSION FILE.
      INTEGER PLEH,ZXINT,WNR,ZAHL,PLOZEI
      COMMON/INOUT/KOUT,KIN
      INTEGER ACCNRV,BDEH,ACCNR,ANZ
      LOGICAL DRUCK,STANZ,PLC,PLOKU,DRUKU,STAKU,PLOSM,DRUSM,STASM,SMGEW,
     1 ADD,PLOFQU,REWD,SEQN,CALPJ,DATADD,NEUZEI,TST,XINTPO,FISS,TSTS,
     2 FODATA
      REAL KURVE,MEDIUM,MEV,NULL,LTR,NEW,NEIN/'NEIN'/
      DIMENSION XINT(1),DELTA(1)
      DIMENSION BRR(6),ACONV(10),ACCNRV(32)
      COMMON XP(1000),YP(1000),ANZ
      COMMON/PARM1/ MAXOPT,PLEH,PLCZEI,ZXINT,IDPLOT,CALPJ
      COMMON/PARM2/ IPLOGR,ISTAFO,ISMO,IFMOUT,ISMNP,DISP,EINHTX,EINHTY,
     1 EINHTD,XEINHT,YEINHT,DEINHT,LTR,KARTEH,FMTOUT(20),STAFMT(20),
     2 FMT(20),DRUCK,STANZ,PLC,KURVE,PLOKU,DRUKU,STAKU,PLOSM,DRUSM,
     3 STASM,SMGEW,NEUZEI,XINTPO,PLOFQU,FISS,SEQN,ADD,DATADD,TST,
     4 REWD,NANZ,NACD,ZAHL,MEDIUM,WNR,
     5 RAST(2),ISMAX,FQU,NAMZ,ZAHLXR,TSTS,FODATA
      COMMON/PARM3/SMOO
      COMMON/PARM4/FROM,TO
      EQUIVALENCE (FROM,EMIN,VON,E1,EUNT),(TO,EMAX,BIS,E2,EOB)
      NAMELIST/NEUDAD/BDEH,ACCNR,DRUCK,STANZ,PLC,KURVE,PLOKU,DRUKU,STAKU
     1,SMOO,PLOSM,DRUSM,STASM,SMGEW,MEDIUM,KARTEH,FMT,ZAHL,WNR,XEINHT,YE
     2INHT,DEINHT,FISS,LTR,SEQN,FROM,TO,XINTPO,PLOZEI,DATADD,TST,DISP
     3,TSTS,EMIN,VON,E1,EUNT,EMAX,BIS,E2,EOB,FODATA
      DATA ACONV/'0   ','1   ','2   ','3   ','4   ','5   ','6   ','7   '
     1,'8   ','9   '/
      DATA ACCNRV/32*0/,PERC/'PERC'/
C----------READ CONTROL INPUT LIST.
      READ(KIN,NEUDAD)
      WRITE(KOUT,636)
      WRITE(KOUT,616) BDEH,ACCNR,LTR,FROM,TO,SEQN,DATADD,DRUCK,STANZ,PLO
     1,PLOZEI, KURVE,PLOKU,DRUKU,STAKU,DISP,XEINHT,YEINHT,DEINHT,
```

```
     2 FISS,XINTPO,FODATA
      IF(SMOO.NE.NEIN) GOTO 125
      WRITE(KOUT,617) SMOO,PLOSM,DRUSM,STASM,SMGEW,MEDIUM,KARTEH,FMT,
     1 ZAHL,WNR
      GOTO 126
  125 ISMO=1
      WRITE(KOUT,618) SMOO,PLOSM,DRUSM,STASM,SMGEW,MEDIUM,KARTEH,FMT,
     1ZAHL,WNR
C----------CONVERT DATA SET ID TO HOLLERITH.
C----------AND CHECK CURRENT TAPE POSITION.
  126 CALL ACONVE(ACCNR,BRR,ACONV)
      BRR(1)=LTR
      IF(.NOT.SEQN) GOTO 122
      IF(ACCNRV(BDEH).EQ.0) GOTO 122
      IF(ACCNRV(BDEH)-ACCNR) 121,122,122
  121 BACKSPACE BDEH
      GOTO 123
  122 REWIND BDEH
C----------FETCH DATA FROM TAPE.
  123 CALL BANDDT(BDEH,BRR,NR,SMGEW,DELTA,MEDIUM,ZAHL,FMT,KARTEH,
     1 WNR,FROM,TO,FODATA)
      IF(ANZ.EQ.0) GOTO 124
      ACCNRV(BDEH)=ACCNR
      IF(NR.EQ.1) ACCNRV(BDEH)=0
C----------CONVERT TO INTERNAL PHYSICAL UNITS.
      CALL ECONV(ANZ,XP,XEINHT)
      CALL ECONV(ANZ,YP,YEINHT)
C----------INTERPOLATION POINTS ARE TO BE READ.
      IF(.NOT.XINTPO) GOTO 127
      ZXINT=ANZ
      CALL STRING(XINT(1),XP(1),4*ZXINT)
      IF(IFMOUT.EQ.1) WRITE(KOUT,646)
      IFMOUT=1
C----------HANDLE ERROR BARS.
  127 IF(.NOT.SMGEW) RETURN
      IF(DEINHT.EQ.PERC) GOTO 128
      CALL ECONV(ANZ,DELTA,DEINHT)
      RETURN
  128 CALL PERCNT(ANZ,YP,DELTA)
      RETURN
  124 ACCNRV(BDEH)=0
      RETURN1
C
C----------INTERPOLATION POINTS ARE REQUESTED ONLY.
      ENTRY NEUINT(XINT,BDEH,ACCNR,*)
      WRITE(KOUT,648) FMTOUT,ZXINT,BDEH,LTR,ACCNR,SEQN,VON,BIS
      CALL ACONVE(ACCNR,BRR,ACONV)
      BRR(1)=LTR
      IF(.NOT.SEQN) GOTO 1695
      IF(ACCNRV(BDEH).EQ.0) GOTO 1695
      IF(ACCNRV(BDEH)-ACCNR) 1694,1695,1695
 1694 BACKSPACE BDEH
      GOTO 1696
 1695 REWIND BDEH
 1696 CALL NEUDEN(BDEH,BRR,VON,BIS,XINT,ZXINT,FODATA,&1697)
      CALL ECONV(ZXINT,XINT,EINHTX)
      WRITE(KOUT,651) EINHTX,ZXINT,(XINT(I),I=1,ZXINT)
      RETURN
 1697 IFMOUT=0
      RETURN 1
  616 FORMAT(///' &NEUDAD BDEH=',I3,', ACCNR=',I6,', LTR=',A2,',  FROM='
     X,E12.4,',  TO=',E12.4,',  SEQN=',L2,',  DATADD=',L2/
     15X,',  DRUCK=',L2,',  STANZ=',L2,',  PLO=',L2,',  PLOZEI=',I2/
     25X,'KURVE=',A4,',  PLOKU=',L2,',  DRUKU=',L2,',  STAKU=',L2,',  DI
     XSP=',A4/
     35X,'XEINHT=',A4,',  YEINHT=',A4,',  DEINHT=',A4,',  FISS=',L2,
     X',  XINTPO=',L2,',  FODATA=',L2)
  617 FORMAT(5X,'SMOO=',A4,', PLOSM=',L2,', DRUSM=',L2,', STASM=',L2/5X,
     1'SMGEW=',L2,', MEDIUM=',A4,', KARTEH=',I3,', FMT=',20A4/5X,'ZAHL='
     2,I5,', WNR=',I3)
  618 FORMAT(5X,'SMOO=',E12.5,', PLOSM=',L2,', DRUSM=',L2,', STASM=',L2/
     15X,'SMGEW=',L2,', MEDIUM=',A4,', KARTEH=',I3,', FMT=',20A4/5X,'ZAH
     2L=',I5,', WNR=',I3)
  636 FORMAT(2X,130('-'))
  646 FORMAT(//' ***** NACHRICHT'/8X,'DIE BISHERIGEN INTERPOLATIONSPUNKT
     1E WERDEN DURCH DIE NEUDADAPUNKTE ERSETZT'//)
  648 FORMAT(/' &OUTINT FMTOUT=',20A4/
     1 5X,'ZXINT=',A4,',  BDEH=',I2,',  LTR=',A2,',  ACCNR=',I6,',  SEQN
     1=',L2/5X,'VON=',E14.6,5X,'BIS=',E14.6/)
  651 FORMAT(//5X,'EINHTX=',A4,'  ZXINT=',I5,'   XINT='/(10E13.6))
      END



*PROCESS OPTIMIZE(1)
      SUBROUTINE LESEN1(KE,FMT,SMGEW,DELTA,FROM,TO,REWD,*)
C----------FORMATTED INPUT  READING ROUTINE.
      COMMON/INOUT/KOUT,KIN
      DIMENSION XP(1000),YP(1000),TEXT(20),FMT(20),DELTA(1000)
      INTEGER ANZ
      LOGICAL SMGEW,REWD
      COMMON XP,YP,ANZ
      NREC=0
C----------MAXIMUM IS 1000 DATA POINTS.
      IF(ANZ.LE.1000) GOTO 1
      WRITE(KOUT,607)
      ANZ=1000
C----------REWIND TO BE DONE?
    1 IF(.NOT.REWD) GOTO 6
C----------REJECT REWIND ON CONTROL INPUT UNIT.
      IF(KE.NE.KIN) GOTO 3
      WRITE(KOUT,601) KIN
      RETURN 1
    3 REWIND KE
C----------SEARCH FOR FROM,TO AS SPECIFIED.
    6 READ(KE,500,END=11,ERR=21) TEXT
      WRITE(KOUT,600) TEXT
      FF=-1.E-6*FROM+FROM
      FT=1.E-6*TO+TO
```

```
      DO 7 I=1,ANZ
    7 XP(I)=-1.E+10
    4 READ(KE,FMT,END=13,ERR=22) XX
      NREC=NREC+1
      IF(XX.GT.FT) GOTO 14
      IF(XX.GE.FF) GOTO 5
      GOTO 4
C----------FOUND. BACKSPACE TO BE ON SAFE SIDE.
    5 BACKSPACE KE
      IF(NREC.EQ.1) GOTO 2
      BACKSPACE KE
      GOTO 2
C----------READ DATA.
C----------REMEMBER THAT AN EVEN NUMBER OF DATA ITEMS
C----------IS PRESCRIBED FOR EACH RECORD.
    2 IF(SMGEW) GOTO 30
      READ(KE,FMT,END=12,ERR=22) (XP(I),YP(I),I=1,ANZ)
      GOTO 40
   11 WRITE(KOUT,603)
      RETURN 1
C----------SEARCH FOR END OF DATA.
   12 WRITE(KOUT,604) TEXT
      WRITE(KOUT,611) (XP(I),YP(I),I=1,ANZ)
  611 FORMAT(' CONTENTS OF ARRAYS XP AND YP:'/(2E15.5))
      DO 15 I=1,ANZ
C----------READ IN CASE ERROR BARS ARE TO BE READ ALSO.
      IF(XP(I).EQ.-1.E+10) GOTO 17
      IF(XP(I).EQ.0..AND.I.NE.1.AND.XP(I-1).GT.XP(I)) GOTO 17
      GOTO 15
   17 J=I-1
      GOTO 16
   15 CONTINUE
      J=ANZ
   16 ANZ=J
      IF(ANZ.NE.0) GOTO 40
      WRITE(KOUT,609)
      RETURN 1
   13 WRITE(KOUT,602) KE
      RETURN 1
   14 WRITE(KOUT,608)
      RETURN 1
   21 WRITE(KOUT,605)
      GOTO 2
   22 BACKSPACE KE
      WRITE(KOUT,606)
      RETURN 1
   30 READ(KE,FMT,END=12,ERR=22) (XP(I),YP(I),DELTA(I),I=1,ANZ)
      GOTO 40
C----------SKIP ALL DATA OUTSIDE SPECIFIED INTERVAL (FROM,TO)
   40 DO 42 I=1,ANZ
      IF(XP(I).LE.FT) GOTO 42
      ANZ=I-1
      GOTO 44
   42 CONTINUE
   44 DO 46 I=1,ANZ
      IF(XP(I).LT.FF) GOTO 46
      JI=I-1
      GOTO 48
   46 CONTINUE
   48 IF(JI.EQ.0) GOTO 50
      ANZ=ANZ-JI
      CALL STRING(XP(1),XP(JI+1),4*ANZ)
      CALL STRING(YP(1),YP(JI+1),4*ANZ)
      CALL STRING(DELTA(1),DELTA(JI+1),4*ANZ)
   50 WRITE(KOUT,610) ANZ
      RETURN
  500 FORMAT(20A4)
  600 FORMAT(10X,20A4/)
  601 FORMAT(//5X,'REWIND AUF',I2,'WURDE VERLANGT.  AKTION: REWIND WIRD
     XNICHT AUSGEFUEHRT.DATENSATZ WIRD NICHT VERARBEITET.'//)
  602 FORMAT(//5X,'DATENFILE ENDE AUF EINHEIT',I2,' .KEIN WERT >FROM WUR
     XDE GEFUNDEN'/
     18X,'AKTION: KEINE VERARBEITUNG'//)
  603 FORMAT(//5X,'FEHLERNACHRICHT'/8X,'BEIM VERSUCH,DEN DATENTITEL EINZ
     1ULESEN WURDE DAS DATENFILEENDE ERREICHT'/5X,'AKTION:DATENSATZ WIRD
     X NICHT VERARBEITET'//)
  604 FORMAT(//5X,'FEHLERNACHRICHT'/8X,'BEIM VERSUCH,INFORMATION ZU'/8X,
     1'***',20A4,'***'/8X,'EINZULESEN,WURDE DAS ENDE DES DATENFILES ERRE
     2ICHT'/5X,'AKTION:BISHER ERHALTENE WERTE WERDEN VERARBEITET'//)
  605 FORMAT(//5X,'FEHLERNACHRICHT'/8X,'BEIM VERSUCH,DEN DATENTITEL EINZ
     1ULESEN TRAT EIN LESEFEHLER AUF'/5X,'AKTION:DATENTITEL WIRD UEBERGA
     2NGEN')
  606 FORMAT(//5X,'FEHLERNACHRICHT'/8X,'LESEFEHLER BEIM EINLESEN VON INF
     1ORMATION ZU'/8X,'***',20A4,'***'/5X,'AKTION:AUSFUEHRUNG DES LETZTE
     2N STEUERBEFEHLS WIRD ABGEBROCHEN.NAECHSTER STEUERBEFEHL WIRD GESUC
     3HT')
  607 FORMAT(//'  ANZ > 1000 NICHT ERLAUBT.DATENSATZ WIRD GEKUERZT'//)
  608 FORMAT(//5X,'KEINE DATEN IM ANGEGEBENEN INTERVALL'//)
  609 FORMAT(10X,'SINCE NO DATA WERE READ BEFORE END OF DATA WAS ENCOUNT
     1ERED,NO PROCESSING IS ATTEMPTED.'//)
  610 FORMAT(//5X,'NUMBER OF DATAPOINTS TRANSMITTED=',I4)
      END
```

```
      SUBROUTINE PLOGXY(XP,YP,ANZ,NCUR,NP,NPA,INDZ,NTEXT,
     1XL,YL,RAST,XMAL,XMIL,XMAX,XMIN,SX,YMAL,YMIL,YMAX,YMIN,SY,
     2ICPLOT,NLGGX,NLGGY,NX,NY,XSKA,YSKA,BSK)
C----------PLOTTING ROUTINE FOR LOG. PLOTS.
      DIMENSION XDUM(2),YDUM(2),NDIR(10),NSC(10),XSKA(100),YSKA(100),
     1XP(2),YP(2),XL(2),YL(2),RAST(2),NTEXT(1)
      DATA NDIR/10*2/,NSC/10*1/
      REAL*8 BSK(200)
      INTEGER ANZ
      DATA XLOG/'XLOG'/,YLOG/'YLOG'/
C----------CONVERT DATA TO LOG.
      CALL CONLOG(XP,XL,ANZ,RAST(1),XLOG)
      CALL CONLOG(YP,YL,ANZ,RAST(2),YLOG)
      IF(INDZ.EQ.0.OR.INDZ.EQ.10) GOTO 50
```

```
      MC=1                                                              150
      DX=80.*SX                                                         160
      DY=80.*SY                                                         170
      IF(INDZ.GT.10) DX=DX+DX                                           180
      IF(INDZ.GT.10) DY=DY+DY                                           190
      IF(NX.EQ.0) DX=XMAX-XMIN                                          200
      IF(NY.EQ.0) DY=YMAX-YMIN                                          210
C----------PLOT AXIS.                                                  220
      CALL PLOTA(XL,YL,ANZ,NCUR,NP,1,1,NPA,INDZ,XMAL,XMIL,SX,           230
     1YMAL,YMIL,SY,NTEXT,IDPLOT,                                        240
     2NLGGX,XMIN,DX,XMAX,4HE9.2,1,-1,+1,NX,                             250
     3NLGGY,YMIN,DY,YMAX,4HE9.2,1,+1,-1,NY,0)                           260
      IF(INDZ.GT.10) L=10                                               270
      IF(INDZ.LT.10) L=0                                                280
      XDUM(1)=XMIL                                                      290
      XDUM(2)=XDUM(1)                                                   300
      YDUM(1)=YMIL                                                      310
      YDUM(2)=YDUM(1)                                                   320
      IF(NX.NE.0) GOTO 20                                               330
C----------PLOT SCALE.                                                 340
      CALL SKALA(XMIN,XMIL,XMAL,SX,YMIL,SY,NSKA,NPLOS,LASKA,            350
     1XSKA,YSKA,BSK,RAST(1),XLOG,1)                                     360
      IF(INDZ.LT.10.OR.RAST(1).EQ.XLOG) GOTO 10                         370
      CALL CALSKA(BSK,XSKA,YSKA,NPLOS,LASKA)                            380
C     M=10*NPLOS+LASKA                                                  390
C     WRITE(KOUT,600) M,(XSKA(I),YSKA(I),BSK(2*I-1),BSK(2*I),I=1,M)     400
C 600 FORMAT(I6,'  SKALENPUNKTE'/(2E12.3,5X,2A8))                       410
   10 IF(NPLOS.EQ.0) GOTO 12                                            420
C----------IDENTIFIERS TO BE PLOTTED ALONG AXIS.                       430
      DO 11 N=1,NPLOS                                                   440
   11 CALL PLOTA(XDUM,YDUM,2,2,0,1,1,1,L,XMAL,XMIL,SX,YMAL,YMIL,SY,1,   450
     1IDPLOT,0,0,                                                       460
     210,XSKA(10*N-9),YSKA(10*N-9),NDIR,NSC,                            470
     3BSK(20*N-19),BSK(20*N-17),BSK(20*N-15),BSK(20*N-13),BSK(20*N-11),  480
     4BSK(20*N- 9),BSK(20*N- 7),BSK(20*N- 5),BSK(20*N- 3),BSK(20*N- 1))  490
   12 IF(LASKA.EQ.0) GOTO 20                                            500
      CALL PLOTA(XDUM,YDUM,2,2,0,1,1,1,L,XMAL,XMIL,SX,YMAL,YMIL,SY,1,   510
     1IDPLOT,0,0,                                                       520
     2LASKA,XSKA(10*NPLOS+1),YSKA(10*NPLOS+1),NDIR,NSC,                 530
     3BSK(20*NPLOS+1),BSK(20*NPLOS+3),BSK(20*NPLOS+5),                  540
     4BSK(20*NPLOS+7),BSK(20*NPLOS+9),BSK(20*NPLOS+11),                 550
     5BSK(20*NPLOS+13),BSK(20*NPLOS+15),BSK(NPLOS*20+17))               560
   20 IF(MC.EQ.2) GOTO 60                                               570
      IF(NY.NE.0) GOTO 60                                               580
      MC=MC+1                                                           590
      CALL SKALA(YMIN,YMIL,YMAL,SY,XMIL,SX,NSKA,NPLOS,LASKA,YSKA,XSKA,  600
     1BSK,RAST(2),YLOG,2)                                               610
C     M=10*NPLOS+LASKA                                                  620
C     WRITE(KOUT,600) M,(XSKA(I),YSKA(I),BSK(2*I-1),BSK(2*I),I=1,M)     630
      GOTO 10                                                           640
   50 CALL PLOTA(XL,YL,ANZ,NCUR,NP,1,1,NPA,INDZ,XMAL,XMIL,SX,           650
     1YMAL,YMIL,SY,1,IDPLOT,0,0,0)                                      660
      GOTO 60                                                           670
   60 RETURN                                                            680
      END                                                               690
```

```
      SUBROUTINE KURVE1(IM,X,Y,M,T,S,S2,A,B,C,D,KENN,*)                 10
C----------CONTROL INTERPOLATION OF DATA POINTS TO CURVE.               20
C----------KURVE1,KURVE2 FOR AUTOMATIC SELECTION OF                    30
C----------INTERPOLATION POINTS (FOR PLOTS ONLY).                      40
      COMMON/INOUT/KOUT                                                 50
      DIMENSION X(IM),Y(IM),T(1000),S(1000),S2(1000),A(1000),B(1000),C(1 60
     1000),D(1000)                                                      70
      GOTO 10                                                           80
      ENTRY KURVE2(IM,X,Y,M,T,S,A,B,C,D,*)                              90
      KENN=0                                                           100
   10 NUMP=999/(IM-1)                                                  110
      M=(IM-1)*NUMP+1                                                  120
      FNUMP=FLOAT(NUMP)                                                130
      J=0                                                              140
      MAX=M-1                                                          150
      DO 6 I=1,MAX,NUMP                                                160
      J=J+1                                                            170
      T(I)=X(J)                                                        180
      IF(NUMP.EQ.1) GOTO 6                                             190
      DT=(X(J+1)-X(J))/FNUMP                                           200
      DO 7 L=2,NUMP                                                    210
    7 T(I+L-1)=T(I+L-2)+DT                                             220
    6 CONTINUE                                                         230
      T(M)=X(IM)                                                       240
      GOTO 8                                                           250
    8 IF(KENN.NE.0) GOTO 20                                            260
C----------KURVE 3,KURVE4 FOR GIVEN INTERPOLATION POINTS.              270
      ENTRY KURVE3(IM,X,Y,M,T,S,A,B,C,D,NA,NB,*)                       280
C----------CHECK RANGE OF INTERPOLATION POINTS.                        290
      CALL PCHECK(IM,X,M,T,NA,NB)                                      300
      IF(NA.GT.NB) RETURN1                                             310
      K=1                                                              320
      DO 4 I=NA,NB                                                     330
    2 IF(T(I).LT.X(K+1)) GOTO 3                                        340
      K=K+1                                                            350
      IF(K.NE.IM) GOTO 2                                               360
      IF(I.EQ.NB) GOTO 5                                               370
      WRITE(KOUT,604)                                                  380
      RETURN                                                           390
    5 K=K-1                                                            400
    3 DT=T(I)-X(K)                                                     410
      S(I)=A(K)+B(K)*DT+C(K)*DT**2+D(K)*DT**3                          420
    4 CONTINUE                                                         430
      RETURN                                                           440
      ENTRY KURVE4(IM,X,Y,M,T,S,S2,A,B,C,D,NA,NB,KENN,*)               450
   20 CALL PCHECK(IM,X,M,T,NA,NB)                                      460
      IF(NA.GT.NB) RETURN1                                             470
      MM=NB-NA+1                                                       480
      IF(KENN.EQ.1) GOTO 30                                            490
      IF(IM.LT.4) GOTO 41                                              500
      CALL SPLINE(IM,X,Y,S2,0,0,MM,T(NA),S(NA),NR,A,E,C,D)             510
```

```
      WRITE(KOUT,600) NR                                                    520
      RETURN                                                                530
   30 IF(IM.LT.2) GOTO 40                                                   540
      CALL LINEAR(IM,X,Y,MM,T(NA),S(NA),NR)                                 550
      WRITE(KOUT,601) NR                                                    560
      RETURN                                                                570
   40 WRITE(KOUT,602)                                                       580
      RETURN1                                                               590
   41 WRITE(KOUT,603)                                                       600
      RETURN1                                                               610
  600 FORMAT(//2X,'SPLINENACHRICHT,NR = ',I3)                               620
  601 FORMAT(//2X,'NACHRICHT DER LINEAREN INTERPOLATION,NR = ',I3)          630
  602 FORMAT(//5X,'FEHLERNACHRICHT'/8X,'KURVE KANN NICHT GELEGT WERDEN.F    640
     1NTWEDER IST DIE ANZAHL DER PUNKTE=1 ODER X(ANZ)-X(1) KLEINER ALS E    650
     2IN PLOTTERSCHRITT'/5X,'AKTION:ES WIRD KEINE KURVE GELEGT UND AUCH     660
     3NICHT AUSGEGEBEN'//)                                                  670
  603 FORMAT(//5X,'FEHLERNACHRICHT'/8X,'SPLINE WURDE GEWUENSCHT.ANZAHL D    680
     1ER ANGELIEFERTEN PUNKTE IST < 4'/5X,'AKTION:ES WIRD KEINE SPLINE I    690
     2NTERPOLATION VORGENOMMEN'//)                                          700
  604 FORMAT(//'  ***** ERROR IN SUBROUTINE KURVE1.'//)                     710
      END                                                                   720
```

```
      SUBROUTINE INIT                                                        10
      INTEGER PLEH,ZXINT,WNR,ZAHL,PLOZEI,ZAHLXR                              20
      COMMON/INOUT/KOUT,KIN,KPUN,KINC                                        30
      LOGICAL DRUCK,STANZ,PLC,PLOKU,DRUKU,STAKU,PLOSM,DRUSM,STASM,SMGEW,     40
     1 NEUZEI,XINTPO,PLOFQU,FISS,SEQN,ADD,CALP,CALPJ,DATADD,TST,REWD         50
     2 ,TSTS,FODATA                                                          60
      REAL NEIN,KURVE,MEDIUM,MEV,NULL,LTR,NEW                                70
      DIMENSION FMTA(2)                                                      80
      COMMON/PARM1/ MAXOPT,PLEH,PLOZEI,ZXINT,IDPLOT,CALPJ                    90
        COMMON/CLP/ CALP                                                    100
      COMMON/PARM2/ IPLOGR,ISTAFO,ISMO,IFMOUT,ISMNP,DISP,EINHTX,EINHTY,     110
     1 EINHTO,XEINHT,YEINHT,DEINHT,LTR,KARTEH,FMTOUT(20),STAFMT(20),        120
     2 FMT(20),DRUCK,STANZ,PLC,KURVE,PLOKU,DRUKU,STAKU,PLOSM,DRUSM,         130
     3 STASM,SMGEW,NEUZEI,XINTPO,PLOFQU,FISS,SEQN,ADD,DATADD,TST,           140
     4 REWD,NANZ,NADD,ZAHL,MEDIUM,WNR,                                      150
     5 RAST(2),ISMAX,FQU,NAMZ,ZAHLXR,TSTS,FODATA                            160
      COMMON/PARM4/FROM,TO                                                  170
      DATA NEW/'NEW'/,EV/'EV'/,BARN/'BARN'/,MEV/'MEV'/,NULL/'0'/,           180
     1 NEIN/'NEIN'/,BLANK/'    '/,FMTA/'(2E1','4.6)'/                       190
C                                                                           200
C---------PROGRAM INITIALISATION.                                          210
       ENTRY INIT1                                                          220
      KOUT=6                                                                230
      KIN=9                                                                 240
      KINC=5                                                                250
      KPUN=8                                                                260
      MAXOPT=7                                                              270
      PLEH=1                                                                280
      PLOZEI=-1                                                             290
      ZXINT=0                                                               300
      IDPLOT=0                                                              310
      CALPJ=.FALSE.                                                         320
      RETURN                                                                330
C                                                                           340
C----------RT  INITIALISATION.                                            350
      ENTRY INIT2                                                           360
      IPLOGR=0                                                              370
      ISTAFO=0                                                              380
      ISMO=0                                                                390
      IFMOUT=0                                                              400
      ISMNP=0                                                               410
      DISP=NEW                                                              420
      EINHTX=EV                                                             430
      EINHTY=BARN                                                           440
      EINHTO=BARN                                                           450
      XEINHT=MEV                                                            460
      YEINHT=BARN                                                           470
      DEINHT=BARN                                                           480
      LTR=NULL                                                              490
      KARTEH=KIN                                                            500
      DO 1001 I=1,2                                                         510
      FMTOUT(I)=FMTA(I)                                                     520
      STAFMT(I)=FMTA(I)                                                     530
 1001 FMT(I)=FMTA(I)                                                        540
      DO 1002 I=3,20                                                        550
      FMTOUT(I)=BLANK                                                       560
      STAFMT(I)=BLANK                                                       570
 1002 FMT(I)=BLANK                                                          580
      DRUCK=.FALSE.                                                         590
      STANZ=.FALSE.                                                         600
      PLC=.TRUE.                                                            610
      KURVE=NEIN                                                            620
      PLOKU=.TRUE.                                                          630
      DRUKU=.FALSE.                                                         640
      STAKU=.FALSE.                                                         650
      PLOSM=.TRUE.                                                          660
      DRUSM=.FALSE.                                                         670
      STASM=.FALSE.                                                         680
      SMGEW=.FALSE.                                                         690
            NEUZEI=.FALSE.                                                  700
      XINTPO=.FALSE.                                                        710
      PLOFQU=.TRUE.                                                         720
      FISS=.FALSE.                                                          730
      SEQN=.TRUE.                                                           740
      CALP=.FALSE.                                                          750
      ADD=.FALSE.                                                           760
      DATADD=.FALSE.                                                        770
      TST=.FALSE.                                                           780
      REWD=.FALSE.                                                          790
      NANZ=0                                                                800
      NADD=0                                                                810
      FROM=0.                                                               820
      TO=1.E+20                                                             830
      ZAHL=0                                                                840
      MEDIUM=BLANK                                                          850
```

```
      WNR=54                                                            860
      RAST(1)=NEIN                                                      870
      RAST(2)=BLANK                                                     880
      ISMAX=0                                                           890
      FCU=0.                                                            900
      NAMZ=2                                                            910
      ZAHLXR=0                                                          920
      TSTS=.FALSE.                                                      930
      FODATA=.TRUE.                                                     940
      RETURN                                                            950
      END                                                               960
      SUBROUTINE BANDDT(KB,BLL,NR,SMGEW,DELTA,MEDIUM,ZAHL,FMT,KE,WNR,    10
     1 FROM,TO,FODATA)                                                  20
C----------READ NEUDADA TAPE.                                           30
      COMMON/INOUT/KOUT,KIN                                             40
      DIMENSION BLL(6),BAND(54),XP(1000),YP(1000),DELTA(1000),FMT(20)   50
      REAL MEDIUM                                                       60
      INTEGER ZAHL,WNR                                                  70
      LOGICAL SMGEW,FODATA                                              80
      COMMON XP,YP,IM                                                   90
      DATA CARD/'CARD'/,TAPE/'TAPE'/                                   100
      CALL ERRSET(214,200,-10,1,1)                                     110
      IERRN=0                                                          120
      NR=0                                                             130
      IM=0                                                             140
      IP=1                                                             150
C----------SEARCH FOR CORRECT ID.                                       160
   19 IF(FODATA) READ(KB,101,END=30,ERR=28) (BAND(I),I=1,54)           170
      IF(.NOT.FODATA) READ(KB,120,END=30,ERR=28) (BAND(I),I=1,54)      180
      IF(IERRN.EQ.2) GOTO 81                                           190
   20 CALL BDVGL(BAND,BLL,&19)                                         200
C----------IF FOUND, VERIFY.                                          210
      WRITE(KOUT,113) (BAND(I),I=1,31)                                 220
C----------DETERMINE PHYSICAL UNITS OF ERROR BARS AND RESOLUTION.      230
      WRITE(KOUT,117)                                                  240
      CALL ERRDET(BAND(33),IRTYP)                                      250
      WRITE(KOUT,112)                                                  260
      CALL ERRDET(BAND(39),IRTYP)                                      270
      IF(.NOT.SMGEW) GOTO 18                                           280
      IF(MEDIUM.EQ.CARD) GOTO 18                                       290
      IF(MEDIUM.EQ.TAPE) GOTO 40                                       300
      WRITE(KOUT,102)                                                  310
      SMGEW=.FALSE.                                                    320
      GOTO 18                                                          330
   40 XX=BAND(48)                                                      340
C----------SEQRCH FOR FROM,TO OR END OF THIS ID.                       350
      IF(XX.LT.FROM) GOTO 21                                           360
      IF(XX.GT.TO) GOTO 31                                             370
      IM=IM+1                                                          380
      IF(IM.GT.1000) GOTO 92                                           390
      XP(IM)=BAND(48)                                                  400
      YP(IM)=BAND(53)                                                  410
      DELTA(IM)=BAND(WNR)                                              420
   21 IF(FODATA) READ(KB,101,END=27,ERR=25) (BAND(I),I=1,54)           430
      IF(.NOT.FODATA) READ(KB,120,END=27,ERR=25) (BAND(I),I=1,54)      440
      IP=2                                                             450
      IF(IERRN.EQ.2) GOTO 81                                           460
   22 CALL BDVGL(BAND,BLL,&27)                                         470
      GOTO 40                                                          480
   18 XX=BAND(48)                                                      490
      IRTYP=0                                                          500
      IF(XX.LT.FROM) GOTO 23                                           510
      IF(XX.GT.TO) GOTO 31                                             520
      IM=IM+1                                                          530
      IF(IM.GT.1000) GOTO 92                                           540
      XP(IM)=BAND(48)                                                  550
      YP(IM)=BAND(53)                                                  560
   23 IF(FODATA) READ(KB,101,END=27,ERR=29) (BAND(I),I=1,54)           570
      IF(.NOT.FODATA) READ(KB,120,END=27,ERR=29) (BAND(I),I=1,54)      580
      IP=3                                                             590
      IF(IERRN.EQ.2) GOTO 81                                           600
   24 CALL BDVGL(BAND,BLL,&27)                                         610
      GOTO 18                                                          620
   27 WRITE(KOUT,107) KB,BLL,IM                                        630
      IF(IRTYP.NE.2) GOTO 26                                           640
      IF(IM.EQ.0)GOTO 26                                               650
      DO 32 I=1,IM                                                     660
   32 DELTA(I)=DELTA(I)*YP(I)*.01                                      670
      GOTO 26                                                          680
C----------ERROR EXITS.                                               690
   28 WRITE(KOUT,108)  BLL                                             700
      IP=1                                                             710
      GOTO 80                                                          720
   25 WRITE(KOUT,116)  BLL                                             730
      IP=2                                                             740
      GOTO 80                                                          750
   29 WRITE(KOUT,109)  BLL                                             760
      IP=3                                                             770
      GOTO 80                                                          780
   31 IF(IM.NE.0) GOTO 27                                              790
      NR=1                                                             800
      WRITE(KOUT,106) KB,BLL                                           810
      GOTO 26                                                          820
   30 WRITE(KOUT,110) BLL                                              830
      NR=1                                                             840
C----------IF ERROR BARS ARE DESIRED, CONTINUE. ELSE RETURN.           850
   26 IF(.NOT.SMGEW) RETURN                                            860
      IF(MEDIUM.EQ.TAPE) RETURN                                        870
C----------ERROR BARS ARE TO BE READ FROM CARDS.                       880
      IF(ZAHL.EQ.IM) GOTO 50                                           890
      WRITE(KOUT,103) IM,ZAHL                                          900
   50 READ(KE,FMT,END=90,ERR=91) (DELTA(I),I=1,ZAHL)                   910
C----------CHECK IF NUMBER OF DATA AGREE.                             920
      IF(ZAHL.GT.IM.OR.ZAHL.EQ.IM) RETURN                              930
      IANF=ZAHL+1                                                      940
      DO 51 I=IANF,IM                                                  950
```

```
 51 DELTA(I)=DELTA(ZAHL)                                                 960
    RETURN                                                               970
 80 WRITE(KOUT,114)                                                      980
    READ(KB)                                                             990
    IERRN=2                                                             1000
    GOTO(19,21,23),IP                                                   1010
 81 WRITE(KOUT,115) BAND                                                1020
    IERRN=0                                                             1030
    GOTO(20,22,24),IP                                                   1040
 90 WRITE(KOUT,104)                                                     1050
    SMGEW=.FALSE.                                                       1060
    RETURN                                                              1070
 91 WRITE(KOUT,105)                                                     1080
    SMGEW=.FALSE.                                                       1090
    BACKSPACE KE                                                        1100
    RETURN                                                              1110
 92 WRITE(KOUT,111)                                                     1120
    IM=IM-1                                                             1130
    GOTO 27                                                             1140
101 FORMAT(47A1,7E12.5,1X)                                              1150
120 FORMAT(47A1,7A4)                                                    1160
102 FORMAT(//5X,'FEHLERNACHRICHT'/8X,'SMGEW=T GEFUNDEN.MEDIUMANWEISUNG 1170
   1 FEHLT'/5X,'AKTION:SMGEW=F GESETZT.DADURCH WERDEN DIE GEWICHTE AUT 1180
   2OMATISCH = 1'//)                                                    1190
103 FORMAT(//5X,'FEHLERNACHRICHT'/8X,'DIE ANZAHL DER AUF NEUDADA GEFUN 1200
   1DENEN DATENPUNKTE IM=',I5/8X,'STIMMT NICHT MIT DER IN DER NEUDAD - 1210
   2 KARTE ANGEGEBENEN ANZAHL ZAHL=',I5,'UEBEREIN'/5X,'AKTION:FALLS IM 1220
   3>ZAHL ERHALTEN DIE LETZTEN IM-ZAHL DATENPUNKTE DAS GEWICHT DES ZAH 1230
   4L-TEN PUNKTES'/8X,'FALLS IM<ZAHL WERDEN DIE LETZTEN ZAHL-IM GEWICH 1240
   5TE IGNORIERT'//)                                                    1250
104 FORMAT(//5X,'FEHLERNACHRICHT'/8X,'BEIM EINLESEN DER SMOOTHGEWICHTE 1260
   1 WURDE DAS ENDE DES DATENFILES ERREICHT'/5X,'AKTION:SMGEW=FALSE GE 1270
   2SETZT'//)                                                           1280
105 FORMAT(//5X,'FEHLERNACHRICHT'/8X,'BEIM EINLESEN DER SMOOTHGEWICHTE 1290
   1 TRAT EIN LESEFEHLER AUF'/5X,'AKTION:SMGEW=FALSE GESETZT'//)        1300
106 FORMAT(' *****FEHLERNACHRICHT'/8X,'ANZAHL DER DATENPUNKTE AUF NEUD 1310
   1ADABAND',I3,' MIT ACCNR=',6A1/                                      1320
   28X,' = 0 ZWISCHEN DEN ANGEGEBENEN ENERGIEGRENZEN')                  1330
107 FORMAT(5X,'ANZAHL DER AUF DEM NEUDADABAND ',I3,' GEFUNDENEN DATEN 1340
   1MIT ACCESSNUMMER ',6A1,' = ',I5)                                    1350
108 FORMAT(2X,6A1,' FEHLER IN 19 READ')                                 1360
109 FORMAT(2X,6A1,' FEHLER IN 23 READ')                                 1370
110 FORMAT(2X,6A1,'NICHT GEFUNDEN,BANDENDE ERREICHT')                   1380
111 FORMAT(//'  ANZAHL DER UEBERTRAGENEN DATENPUNKTE HAT DIE MAXIMALZA 1390
   1HL 1000 ERREICHT'/'     REST WIRD ABGESCHNITTEN.'//)                1400
112 FORMAT(/'  ERRORTYPE OF Y-VALUES:')                                 1410
113 FORMAT(////5X,31A1/)                                                1420
114 FORMAT(//5X,'NACHRICHT'/8X,'  RECORDS WURDEN AUSGELASSEN WEGEN IHC 1430
   1218-ERROR'/8X,'ES IST ZU UEBERPRUEFEN,OB DARUNTER KEINES DER BEIM  1440
   2AUFRUF GEWUENSCHTEN IST'/8X,'ZU DIESEM ZWECK WIRD VERSUCHT,DAS NAC 1450
   3HFOLGENDE RECORD AUSZUDRUCKEN')                                     1460
115 FORMAT(5X,'RECORD LAUTET:'/1X,47A1,7E12.5//)                        1470
116 FORMAT(2X,6A1,' FEHLER IN 21 READ')                                 1480
117 FORMAT(/' ERRORTYPE OF X-VALUES:')                                  1490
    END                                                                 1500
```

```
      SUBROUTINE TSTPRO(ISMAX,XS,YS,P,A,FQU)                             10
C----------TEST PRINTOUT.                                               20
      DIMENSION P(1),A(1),XS(1),YS(1)                                    30
      COMMON/INOUT/KOUT                                                  40
      WRITE(KOUT,604)                                                    50
  604 FORMAT('1')                                                        60
      WRITE(KOUT,600)                                                    70
  600 FORMAT(    50X,'TESTPRINTOUT OF SMOOTH RESULTS'///)                80
      WRITE(KOUT,602)                                                    90
      DDX=0                                                             100
      DATA LLM/5)/                                                      110
      L=0                                                               120
      DO 100 I=1,ISMAX                                                  130
      IF(L.LT.LLM) GOTO 10                                              140
      WRITE(KOUT,604)                                                   150
      WRITE(KOUT,602)                                                   160
      L=0                                                               170
   10 DX=ABS(YS(I)-A(I))                                                180
      DDP=DX/P(I)                                                       190
      DDX=DDX+DDP**2                                                    200
  601 FORMAT(1P6E15.5)                                                  210
  602 FORMAT(    T8,'ENERGY',T19,'EXPERIMENTAL',T36,'FITTED',T49,'ABS(FI 220
     1T-EXP)',T65,'EXP.ERROR',T79,'D(FIT)/D(EXP)'/)                     230
      WRITE(KOUT,601) XS(I),YS(I),A(I),DX,P(I),DDP                      240
  100 L=L+1                                                             250
      WRITE(KOUT,603) FQU,DDX                                           260
  603 FORMAT(///10X,'FQU=',1PE13.5,5X,'RESIDUAL OF FIT=',1PE13.5///)    270
      RETURN                                                            280
      END                                                               290
```

```
C----------DATA POINTS WITH EQUAL ENERGIES ARE NOT PERMITTED FOR          10
C----------SMOOTH. THESE DATA POINTS ARE COMBINED TO A WEIGHTED AVERAGE   20
C----------IN THIS ROUTINE.                                              30
      SUBROUTINE EQUEN(MAX,X,Y,P)                                        40
      COMMON/INOUT/KOUT                                                  50
      DIMENSION X(1),Y(1),P(1)                                           60
      DATA VH/1.00001/                                                   70
      IF(MAX.LT.2) GOTO 200                                              80
      MT=0                                                               90
      MP=0                                                              100
      DO 100 I=2,MAX                                                    110
      IF(X(I).GT.VH*X(I-1)) GOTO 90                                     120
      MP=MP+1                                                           130
      GOTO 99                                                           140
   80 IF(MT.EQ.0) WRITE(KOUT,600)                                       150
  600 FORMAT(///20X,'DATAPOINTS WITH EQUAL ENERGIES ARE NOT ALLOWED FOR 160
     1SMOOTH'//T20,'FOR THE FOLLOWING ENERGIES DATAPOINTS WERE COMBINED 170
     1FOR THIS REASON.'//)                                              180
```

```
      WRITE(KOUT,601) X(I-MP)
  601 FORMAT(10X,1PE13.5)
      MT=MT+MP
      MP=MP+1
      YY=0
      PP=0
      PS=0
      DO 81 J=1,MP
      YY=YY+Y(I-J)/P(I-J)
      PP=PP+1./P(I-J)
   81 PS=PS+P(I-J)**2
      Y(I-MT-1)=YY/PP
      P(I-MT-1)=SQRT(PS)/MP
      MP=0
      GOTO 92
   90 IF(MP.NE.0) GOTO 80
   92 Y(I-MT)=Y(I)
      X(I-MT)=X(I)
      P(I-MT)=P(I)
      GOTO 100
   99 IF(I.NE.MAX) GOTO 100
      GOTO 80
  100 CONTINUE
      MAX=MAX-MT
  200 RETURN
      END
```

```
      SUBROUTINE LESEN(KE,FMT,RWD,N1,N2,XMIN,XMAX,X,*)
C----------READ INTERPOLATION ENERGIES FROM FORMATTED INPUT, E.G. CARDS.
      DIMENSION X(1),FMT(1),TEXT(20)
      COMMON/INOUT/KOUT,KIN
      LOGICAL RWD
      NREC=0
      IF(N2.LE.1000) GOTO 1
      WRITE(KOUT,607)
  607 FORMAT(//'  ANZ > 1000 NICHT ERLAUBT.DATENSATZ WIRD GEKUERZT'//)
      N2=1000
C----------REWIND? IF ON CONTROL INPUT, REJECT.
    1 IF(.NOT.RWD) GOTO 6
      IF(KE.NE.KIN) GOTO 3
      WRITE(KOUT,601) KIN
  601 FORMAT(//5X,'REWIND AUF',I2,'WURDE VERLANGT.  AKTION: REWIND WIRD
     1NICHT AUSGEFUEHRT.DATENSATZ WIRD NICHT VERARBEITET.'//)
      RETURN 1
    3 REWIND KE
    6 READ(KE,500,END=11) TEXT
C----------REMEMBER, THAT COMMENT MUST PRECEDE.
  500 FORMAT(20A4)
      WRITE(KOUT,600) TEXT
  600 FORMAT(//'  COMMENT=',2CA4)
      FF=XMIN*0.999999
      FT=XMAX*1.000001
      DO 7 I=1,N2
    7 X(I)=-1.E+10
C----------SEARCH FOR REQUESTED INTERVAL.
C----------ALLOWANCE FOR MACHINE INTERNAL INACCURACIES
C----------WAS PROVIDED.
    4 READ(KE,FMT,END=11) XX
      NREC=NREC+1
      IF(XX.GT.FT) GOTO 14
      IF(XX.GE.FF) GOTO 5
      GOTO 4
    5 BACKSPACE KE
      IF(NREC.EQ.1) GOTO 2
      BACKSPACE KE
      GOTO 2
    2 READ(KE,FMT,END=12) (X(I),I=1,N2)
      GOTO 40
   11 WRITE(KOUT,603) KE
  603 FORMAT(//'      END OF DATA ON UNIT=',I3,'. DATASET IS NOT TRANSMI
     1TTED'//)
      RETURN 1
C----------END OF DATA. TRANSMIT REST.
   12 WRITE(KOUT,604) KE
  604 FORMAT(//5X,'END OF DATA ON UNIT ',I3,' ENCOUNTERED.DATA ALREADY R
     1EAD ARE USED.'//)
      DO 15 I=1,N2
      IF(X (I).NE.-1.E+10) GOTO 15
      J=I-1
      GOTO 16
   15 CONTINUE
      J=N2
   16 N2=J
      IF(N2.NE.0) GOTO 40
      WRITE(KOUT,609)
  609 FORMAT(5X,'SINCE NO DATA WERE READ ,NO PROCESSING IS ATTEMPTED.'//
     1)
      RETURN 1
C----------ERROR: NO DATA FOUND.
   14 WRITE(KOUT,608)
  608 FORMAT(//5X,'KEINE DATEN IM ANGEGEBENEN INTERVALL'//)
      RETURN 1
   40 DO 42 I=1,N2
      IF(X(I).LE.FT) GOTO 42
      N2=I-1
      GOTO 44
   42 CONTINUE
   44 DO 46 I=1,N2
      IF(X(I).LE.FF) GOTO 46
      JI=I-1
      GOTO 48
   46 CONTINUE
   48 IF(JI.EQ.0) GOTO 50
      N2=N2-JI
      CALL STRING(X(1),X(JI+1),4*N2)
   50 N1=N2
      WRITE(KOUT,602) ANZ
```

```
  602 FORMAT(//5X,'NUMBER OF DATAPOINTS TRANSMITTED=',I4)                  810
      RETURN                                                               820
      END                                                                  830
      SUBROUTINE SKALA(XMIN,XMIL,XMAL,SX,YMIL,SY,NSKA,NPLOS,LASKA,          10
     1XSKA,YSKA,BSK,RAST,SLOG,KONTR)                                       20
C----------PROVIDE SCALING FOR PLOTS.                                      30
      DIMENSION XSKA(2),YSKA(2),BSK(2),DX(2),DY(2),DSKA(9),DXL(2),DYL(2)   40
     1,DYZW(2)                                                             50
      REAL*8 POINT1/'. '/,POINT2/'     .. '/,POINT3/'  ..  '/,              60
     1BLANK/'        '/,BSK,ZERO/'0.     .'/,BZERO/'   0.  .'/             70
      DATA DX/30.,8./,DY/24.,56./,DXL/12.,8./,DYL/28.,56./,DYZW/12.,36./   80
      DATA DSKA/.0458,.3010,.1761,.1250,.0969,.0792,.0669,.0580,.0511/      90
      LOGICAL*1 BV(10),BW(7),PLUS/'+'/,PT/'.'/,MINUS/'-'/                  100
      INTEGER*2 TST,TSTB/'E '/                                             110
      LOGICAL *1 BL/' '/                                                   120
      INTEGER*2 INTNUM(11)/' 0',' 1',' 2',' 3',' 4',' 5',' 6',' 7',' 8',   130
     1' 9',' 0'/,                                                          140
     2NUMEX,NUMAX,ST/'**'/                                                 150
      EQUIVALENCE (NUMEX,BV(9)),(NUMAX,BV(7))                              160
      IF(RAST.EQ.SLOG) GOTO 3                                              170
C----------SCALE FOR LIN. AXIS.                                            180
      XSKA(1)=XMIL                                                         190
      NSKA=0                                                               200
    1 NSKA=NSKA+1                                                          210
      IF(NSKA.GE.100) GOTO 11                                              220
      XSKA(NSKA+1)=XSKA(NSKA)+80.*SX                                       230
      IF(XSKA(NSKA+1).LE.XMAL) GOTO 1                                      240
   11 NPLOS=NSKA/10                                                        250
      LASKA=NSKA-NPLOS*10                                                  260
      DO 2 I=1,NSKA                                                        270
      J=2*I-1                                                              280
      BSK(J)=BLANK                                                         290
      BSK(J+1)=POINT1                                                      300
      CALL CONVX(XSKA(I),BV(1),5HE10.3)                                    310
      BV(1)=BV(4)                                                          320
      BV(2)=BV(3)                                                          330
      DO 18 K=3,6                                                          340
   18 BV(K)=BV(K+2)                                                        350
      CALL STRING(TST,BV(7),2)                                             360
      IF(TST.EQ.TSTB) BV(6)=PLUS                                           370
      BV(9)=BL                                                             380
      IF(TST.EQ.TSTB) GOTO 21                                              390
      DO 19 K=1,10                                                         400
      IF(NUMEX.NE.INTNUM(K)) GOTO 19                                       410
      NUMEX=INTNUM(K+1)                                                    420
      GOTO 20                                                              430
   19 CONTINUE                                                             440
   24 NUMEX=ST                                                             450
      GOTO 20                                                              460
   21 DO 22 K=1,9                                                          470
      IF(NUMEX.NE.INTNUM(K)) GOTO 22                                       480
      IF(K.EQ.1) GOTO 23                                                   490
      NUMEX=INTNUM(K-1)                                                    500
      GOTO 20                                                              510
   23 NUMEX=INTNUM(2)                                                      520
      BV(6)=MINUS                                                          530
      GOTO 20                                                              540
   22 CONTINUE                                                             550
      GOTO 24                                                              560
   20 BV(7)=BV(10)                                                         570
      BV(8)=PT                                                             580
      CALL STRING(BSK(J),BV(1),8)                                          590
      IF(BSK(J).EQ.ZERO) BSK(J)=BZERO                                      600
      XSKA(I)=XSKA(I)-DX(KONTR)*SX                                         610
    2 YSKA(I)=YMIL-DY(KONTR)*SY                                            620
      GOTO 10                                                              630
C----------SCALE IS LOG.SCALING DETERMINED BY LENGTH OF DECADE.            640
    3 XSKA(1)=XMIL-DXL(KONTR)*SX                                           650
      YSKA(1)=YMIL-DYL(KONTR)*SY                                           660
      NSKA=0                                                               670
    4 NSKA=NSKA+1                                                          680
      IF(NSKA.GE.100) GOTO 41                                              690
      XSKA(NSKA+1)=XSKA(NSKA)+DSKA(MOD(NSKA,9)+1)                          700
      YSKA(NSKA+1)=YSKA(1)                                                 710
      IF(XSKA(NSKA+1).LE.XMAL) GOTO 4                                      720
   41 NPLOS=NSKA/10                                                        730
      LASKA=NSKA-NPLOS*10                                                  740
      JEXP=0                                                               750
      DO 6 I=1,NSKA,9                                                      760
      J=2*I-1                                                              770
      BSK(J)=POINT2                                                        780
      BSK(J+1)=BLANK                                                       790
      CALL CONVX(XMIN*(10.**JEXP),BV(1),4HE8.1)                            800
      BV(1)=BV(4)                                                          810
      BV(2)=BV(3)                                                          820
      BV(3)=BV(5)                                                          830
      BV(4)=BV(6)                                                          840
      CALL STRING(TST,BV(5),2)                                             850
      IF(TST.EQ.TSTB) BV(4)=PLUS                                           860
      BV(7)=BL                                                             870
      IF(TST.EQ.TSTB) GOTO 45                                              880
      DO 43 K=1,10                                                         890
      IF(NUMAX.NE.INTNUM(K)) GOTO 43                                       900
      NUMAX=INTNUM(K+1)                                                    910
      GOTO 44                                                              920
   43 CONTINUE                                                             930
   48 NUMAX=ST                                                             940
      GOTO 44                                                              950
   45 DO 46 K=1,9                                                          960
      IF(NUMAX.NE.INTNUM(K)) GOTO 46                                       970
      IF(K.EQ.1) GOTO 47                                                   980
      NUMAX=INTNUM(K-1)                                                    990
      GOTO 44                                                             1000
   47 NUMAX=INTNUM(2)                                                     1010
      BV(4)=MINUS                                                         1020
      GOTO 44                                                             1030
```

```
   46 CONTINUE                                                          1040
      GOTO 48                                                           1050
   44 BV(5)=BV(8)                                                       1060
      CALL STRING(BSK(J),BV(1),5)                                       1070
      DO 5 N=1,8                                                        1080
      BSK(J+2*N)=POINT3                                                 1090
      BSK(J+2*N+1)=BLANK                                                1100
      YSKA(I+N)=YSKA(I+N)+DYZW(KONTR)*SY                                1110
    5 CALL CONVX(N+1,BSK(J+2*N),2HI2)                                   1120
      JEXP=JEXP+1                                                       1130
    6 CONTINUE                                                          1140
      GOTO 10                                                           1150
   10 RETURN                                                            1160
      END                                                               1170
```

```
      SUBROUTINE KEDSUB(NR,NAMZ,NAME ,EUNT,EOB,XINTPO,XINT,ZXINT,IFMOUT,    10
     1*)                                                                    20
C----------READ KEDAK DATA.                                               30
      REAL*8 NAME(1),NAMEN(3)                                           40
      COMMON/INOUT/KOUT                                                 50
      INTEGER ANZ,ZXINT                                                 60
      LOGICAL XINTPO                                                    70
      DIMENSION NARG(3),ARG(2),XP(1000),YP(1000),XINT(ZXINT)            80
      COMMON XP,YP,ANZ                                                  90
      DO 1 I=1,NAMZ                                                     100
    1 NAMEN(I)=NAME(I)                                                  110
      NR=0                                                              120
      ANZ=0                                                             130
      NARG(1)=NAMZ                                                      140
C----------LOCATE DATA.                                                150
      CALL LOFLOC(NR,NARG,NAMEN,ARG)                                    160
      IF(NR.EQ.0) GOTO 5                                                170
    3 IF(ARG(1).LT.EUNT) GOTO 2                                         180
      IF(ARG(1).GT.EOB) GOTO 4                                          190
      ANZ=ANZ+1                                                         200
      XP(ANZ)=ARG(1)                                                    210
      YP(ANZ)=ARG(2)                                                    220
      IF(ARG(1).EQ.EOB) GOTO 4                                          230
      IF(ANZ.LT.1000) GOTO 2                                            240
      WRITE(KOUT,604)                                                   250
      GOTO 4                                                            260
C----------LOOP  TO SEARCH FOR ENERGY INTERVAL.                        270
    2 CALL LOFNXT(NR,NARG,NAMEN,ARG)                                    280
      IF(NR.EQ.0) GOTO 6                                                290
      GOTO 3                                                            300
    4 WRITE(KOUT,603) ANZ                                               310
      IF(XINTPO) GOTO 10                                                320
      RETURN                                                            330
    5 WRITE(KOUT,601) NAMZ,(NAME(I),I=1,2)                              340
      IF(NAMZ.GE.3) WRITE(KOUT,606) NAME(3)                             350
      WRITE(KOUT,602)                                                   360
      RETURN 1                                                          370
```

```
    6 WRITE(KOUT,601) NAMZ,(NAME(I),I=1,2)                              380
      IF(NAMZ.GE.3) WRITE(KOUT,606) NAME(3)                             390
      WRITE(KOUT,608) ANZ                                               400
      RETURN                                                            410
C----------INTERPOLATION ENERGIES WERE REQUESTED ALSO.                  420
   10 IF(IFMOUT.EQ.1) GOTO 12                                           430
    9 IFMOUT=1                                                          440
      ZXINT=ANZ                                                         450
      DO 11 I=1,ZXINT                                                   460
   11 XINT(I)=XP(I)                                                     470
      RETURN                                                            480
   12 WRITE(KOUT,605)                                                   490
      GOTO 9                                                            500
  601 FORMAT(//5X,'MESSAGE FROM DATARETRIEVAL FROM KEDAK FOR:'/         510
     112X,'NUMBER OF NAMES=',I2/                                        520
     1 12X,'NAMES=',A8,2X,A8)                                           530
  602 FORMAT(8X,'NO ENTRY ON KEDAK FOR THIS DATA TYPE'//)               540
  603 FORMAT(5X,'ANZAHL DER VON KEDAK GEHOLTEN DATENPUNKTE = ',I5)      550
  604 FORMAT(//5X,'NACHRICHT'/8X,'BEIM KEDAKAUFRUF WURDEN 1000 DATENPUNK 560
     1TE EINGELESEN.DAMIT IST DAS ZUR VERFUEGUNG STEHENDE FELD GEFUELLT' 570
     2/5X,'AKTION:KEDAK EINLESEN WIRD ABGEBROCHEN.BISHER ERHALTENE WERTE 580
     3 WERDEN VERARBEITET'//)                                           590
  605 FORMAT(//5X,'NACHRICHT'/8X,'DIE UEBER &OUTINT EINGELESENEN INTERPO 600
     1LATIONSPUNKTE VERLIEREN IHRE GUELTIGKEIT.SIE WERDEN DURCH DIE KEDA 610
     2KENERGIEN ERSETZT')                                               620
  606 FORMAT('+',36X,E14.6)                                             630
  607 FORMAT(8X,'EO OR EU > HIGHEST ENERGY FOR THIS TYPE')              640
  608 FORMAT(8X,'NUMBER OF DATAPOINTS FOUND =',I5//)                    650
      END                                                               660
```

```
      SUBROUTINE DRUCK3(IM,X,Y,P)                                       10
C----------PRINT DATA POINTS PLUS ERROR BARS.                           20
C----------60 LINES PER PAGE, 3 TIMES 3 COLUMNS.                        30
      DIMENSION X(IM),Y(IM),P(IM)                                       40
      COMMON/INOUT/KOUT                                                 50
      NUMSPA=3                                                          60
      GOTO 10                                                           70
      ENTRY DRUCKX(X,Y,IM)                                              80
      NUMSPA=4                                                          90
   10 JM=IM/NUMSPA                                                      100
      JMS=JM*NUMSPA                                                     110
      IF(JMS.LT.IM) JM=JM+1                                             120
      IMS=0                                                             130
    1 IF(JM.GT.60) GOTO 3                                               140
      JAN=IMS+1                                                         150
      JMS=IMS+JM                                                        160
      IMS=IM                                                            170
      IF(NUMSPA.EQ.3) GOTO 31                                           180
      DO 2 I=JAN,JMS                                                    190
    2 WRITE(KOUT,601) (X(J),Y(J),J=I,IMS,JM)                            200
      RETURN                                                            210
   31 DO 32 I=JAN,JMS                                                   220
```

```
   32 WRITE(KOUT,602) (X(J),Y(J),P(J),J=I,IMS,JM)
      RETURN
    3 JAN=IMS+1
      JMS=IMS+60
      IMS=IMS+60*NUMSPA
      IF(NUMSPA.EQ.3) GOTO 33
      DO 4 I=JAN,JMS
    4 WRITE(KOUT,601) (X(J),Y(J),J=I,IMS,60)
    5 JM=JM-60
      WRITE(KOUT,600)
      GOTO 1
   33 DO 34 I=JAN,JMS
   34 WRITE(KOUT,602) (X(J),Y(J),P(J),J=I,IMS,60)
      GOTO 5
  600 FORMAT(1H1)
  601 FORMAT (4(5X,2(1PE13.5)))
  602 FORMAT (3(5X,3(1PE13.5)))
      END
```

```
      SUBROUTINE ADF(XL,YL,Z6,NADD,NANZ,ANZ,XP,YP,DELTA,DATADD,*)
C----------APPEND DATA TO EXISTING SET.
      COMMON/INOUT/KOUT
      DIMENSION XL(1),YL(1),Z6(1),XP(1),YP(1),DELTA(1)
      LOGICAL DATADD
      INTEGER ANZ
      IF(NADD.NE.1) GOTO 1290
      IF(ANZ+NANZ.LE.1000) GOTO 129
      WRITE(KOUT,649)
      NANZ=1000-ANZ
  129 CALL STRING(XP(ANZ+1),XL(1),4*NANZ)
      CALL STRING(YP(ANZ+1),YL(1),4*NANZ)
      CALL STRING(DELTA(ANZ+1),Z6(1),4*NANZ)
      ANZ=ANZ+NANZ
      NADD=0
 1290 IF(.NOT.DATADD) RETURN
      NADD=1
      DO 1291 J=1,ANZ
      XL(J)=XP(J)
      YL(J)=YP(J)
 1291 Z6(J)=DELTA(J)
      NANZ=ANZ
      RETURN 1
  649 FORMAT(//'  ADD=T WAR ANGEGEBEN.GESAMTZAHL DER PUNKTE WIRD DADURCH
     1 > 1000.'/'      ERSTER DATENSATZ WIRD GEKUERZT'//)
      END
```

```
      SUBROUTINE ASSIGN(S,T,J,NR,M)
C----------COMPARE HOLLERITH DATA.
      DIMENSION S(2),T(2,M)
      NR=0
      DO 1 I=1,2
      IF(S(I).NE.T(I,J)) GOTO 2
    1 CONTINUE
      NR=1
    2 RETURN
      END
```

```
      SUBROUTINE BDVGL(B,T,*)
C----------COMPARE NEUDADA ID.
      DIMENSION B(54),T(6)
      DO 1 J=1,6
      JB=41+J
      IF(B(JB).NE.T(J)) GOTO 2
    1 CONTINUE
      RETURN
    2 RETURN1
      END
```

```
      SUBROUTINE ACONVE(ACCNR,BRR,ACONV)
C----------CONVERT INTEGER ID TO HOLLERITH.
      DIMENSION BRR(6),ACONV(10),ID(5)
      INTEGER ACCNR
      IZ=ACCNR
      ID(1)=1
      DO 1 K=2,5
    1 ID(K)=ID(K-1)*10
      DO 2 I=1,5
      K=6-I
      JZ=IZ/ID(K)
      BRR(I+1)=ACONV(JZ+1)
    2 IZ=IZ-JZ*ID(K)
      RETURN
      END
```

```
      SUBROUTINE XCORR
C----------PLOTA DOES NOT ALLOW PLOTTING OF A SINGLE DATA POINT.
C----------HERE A POINT IS ADDED FOR PLOT ONLY WHICH IS REMOVED LATER.
      DIMENSION XP(1000),YP(1000)
      COMMON XP,YP,IM
      COMMON/PAR/XMAX,SX,YMAX,SY
      XP(2)=XP(1)
      YP(2)=YP(1)
```

```
      IM=2                                                              90
      RETURN                                                           100
      END                                                              110
```

```
      SUBROUTINE SUB1(INDZ,RAST,NEIN,XLOG,YLOG,KREX,KREY,*)             10
C----------SET FLAGS FOR PLOT SCALES.                                   20
      COMMON/CLP/CALP                                                  30
      LOGICAL CALP                                                     40
      REAL NEIN                                                        50
      DIMENSION RAST(2)                                                60
      IF(INDZ.EQ.0.OR.INDZ.EQ.10) RETURN1                              70
      INDZ=0                                                           80
      IF(CALP) INDZ=10                                                 90
      IF(RAST(1).EQ.NEIN) RETURN1                                     100
      KREX=1                                                          110
      IF(RAST(1).EQ.XLOG) KREX=-1                                     120
      KREY=1                                                          130
      IF(RAST(2).EQ.YLOG) KREY=-1                                     140
      RETURN                                                          150
      END                                                             160
```

```
      SUBROUTINE QFISS(MAX,T,S,F)                                       10
C----------CALCULATE FIISION SPECTRUM AVERAGE.                          20
C----------AN ANALYTICAL FISSION SPECTRUM IS USED BY DEFAULT.           30
C----------BUT BY REPLACING CHIF THE USER MAY SUPPLY HIS OWN            40
C----------SPECTRUM.                                                    50
      DIMENSION T(MAX),S(MAX),F(MAX)                                    60
      COMMON/INOUT/KOUT                                                 70
      CHIF(E)=.4527*EXP(-E/.965E+6)*SINH(SQRT(2.29E-6*E))               80
      VSUM=0.                                                           90
      SUM=0.                                                           100
      SUMCHI=0.                                                        110
      WRITE(KOUT,600)                                                  120
  600 FORMAT(//5X,' ES BEDEUTEN IM NACHFOLGENDEN AUSDRUCK:'/           130
     A10X,' E=ENERGIE'/                                                140
     110X,' QF=MIT SPALTSPEKTRUM GEMITTELTER QUERSCHNITT ZWISCHEN DIESER 150
     2 ENERGIE UND DER VORHERGEHENDEN'//                               160
     36X,'E',11X,'QF'/                                                 170
     45X,'0.')                                                         180
      IF(T(1).LE.0.) GOTO 9                                            190
      DT=T(1)/100.                                                     200
      X1=0.                                                            210
      X2=DT                                                            220
      DO 5 I=1,100                                                     230
      SUMCHI=SUMCHI+DT*(CHIF(X2)+CHIF(X1))                             240
      X1=X2                                                            250
    5 X2=X1+DT                                                         260
    9 F(1)=S(1)*CHIF(T(1))                                             270
      DO 10 I=2,MAX                                                    280
      F(I)=S(I)*CHIF(T(I))                                             290
      SUMCHI=SUMCHI+(CHIF(T(I))+CHIF(T(I-1)))*(T(I)-T(I-1))            300
         SUM=SUM+(F(I)+F(I-1))*(T(I)-T(I-1))                           310
      IF((I/16)*16.NE.I) GOTO 10                                       320
      DSUM=(SUM-VSUM)*.5E-3                                            330
      VSUM=SUM                                                         340
      CHIE=CHIF(T(I))                                                  350
      WRITE(KOUT,601) T(I),DSUM,CHIE                                   360
   10 CONTINUE                                                         370
  601 FORMAT(1X,E12.3,1X,E12.3,'  CHIF=',E10.3)                        380
      SUM=SUM*.5E-3                                                    390
      WRITE(KOUT,602) SUM                                              400
  602 FORMAT(//'  QFISS(CHIF(U235))=',E12.4,'MILLIBARN')               410
      SUMCHI=SUMCHI*.5E-6                                              420
      WRITE(KOUT,603) SUMCHI                                           430
  603 FORMAT(//' SUMCHI=',5X,F12.6)                                    440
      RETURN                                                           450
      END                                                              460
```

```
      SUBROUTINE RASTER(KREX,KREY,XMAX,XMIN,SX,YMAX,YMIN,SY,INDZ,X,Y)   10
C----------RASTER CONTROLS PLOTTING OF SCALES.                          20
      DIMENSION X(2),Y(2)                                              30
      DATA NRAST/9999/                                                 40
      IF(INDZ.NE.0.AND.INDZ.NE.10) GOTO 50                             50
      IF(KREY.EQ.-1) GOTO 10                                           60
      CALL RALIN(YMIN,YMAX,SY,XMIN,XMAX,M,X,Y)                         70
      GOTO 20                                                          80
   10 CALL RALOG(YMIN,YMAX,SY,XMIN,XMAX,M,X,Y)                         90
   20 CALL PLOTA(X,Y,M,2,0,1,1,1,INDZ,XMAX,XMIN,SX,YMAX,YMIN,SY,      100
     11,NRAST,0,0,0)                                                  110
      IF(KREX.EQ.-1) GOTO 30                                          120
      CALL RALIN(XMIN,XMAX,SX,YMIN,YMAX,M,Y,X)                        130
      GOTO 40                                                         140
   30 CALL RALOG(XMIN,XMAX,SX,YMIN,YMAX,M,Y,X)                        150
   40 CALL PLOTA(X,Y,M,2,0,1,1,1,INDZ,XMAX,XMIN,SX,YMAX,YMIN,SY,      160
     11,NRAST,0,0,0)                                                  170
   50 RETURN                                                          180
      END                                                             190
```

```
      SUBROUTINE CALSKA(BSK,XSKA,YSKA,NPLOS,LASKA)                      10
      DIMENSION XSKA(1),YSKA(1)                                         20
      REAL*8 BSK(1)                                                     30
C CALSKA BEWIRKT,DASS NUR JEDE ZWEITE SKALENTEILUNG BESCHRIFTET WIRD.   40
C                                                                       50
      M=10*NPLOS+LASKA                                                  60
      L=1                                                               70
      DO 10 I=1,M,2                                                     80
      BSK(I)=BSK(2*I-1)                                                 90
      BSK(I+1)=BSK(2*I)                                                100
```

```
      XSKA(L)=XSKA(I)
      YSKA(L)=YSKA(I)
   10 L=L+1
      M=L-1
      NPLOS=M/10
      LASKA=M-10*NPLOS
      RETURN
      END
```

```
      SUBROUTINE CONLOG(X,XL,MAX,RAST,SLOG)
C----------CONVERT DATA TO LOG, IF REQUIRED.
      DIMENSION X(2),XL(2)
      IF(RAST.NE.SLOG) GOTO 10
      DO 5 M=1,MAX
    5 XL(M)=ALOG10(X(M))
      GOTO 20
   10 CALL STRING(XL(1),X(1),4*MAX)
   20 RETURN
      END
```

```
      SUBROUTINE NEUDEN(KB,BRR,FROM,TO,X,NX,FODATA,*)
C----------FETCH INTERPOLATION POINTS FROM NEUDAD.
C----------SAME TECHNIQUE IS USED AS IN NEUDA-ROUTINE.
C----------ERRSET IS USED TO HANDLE BAD TAPES.
      DIMENSION X(2),BAND(54),BRR(6)
      LOGICAL*4 FODATA
      COMMON/INOUT/KOUT
      CALL ERRSET(214,200,-10,1,1)
      NR=0
      NX=0
   10 IF(FODATA) READ(KB,100,END=80,ERR=90)  (BAND(L),L=1,54)
      IF(.NOT.FODATA) READ(KB,120,END=80,ERR=90) (BAND(L),L=1,54)
  100 FORMAT(47A1,7E12.5,1X)
  120 FORMAT(47A1,7A4)
      CALL BDVGL(BAND,BRR,&10)
      WRITE(KOUT,101) (BAND(I),I=1,31)
  101 FORMAT(/'  GEFUNDEN:',31A1)
C----------CONTROL SCALING OF PLOTS.
   15 XX=BAND(48)
      IF(XX.LT.FROM) GOTO 20
      IF(XX.GT.TO) GOTO 93
      GOTO 25
   20 IF(FODATA) READ(KB,100,END=81,ERR=91) (BAND(I),I=1,54)
      IF(.NOT.FODATA) READ(KB,120,END=81,ERR=91) (BAND(I),I=1,54)
      CALL BDVGL(BAND,BRR,&94)
      GOTO 15
   25 NX=NX+1
      X(NX)=XX
   27 IF(FODATA) READ(KB,100,END=82,ERR=92) (BAND(I),I=1,54)
      IF(.NOT.FODATA) READ(KB,120,END=82,ERR=92) (BAND(I),I=1,54)
      CALL BDVGL(BAND,BRR,&70)
      XX=BAND(48)
      IF(XX.GT.TO) GOTO 72
      IF(NX.EQ.1000) GOTO 74
      GOTO 25
   70 WRITE(KOUT,102) NX
  102 FORMAT('  ENDE DIESES DATENSATZES.ES WURDEN ',I5,' DATENPUNKTE UEB
     1ERTRAGEN'/)
      RETURN
   72 WRITE(KOUT,103) NX
  103 FORMAT('  IM ANGEGEBENEN ENERGIEBEREICH WURDEN ',I5,' DATENPUNKTE
     1GEFUNDEN'/)
      RETURN
   74 WRITE(KOUT,104) NX
  104 FORMAT('  ERLAUBTE MAXIMALZAHL VON DATEN =',I5,' UEBERTRAGEN.EINLE
     1SEN WIRD ABGEBROCHEN'/)
      RETURN
   80 WRITE(KOUT,105) BRR,KB
  105 FORMAT(/'  FUER:',6A1,'  WURDEN KEINE DATEN AUF BAND ',I2,' GEFUND
     1EN'/)
      RETURN 1
   81 WRITE(KOUT,106) FROM,TO,BRR,KB
  106 FORMAT(/' ZWISCHEN EMIN=',E14.6,' UND EMAX=',E14.6,'  WURDEN FUER
     1',6A1,' KEINE DATEN AUF BAND ',I2,' GEFUNDEN'/)
      RETURN 1
   82 WRITE(KOUT,102) NX
      RETURN
   90 WRITE(KOUT,107)
  107 FORMAT(/'  *** ERROR ON TAPE. RECORDS HAD TO BE SKIPPED.'/)
      READ(KB)
      GOTO 10
   91 WRITE(KOUT,107)
      READ(KB)
      GOTO 20
   92 WRITE(KOUT,107)
      WRITE(KOUT,108)
  108 FORMAT('    UNTER DEN AUSGELASSENEN RECORDS BEFINDEN SICH WAHRSCHE
     1INLICH AUCH RECCRDS,DIE DEM GESUCHTEN DATENSATZ ANGEHOEREN'/)
      GOTO 27
   93 WRITE(KOUT,106) FROM,TO,BRR,KB
      RETURN 1
   94 WRITE(KOUT,109) BRR,KB,FROM
  109 FORMAT(/'  FUER ',6A1,'  WURDEN AUF BAND ',I2,' KEINE DATENPUNKTE
     1OBERHALB EMIN=',E14.6,'EV GEFUNDEN'/)
      RETURN 1
      END
```

```
      SUBROUTINE LINEAR(IM,XP,YP,M,T,S,NR)
C----------LINEAR INTERPOLATION.
      DIMENSION XP(IM),YP(IM),T(M),S(M)
      COMMON/INOUT/KOUT
      NR=0
      IKU=1
      IF(IM-1) 1701,1701,1702
 1701 NR=1
      GOTO 1703
 1702 DO 1704 K=1,M
      DO 1705 I=IKU,IM
      IF(XP(I)-T(K)) 1705,1705,1706
 1706 IK=I
      GOTO 1707
 1705 CONTINUE
      IF(T(K).NE.XP(IM)) GOTO 1710
      IK=IM
      GOTO 1707
 1707 IF(IK-1) 1708,1708,1709
 1708 NR=2
      WRITE(KOUT,600) K,T(K),IK,XP(IK)
  600 FORMAT(//5X,'DER ',I5,'. INTERPOLATIONSPUNKT ',E13.6,' LIEGT LINKS
     X VOM ',I3,'. DATENPUNKT: ',E13.6/
     18X,'AKTION: DIESER INTERPOLATIONSPUNKT WIRD UEBERGANGEN'//)
      GOTO 1704
 1709 D=(YP(IK)-YP(IK-1))/(XP(IK)-XP(IK-1))
      S(K)=D*(T(K)-XP(IK))+YP(IK)
      IKU=IK
 1704 CONTINUE
 1703 RETURN
 1710 WRITE(KOUT,601) K
  601 FORMAT(//5X,'AB DEM ',I5,'. INTERPOLATIONSPUNKT LIEGEN DIE INTERPO
     XLATIONSPUNKTE RECHTS VOM LETZTEN DATENPUNKT'/
     18X,'AKTION: DIESE INTERPOLATIONSPUNKTE WERDEN UEBERGANGEN'//)
      GOTO 1703
      END
```

```
      SUBROUTINE PREPLO
C----------INITIALIZE PLOT  ACCORDING CONTROL INPUT FOR
C----------PLOT SIZE.
      INTEGER PLEH,ZXINT,WNR,ZAHL,PLOZEI
      COMMON/INOUT/KOUT,KIN
      LOGICAL DRUCK,STANZ,PLO,PLOKU,DRUKU,STAKU,PLOSM,DRUSM,STASM,SMGEW,
     1 NEUZEI,XINTPO,PLOFQU,FISS,SEQN,ADD,CALP,CALPJ,DATADD,TST,REWD
      REAL NEIN,KURVE,MEDIUM,MEV,NULL,LTR,NEW
      COMMON/CLP/CALP
      COMMON/PARM1/ MAXOPT,PLEH,PLOZEI,ZXINT,IDPLOT,CALPJ
      COMMON/PARM2/ IPLOGR,ISTAFO,ISMO,IFMOUT,ISMNP,DISP,EINHTX,EINHTY,
     1 EINHTD,XEINHT,YEINHT,DEINHT,LTR,KARTEH,FMTOUT(20),STAFMT(20),
     2 FMT(20),DRUCK,STANZ,PLO,KURVE,PLOKU,DRUKU,STAKU,PLOSM,DRUSM,
     3 STASM,SMGEW,NEUZEI,XINTPO,PLOFQU,FISS,SEQN,ADD,DATADD,TST,
     4 REWD,NANZ,NADD,ZAHL,MEDIUM,WNR,
      5 RAST(2),ISMAX,FQU,NAMZ
      COMMON/PAR/XMAX,SX,YMAX,SY,XMIN,YMIN,XMAL,XMIL,YMAL,YMIL,NLGGX,
     1 NLGGY,NX,NY,INDZ,JNDZ,XMILJ,YMILJ,JP
      DATA XLOG/'XLOG'/,YLOG/'YLOG'/,NEIN/'NEIN'/,XSTA/-1.E-60/
      DATA XMINO,XMAXO,YMINO,YMAXO/4*0./
      NAMELIST/PLOTGR/XMIN,XMAX,XLG,YMIN,YMAX,YLG,RAST,EINHTX,EINHTY,
     1CALP
      COMMON/PARM4/FROM,TO
      XMIN=XSTA
C----------TEST WHETHER DATA FOR PLOT LIMITS WERE ACTUALLY IN INPUT.
C----------IF YES, PERFORM CONVERSION TO INTERNAL PHYSICAL UNITS.
      XMAX=XSTA
      YMIN=XSTA
      YMAX=XSTA
      READ(KIN,PLOTGR)
      IF(XMIN.EQ.XSTA) GOTO 1410
      CALL ECONV1(XMIN,EINHTX)
      XMINO=XMIN
      GOTO 1415
 1410 XMIN=XMINO
 1415 IF(XMAX.EQ.XSTA) GOTO 1420
      CALL ECONV1(XMAX,EINHTX)
      XMAXO=XMAX
      GOTO 1425
 1420 XMAX=XMAXO
 1425 IF(YMIN.EQ.XSTA) GOTO 1430
      CALL ECONV1(YMIN,EINHTY)
      YMINO=YMIN
      GOTO 1435
 1430 YMIN=YMINO
 1435 IF(YMAX.EQ.XSTA) GOTO 1440
      CALL ECONV1(YMAX,EINHTY)
      YMAXO=YMAX
      GOTO 1450
 1440 YMAX=YMAXO
C----------INCREMENT PLOTID AND SET PLOT SIGN SELECTION FLAG.
 1450 ISMNP=0
      IDPLOT=IDPLOT+1
      PLOZEI=-1
      JP=0
      IF(RAST(1).EQ.NEIN) RAST(2)=NEIN
C----------CONFIRM INPUT.
      WRITE(KOUT,636)
      WRITE(KOUT,615) XMIN,XMAX,XLG,YMIN,YMAX,YLG,RAST,EINHTX,EINHTY,
     *CALP
      WRITE(KOUT,636)
      IPLOGR=1
       FROM=XMIN
         TO=XMAX
      SX=(XMAX-XMIN)/(40.*XLG)
      SY=(YMAX-YMIN)/(40.*YLG)
      XMAL=XMAX
      XMIL=XMIN
      YMAL=YMAX
      YMIL=YMIN
```

```
      NLGGX=+1                                                          710
      NLGGY=+1                                                          720
      NX=+1                                                             730
      NY=+1                                                             740
C----------LOG. SCALE.                                                 750
      IF(RAST(1).NE.XLOG) GOTO 146                                      760
      XMAL=ALOG10(XMAX)                                                 770
      XMIL=ALOG10(XMIN)                                                 780
      SX=(XMAL-XMIL)/(40.*XLG)                                          790
      NLGGX=-1                                                          800
      IF(XMIL.LT.0.) NY=0                                               810
  146 IF(RAST(2).NE.YLOG) GOTO 147                                      820
      YMAL=ALOG10(YMAX)                                                 830
      YMIL=ALOG10(YMIN)                                                 840
      SY=(YMAL-YMIL)/(40.*YLG)                                          850
      NLGGY=-1                                                          860
      IF(YMIL.LT.0.) NX=0                                               870
C----------ESTABLISH PLOT SIZE INDICATOR FOR PLOTA.                     880
  147 F=XLG/25.                                                         890
      IF(CALP) F=XLG/60.                                                900
      IF(F.LT.4.) INDZ=INT(F)+1                                         910
      IF(F.EQ.4.) INDZ=4                                                920
      IF(F.GT.4.) GOTO 806                                              930
      IF(.NOT.CALP) GOTO 148                                            940
      INDZ=INDZ+10                                                      950
      CALPJ=.TRUE.                                                      960
      XMILJ=XMIL                                                        970
      YMILJ=YMIL                                                        980
  148 JNDZ=INDZ                                                         990
      RETURN                                                           1000
  806 WRITE(KOUT,609) F                                                1010
      INDZ=4                                                           1020
      IF(CALP) INDZ=INDZ+10                                            1030
      JNDZ=INDZ                                                        1040
      RETURN                                                           1050
  609 FORMAT(//5X,'FEHLERNACHRICHT'/8X,'F=XLG/YLG>4 NICHT ERLAUBT:F =',E 1060
     112.5/5X,'AKTION:PLOTFORMAT - F=4 GESETZT'//)                     1070
  615 FORMAT(' &PLOTGR XMIN=',E12.3,',  XMAX=',E12.3,',  XLG=',E12.3/9X, 1080
     1'YMIN=',E12.3,',  YMAX=',E12.3,',  YLG=',E12.3/8X,' RAST=',A4,',', 1090
     2A4/9X,'EINHTX=',A4,', EINHTY=',A4,', CALP=',L2)                  1100
  636 FORMAT(2X,130('-'))                                              1110
      END                                                              1120
```

```
      SUBROUTINE CDINPT(A,TEND)                                          10
C----------COPY CONTROL INPUT TO FT09 TO ENABLE BACKSPACING.             20
      DIMENSION A(20),TEND(20)                                           30
      COMMON/INOUT/KOUT,KIN,DUMMY,KINC                                  40
      DATA TTEND/'ENDE'/,BLANK/'    '/                                   50
      TEND(1)=TTEND                                                      60
      DO 100 I=2,20                                                      70
  100 TEND(I)=BLANK                                                      80
      WRITE(KOUT,600)                                                    90
  600 FORMAT('1',10X,'NACHRICHT DER SUBROUTINE *CDINPT*'/11X,'CDINPT LEG 100
     1T DIE KARTENEINGABE AUF G.FT09'/                                  110
     211X,'DIE MAXIMALZAHL VON EINGABEKARTEN IST DURCH DEN SPACEPARAMETE 120
     3R FUER FT09 GEGEBEN')                                             130
      N=1                                                               140
 1000 READ(KINC,500,END=99,ERR=9) A                                     150
      WRITE(KIN,500) A                                                  160
  500 FORMAT(20A4)                                                      170
      N=N+1                                                             180
      GOTO 1000                                                         190
    9 WRITE(KOUT,601) N                                                 200
  601 FORMAT(/' FEHLER BEIM LESEN DER',I5,'-TEN KARTE'/)                210
      GOTO 1000                                                         220
   99 WRITE(KIN,500) TEND                                               230
      REWIND KIN                                                        240
      WRITE(KOUT,602) N                                                 250
  602 FORMAT(/' DER DATASET AUF G.FT09 BESTEHT AUS',I5,'  RECORDS.'/5X,' 260
     1ALS LETZTES RECORD IST EIN *ENDE*-RECORD HINZUGEFUEGT WORDEN')     270
      RETURN                                                            280
      END                                                               290
```

```
      SUBROUTINE ORDNEN(JMAX,X,Y,Z,U,V,W)                                10
C----------SORT DATA. ROUTINE IS OPTIMIZED ASSUMING THAT DATA             20
C----------WILL BE SORTED USUALLY.                                        30
      DIMENSION X(JMAX),Y(JMAX),Z(JMAX),U(JMAX),V(JMAX),W(JMAX)          40
      U(1)=X(1)                                                          50
      V(1)=Y(1)                                                          60
      W(1)=Z(1)                                                          70
      IF(JMAX.LT.2) GOTO 20                                              80
      DO 10 J=2,JMAX                                                     90
      M=J-1                                                             100
    1 IF(X(J).LT.U(M)) GOTO 2                                           110
      IF(J.NE.(M+1)) GOTO 3                                             120
      U(J)=X(J)                                                         130
      V(J)=Y(J)                                                         140
      W(J)=Z(J)                                                         150
      GOTO 10                                                           160
    2 M=M-1                                                             170
      IF(M.NE.0) GOTO 1                                                 180
    3 IA=M+1                                                            190
      IE=J-1                                                            200
      DO 4 K=IA,IE                                                      210
      I=IA+IE-K                                                         220
      U(I+1)=U(I)                                                       230
      V(I+1)=V(I)                                                       240
    4 W(I+1)=W(I)                                                       250
      U(IA)=X(J)                                                        260
      V(IA)=Y(J)                                                        270
      W(IA)=Z(J)                                                        280
   10 CONTINUE                                                          290
   20 RETURN                                                            300
      END                                                               310
```

```
      SUBROUTINE KEDEN(NARG,NAME,EUNT,EOB,XINT,ZXINT,*)                 10
C----------RETRIEVE INTERPOLATION ENERGIES FROM KEDAK.                  20
C----------SAME TECHNIQUE IS USED AS IN ROUTINE KEDSUB.                 30
      REAL*8 NAME(1),NAMEN(3)                                           40
       COMMON/INOUT/KOUT                                                50
      INTEGER ZXINT                                                     60
      DIMENSION XINT(2),NAMZ(3),VAL(2)                                  70
      DO 10 I=1,NARG                                                    80
   10 NAMEN(I)=NAME(I)                                                  90
      NR=0                                                             100
      ZXINT=0                                                          110
      NAMZ(1)=NARG                                                     120
      CALL LDFLOC(NERR,NAMZ,NAMEN,VAL)                                 130
      IF(NERR.EQ.0)  GOTO 100                                          140
      IF(VAL(1).GT.EOB)  GOTO 105                                      150
      IF(VAL(1).LT.EUNT)  GOTO 15                                      160
      GOTO 20                                                          170
   15 CALL LDFNXT (NERR,NAMZ,NAMEN,VAL)                                180
      IF(NERR.EQ.0)  GOTO 110                                          190
      IF(VAL(1).LT.EUNT)  GOTO 15                                      200
      IF(VAL(1).GT.EOB)  GOTO 115                                      210
      GOTO 20                                                          220
   25 CALL LDFNXT (NERR,NAMZ,NAMEN,VAL)                                230
      IF(NERR.EQ.0)  GOTO 50                                           240
      IF(VAL(1).GT.EOB)  GOTO 60                                       250
      IF(ZXINT.EQ.1000) GOTO 120                                       260
   20 ZXINT=ZXINT+1                                                    270
      XINT(ZXINT)=VAL(1)                                               280
      GOTO 25                                                          290
   50 WRITE(KOUT,601) (NAME(I),I=1,2)                                  300
      IF(NARG.GE.3) WRITE(KOUT,608) NAME(3)                            310
      WRITE(KOUT,609) XINT(ZXINT),ZXINT                                320
  601 FORMAT(//5X,'MESSAGE FROM DATARETRIEVAL FROM KEDAK FOR:'/        330
     1 12X,A8,2X,A8)                                                   340
  608 FORMAT('+',32X,E14.6)                                            350
  609 FORMAT(8X,'LAST KEDAKVALUE ENCOUNTERED BEFORE SPECIFIED EOB WAS RE 360
     1ACHED'/                                                          370
     2 8X,'LAST ENERGY TRANSMITTED:',E14.6/                            380
     3 8X,'NUMBER OF ENERGIES TRANSMITTED:',I6//)                      390
      RETURN                                                           400
   60 WRITE(KOUT,601) (NAME(I),I=1,2)                                  410
      IF(NARG.GE.3) WRITE(KOUT,608) NAME(3)                            420
      WRITE(KOUT,602) XINT(ZXINT),ZXINT                                430
  602 FORMAT(8X,'LAST ENERGY IN SPECIFIED INTERVAL FOUND'/             440
     1 8X,'ITS VALUE IS =',E14.6/                                      450
     2 8X,'AND THE NUMBER OF ENERGIES TRANSMITTED IS =',I6//)          460
      RETURN                                                           470
  100 WRITE(KOUT,601) (NAME(I),I=1,2)                                  480
      IF(NARG.GE.3) WRITE(KOUT,608) NAME(3)                            490
      WRITE(KOUT,603)                                                  500
  603 FORMAT(8X,'NO ENTRY FOR THIS DATA TYPE ON KEDAK'//)              510
      RETURN 1                                                         520
  105 WRITE(KOUT,601) (NAME(I),I=1,2)                                  530
      IF(NARG.GE.3) WRITE(KOUT,608) NAME(3)                            540
      WRITE(KOUT,604)                                                  550
  604 FORMAT(8X,'FIRST ENERGY OF THIS DATA TYPE IS > SPECIFIED EUNT'//) 560
      RETURN 1                                                         570
  110 WRITE(KOUT,601) (NAME(I),I=1,2)                                  580
      IF(NARG.GE.3) WRITE(KOUT,608) NAME(3)                            590
      WRITE(KOUT,605)                                                  600
  605 FORMAT(8X,'ALL ENERGIES OF THIS DATA TYPE ARE < SPECIFIED EUNT'//) 610
      RETURN 1                                                         620
  115 WRITE(KOUT,601) (NAME(I),I=1,2)                                  630
      IF(NARG.GE.3) WRITE(KOUT,608) NAME(3)                            640
      WRITE(KOUT,606)                                                  650
  606 FORMAT(8X,'NO DATAPOINTS IN SPECIFIED ENERGY INTERVAL'//)        660
      RETURN 1                                                         670
  120 WRITE(KOUT,607) ZXINT                                            680
  607 FORMAT(//'  ERLAUBTE MAXIMALZAHL VON DATEN=',I5,' WURDE UEBERTRAGE 690
     1N.EINLESEN WIRD ABGEBROCHEN.'//)                                 700
      RETURN                                                           710
      END                                                              720
```

```
      SUBROUTINE RALOG(YMIN,YMAX,SY,XMIN,XMAX,NPOINT,X,Y)               10
C----------PROVIDE LOG. SCALE LINES.                                    20
      DIMENSION X(2),Y(2),DYL(9)                                        30
       COMMON/INOUT/KOUT                                                40
      DATA DYL/.0458,.3010,.1761,.1250,.0969,.0792,.0669,.0580,.0511/   50
      YL=YMAX-YMIN                                                      60
      NDEK=INT(YL)                                                      70
      IF((YL-FLOAT(NDEK)).GT.0.8) NDEK=NDEK+1                           80
      IF(NDEK.GT.11) GOTO 25                                            90
      IF(SY.GT..005) GOTO 25                                           100
      NLIN=9*NDEK+1                                                    110
      NPOINT=2*NLIN                                                    120
      J=0                                                              130
      Y1=YMIN                                                          140
      Y2=YMIN+DYL(2)                                                   150
      DO 20 I=1,NPOINT,4                                               160
      X(I)=XMIN                                                        170
      X(I+1)=XMAX                                                      180
      X(I+2)=XMAX                                                      190
      X(I+3)=XMIN                                                      200
      Y(I)=Y1                                                          210
      Y(I+1)=Y1                                                        220
      Y(I+2)=Y2                                                        230
      Y(I+3)=Y2                                                        240
      J=J+2                                                            250
      Y1=Y2+DYL(MOD(J,9)+1)                                            260
      Y2=Y1+DYL(MOD(J+1,9)+1)                                          270
   20 CONTINUE                                                         280
      Y(NPOINT-1)=YMAX                                                 290
      Y(NPOINT)=YMAX                                                   300
```

```
      GCTO 30                                                           310
   25 NPOINT=0                                                          320
      WRITE(KOUT,601)                                                   330
  601 FCRMAT(' ***** ERRORMESSAGE FROM RASTER - ROUTINE. . .MINIMUM LENG 340
     1TH OF A DECADE IS 5 CM.'/                                         350
     220X,'DECREASE NUMBER OF DECADES OR USE CALCOMP - PLOTTER.')       360
   30 RETURN                                                            370
      END                                                               380
```

```
      SUBROUTINE RALIN(YMIN,YMAX,SY,XMIN,XMAX,NPOINT,X,Y)                10
C----------PROVIDE LIN SCALE LINES.                                      20
      DIMENSION X(2),Y(2)                                               30
      DY=80.*SY                                                         40
      YL=YMAX-YMIN                                                      50
      NLIN=INT(YL/DY)                                                   60
      IF((YL-FLOAT(NLIN)*DY).GT.0.99*DY) NLIN=NLIN+1                    70
      NLIN=NLIN+1                                                       80
      NPOINT=2*NLIN                                                     90
      IF(NPOINT.GT.200) NPOINT=200                                     100
      Y1=YMIN                                                          110
      Y2=YMIN+DY                                                       120
      DC 10 I=1,NPOINT,4                                               130
      X(I)=XMIN                                                        140
      X(I+1)=XMAX                                                      150
      X(I+2)=XMAX                                                      160
      X(I+3)=XMIN                                                      170
      Y(I)=Y1                                                          180
      Y(I+1)=Y1                                                        190
      Y(I+2)=Y2                                                        200
      Y(I+3)=Y2                                                        210
      Y1=Y2+DY                                                         220
      Y2=Y1+DY                                                         230
   10 CONTINUE                                                         240
      Y(NPOINT)=YMAX                                                   250
      Y(NPOINT-1)=YMAX                                                 260
      RETURN                                                           270
      END                                                              280
```

```
      SUBROUTINE ECONV(M,X,E)                                           10
C----------CONVERT DATA TO INTERNAL PHYSICAL UNITS (EV USUALLY.)         20
      DIMENSION X(M),D(M)                                               30
      REAL MEV,KEV,MIBA                                                 40
      KCNTR=0                                                           50
      GCTO 10                                                           60
      ENTRY ECONV1(Y,E)                                                 70
      KCNTR=1                                                           80
   10 K=0                                                               90
      DATA MEV/'MEV '/,KEV/'KEV '/,MIBA/'MB  '/,PERC/'PERC'/           100
      IF(E.EQ.PERC) GOTO 10                                            110
      IF(E.EQ.MEV) K=6                                                 120
      IF(E.EQ.KEV) K=3                                                 130
      IF(E.EQ.MIBA) K=-3                                               140
      IF(KONTR.EQ.1) GOTO 2                                            150
      DO 1 I=1,M                                                       160
    1 X(I)=X(I)*(10.**K)                                               170
      RETURN                                                           180
    2 Y=Y*(10.**K)                                                     190
      RETURN                                                           200
      ENTRY PERCNT(M,X,D)                                              210
      DO 3 I=1,M                                                       220
    3 D(I)=D(I)*X(I)*.01                                               230
      RETURN                                                           240
      END                                                              250
```

```
      SUBROUTINE ERRDET(ET,IRTYP)                                       10
C----------DETERMINE PHYSICAL UNITS OF ERROR BARS                        20
C----------READ FROM NEUDADA TAPE ACCORDING NEUDADA CONVENTIONS.         30
      INTEGER*2 ETS,IRA,IRP,IRS,ERA(6),ERP(6),ERS(2)                    40
       COMMON/INOUT/KOUT                                                50
      INTEGER*2 ITS                                                     60
      DIMENSION ET(2)                                                   70
      DIMENSION ATEXT(5,6),PTEXT(5,6),STEXT(5,2)                        80
      DATA IRA/6/,IRP/6/,IRS/2/                                         90
      DATA ERA/'LE','LR','LL','L<','L>','LS'/                          100
      DATA ERP/'PE','PR','PP','P<','P>','PS'/                          110
      DATA ERS/'MR','NR'/                                              120
      DATA ATEXT     /'ABSO','LUTE',' ERR','OR-U','NITS','ABSO','LUTE',130
     1               ' ERR','OR-U','NITS','ABSO','LUTE',' ERR','OR-U',140
     2'NITS','AVER','AGE ','ABS.',' ERR','OR <','AVER','AGE ','ABS.',  150
     3' ERR','OR >','ABS.',' STA','TIST','. ER','ROR '/                160
      DATA PTEXT     /'ERRO','R IN',' PER','CENT','    ','ERRO','R IN',170
     1               ' PER','CENT','    ','ERRO','R IN',' PER','CENT',180
     2'    ','AVER','. PE','RC. ','ERRO','R < ','AVER','. PE','RC. ',  190
     3'ERRO','R > ','STAT','IST.',' ERR','OR I','N %'/                 200
      DATA STEXT/'ERRO','R IN',' MS/','METE','R   ','ERRO','R IN',     210
     1           ' NS/','METE','R   '/                                 220
      LOGICAL*1 LETS(2)                                                230
      EQUIVALENCE(LETS(1),ETS)                                         240
      CALL STRING(LETS(1),ET(1),1)                                     250
      CALL STRING(LETS(2),ET(2),1)                                     260
      GCTO 5                                                           270
      ENTRY ERRDT(ITS,IRTYP)                                           280
      ETS=ITS                                                          290
    5 DC 10 J=1,IRA                                                    300
      IF(ETS.NE.ERA(J)) GOTO 10                                        310
      IRTYP=1                                                          320
      WRITE(KOUT,601) ETS,(ATEXT(I,J),I=1,5)                           330
  601 FCRMAT(8X,' ERRORTYPE ',A2,' MEANS ',5A4//)                      340
      GOTO 90                                                          350
   10 CONTINUE                                                         360
      DC 20 J=1,IRP                                                    370
```

```
      IF(ETS.NE.ERP(J)) GOTO 20                                          380
      IRTYP=2                                                            390
      WRITE(KOUT,601) ETS,(PTEXT(I,J),I=1,5)                             400
      GOTO 90                                                            410
   20 CONTINUE                                                           420
      DO 30 J=1,IRS                                                      430
      IF(ETS.NE.ERS(J)) GOTO 30                                          440
      IRTYP=2+J                                                          450
      WRITE(KOUT,601) ETS,(STEXT(I,J),I=1,5)                             460
      GOTO 90                                                            470
   30 CONTINUE                                                           480
      WRITE(KOUT,600) ETS                                                490
  600 FORMAT('      FEHLERTYP=',A2,' NICHT ERKANNT'//)                   500
      IRTYP=0                                                            510
      GOTO 90                                                            520
   90 RETURN                                                             530
      END                                                                540
```

```
      SUBROUTINE PCHECK(N1,X1,N2,X2,NA,NB)                               10
      DIMENSION X1(1),X2(1)                                              20
      COMMON/INOUT/KOUT                                                  30
C                                                                        40
C         CHECK WHETHER ALL INTERPOLATION POINTS LY WITHIN THE BOUNDARY   50
C         VALUES X1(1),X1(N1)                                            60
      DO 10 I=1,N2                                                       70
      IF(X2(I).LT.X1(1)) GOTO 10                                         80
      NA=I                                                               90
      GOTO 20                                                           100
   10 CONTINUE                                                          110
      NA=N1+1                                                           120
      NB=N1                                                             130
      GOTO 100                                                          140
   20 J=N2                                                              150
      DO 30 I=NA,N2                                                     160
      IF(X2(J).GT.X1(N1)) GOTO 30                                       170
      NB=J                                                              180
      GOTO 100                                                          190
   30 J=J-1                                                             200
      NB=NA-1                                                           210
  100 IF(NA.EQ.1) GOTO 110                                              220
      WRITE(KOUT,601) X2(NA),X1(1)                                      230
  601 FORMAT(//5X,'MESSAGE:ALL INTERPOLATION POINTS <=',1PE13.5/        240
     1 10X,'ARE BELOW FIRST POINT FOR WHICH THE CURVE IS DEFINED =',1PE1 250
     23.5/10X,'ACTION TAKEN: THESE POINTS ARE SKIPPED.'//)              260
  110 IF(NB.EQ.N2) GOTO 120                                             270
      WRITE(KOUT,602) X2(NB),X1(N1)                                     280
  602 FORMAT(//5X,'MESSAGE: ALL POINTS >= ',1PE13.5/                    290
     1 10X,'ARE ABOVE LAST POINT FOR WHICH THE CURVE IS DEFINED =',1PE13 300
     2.5/10X,'ACTION TAKEN:THESE POINTS ARE SKIPPED.'//)                310
  120 IF(NB.GE.NA) GOTO 130                                             320
      WRITE(KOUT,603)                                                   330
  603 FORMAT(//5X,'ERRORMESSAGE:NONE OF THE INTERPOLATION POINTS IS WITH 340
     1N THE RANGE,WHERE THE CURVE IS DEFINED.'/                         350
     2 10X,'ACTION TAKEN : CURVE IS NOT INTERPOLATED.'//)               360
  130 RETURN                                                            370
      END                                                               380
```

4.2 Karlsruhe Version of SCORE

The SCORE program developped by C.L. Dunford for interactive graphic cross section evaluation was adapted to the file formats used at Karlsruhe (KEDAK for evaluated data, NEUDADA for experimental data) by I. Langner and R. Meyer. The following user guide is essentially a translation of unpublished internal reports written by I. Langner.

Adaptation of SCØRE to KEDAK and NEUDADA Data

SCØRE Version III-K-172

I. Langner

Contents

Introduction

SCØRE - a program system for cross section evaluation - was developped by Atomics International and IBM Palo Alto for interactive graphic data evaluation at special terminal stations, the IBM 2250 terminal /9/.

The SCØRE system /1/, /12/ has recently been adapted to experimental data in NEUDADA format /7/ and to evaluated data in KEDAK format /3/, /4/.

Originally SCØRE was designed for processing of experimental data in SCISRS format (SCISRS: Sigma Center Information Storage and Retrieval System) and evaluated data in ENDF/B format /2/.

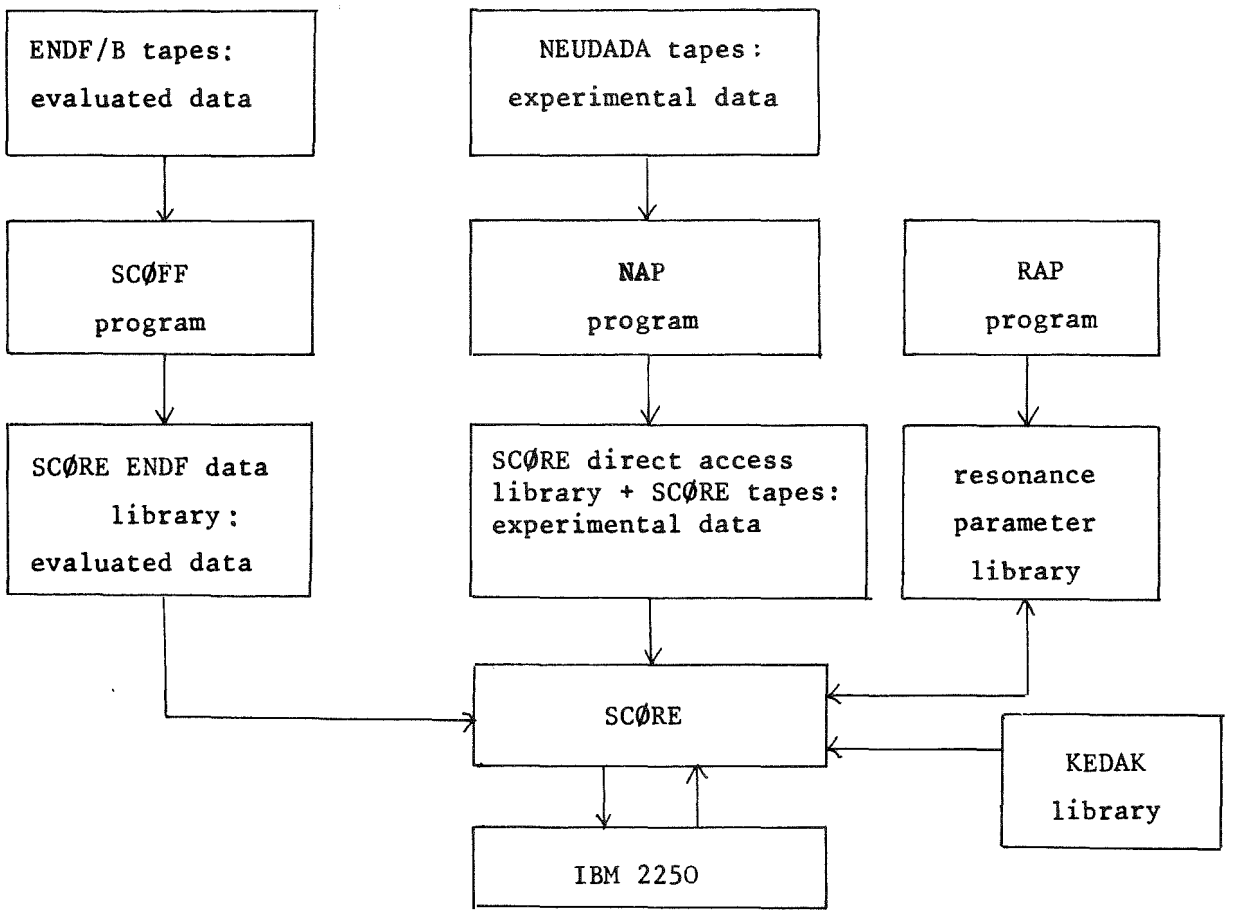
The Karlsruhe Version III-K-1972 is based on SCØRE Version III-1970. The changes and additions permit reading of KEDAK data from the KNDF library (Karlsruhe Nuclear Data File) and their display on screen. Curves of KEDAK data can be displayed either alone or together with experimental data. The SCØRE Version III-K-1972 was initiated by Dr. R. Meyer who also participated in its development.

A very informative general description of SCØRE with reproductions of the various Figures displayed on the screen can be found in Ref. /12/.

4.2.1 Information for SCØRE users

4.2.1.1 General Introduction to SCØRE

SCØRE consists of the SCØRE program proper and three administration programs for the associated data libraries: SCØFF, RAP and NAP.



The structure of the data files, as they are distributed on tape by the neutron data centers, does not allow rapid processing. Therefore the data are reorganized in special formats that were developped for SCØRE:

1. The SCØFF program treats ENDF/B tapes.

2. The RAP program establishes the resonance parameter library.

3. The NAP program (NEUDADA Adaptation Program) writes NEUDADA files /7/ of experimental data in a SCØRE format and produces, apart from SCØRE tapes, a direct-access library on disc (see 2.3).

   SCØRE can process data from tape, disc (direct access) and from the IBM-2250 terminal - via keybord, light pen and programmed-function keys.

4.2.1.2 The job control cards for SCØRE

The following listing shows all possible job control cards for SCØRE:

```
//INR048SC JOB (0048,101,P6M1A),LANGNER,CLASS=G,REGION=160K,TIME=3
/*SETUP DEVICE=2250,ID=RESERV
/*SETUP DEVICE=TAPE9,ID=(SCORE1,NORING,,NL)
/*SETUP DEVICE=TAPE9,ID=(ENDFB1,NORING,,NL)
/*SETUP DEVICE=2314,ID=GFKC16
//EXEC FFG,LIB=NUSYS,NAME=SCORE,REGION.G=160K,TIME.G=5
//G.SCOPE DD UNIT=2250
//TAPE01 DD UNIT=2400,VOL=SER=SCORE1,DISP=OLD,LABEL=(1,NL)
//TAPE02 DD UNIT=2400,VOL=SER=ENDFB1,DISP=OLD,LABEL=(1,NL)
//DISK01 DD UNIT=2314,VOL=SER=GFK016,DISP=SHR
//G.FT09F001 DD VOL=REF=*.DISK01,DISP=(NEW,DELETE,DELETE),
//    DCB=(RECFM=V,LRECL=3524,BLKSIZE=3528,BUFNO=1),
//    SPACE=(3524,(5,1)),DSNAME=RESTEMP
//G.FT16F001 DD VOL=REF=*.DISK01,DISP=OLD,DSNAME=ENDFB,
//    DCB=(RECFM=F,BLKSIZE=3600,LRECL=3600,BUFNO=1)
//G.FT17F001 DD VOL=REF=*.DISK01,DISP=OLD,DSNAME=NEUDAD,
//    DCB=(RECFM=F,BLKSIZE=3520,LRECL=3520,BUFNO=1)
//G.FT19F001 DD VOL=REF=*.DISK01,DISP=OLD,DSNAME=RESLIB
//    DCB=(RECFM=U,BLKSIZE=400,BUFNO=1)
//G.FT20F001 DD VOL=REF=*.TAPE01,DISP=OLD,LABEL=(1,NL,,IN),
//    DSN=NAP.LABEL,DCB=(RECFM=VBS,LRECL=68,BLKSIZE=3744)
//G.FT21F001 DD VOL=REF=*.TAPE01,DISP=OLD,LABEL=(2,NL,,IN),
//    DSN=NA023,DCB=*.FT20F001
//G.FT22F001 DD VOL=REF=*.TAPE01,DISP=OLD,LABEL=(3,NL,,IN),
//    DSN=FE000,DCB=*.FT20F001
//G.FT23F001 DD VOL=REF=*.TAPE01,DISP=OLD,LABEL=(4,NL,,IN),
//    DSN=FE054,DCB=*.FT20F001
//G.FT24F001 DD VOL=REF=*.TAPE01,DISP=OLD,LABEL=(5,NL,,IN),
//    DSN=LI000,DCB=*.FT20F001
//G.FT25F001 DD VOL=REF=*.TAPE01,DISP=OLD,LABEL=(6,NL,,IN),
//    DSN=LI007,DCB=*.FT20F001
//G.FT26F001 DD VOL=REF=*.TAPE01,DISP=OLD,LABEL=(7,NL,,IN),
//    DSN=PU241,DCB=*.FT20F001
//G.FT27F001 DD VOL=REF=*.TAPE02,DISP=OLD,LABEL=(1,NL,,IN),
//    DSN=SCOFF.LABEL,DCB=(RECFM=V,LRECL=360,BLKSIZE=3608,BUFNO=1)
//G.FT28F001 DD VOL=REF=*.TAPE02,DISP=OLD,LABEL=(2,NL,,IN),
//    DSN=LI007B,DCB=*.FT27F001
//G.FT29F001 DD VOL=REF=*.TAPE02,DISP=OLD,LABEL=(3,NL,,IN),
//    DSN=NA023B,DCB=*.FT27F001
//G.FT30F001 DD VOL=REF=*.TAPE02,DISP=OLD,LABEL=(4,NL,,IN),
//    DSN=FE000B,DCB=*.FT27F001
//G.FT31F001 DD VOL=REF=*.TAPE02,DISP=OLD,LABEL=(5,NL,,IN),
//    DSN=PU241B,DCB=*.FT27F001
//G.SYSIN DD DUMMY
```

The following example of job control cards requires only KEDAK and NEUDADA data.

```
//INRC48SC JOB (C048,101,P6M1A),LANGNER,CLASS=G,REGION=160K,TIME=3
// EXEC FHG,LIB=NUSYS,NAME=SCORE,REGION.G=160K,TIME.G=5
//G.SCOPE DD UNIT=250
//TAPE01 DD UNIT=2400,VOL=SER=SCORE1,DISP=OLD,LABEL=(1,NL)
//G.FT01F001 DD UNIT=2314,VOL=SER=NUSYS0,DISP=SHR,DSN=KNDF
//G.FT17F001 DD UNIT=2314,VOL=SER=GFK016,DISP=SHR,DSN=NEUDAD.DA.LAN,
//    DCB=(RECFM=F,LRECL=3520,BLKSIZE=3520,BUFNO=1)
//G.FT20F001 DD VOL=REF=*.TAPE01,DISP=OLD,LABEL=(1,NL,,IN),
//     DCB=(RECFM=VBS,LRECL=68,BLKSIZE=3744,BUFNO=1),DSN=NAP.LABEL
//G.FT21F001 DD VOL=REF=*.TAPE01,DISP=OLD,LABEL=(2,NL,,IN),
//    DSN=NA023,DCB=*.FT20F001
//G.FT22F001 DD VOL=REF=*.TAPE01,DISP=OLD,LABEL=(3,NL,,IN),
//    DSN=FE000,DCB=*.FT20F001
//G.FT23F001 DD VOL=REF=*.TAPE01,DISP=OLD,LABEL=(4,NL,,IN),
//    DSN=FE054,DCB=*.FT20F001
//G.FT24F001 DD VOL=REF=*.TAPE01,DISP=OLD,LABEL=(5,NL,,IN),
//    DSN=LI000,DCB=*.FT20F001
//G.FT25F001 DD VOL=REF=*.TAPE01,DISP=OLD,LABEL=(6,NL,,IN),
//    DSN=LI007,DCB=*.FT20F001
//G.FT26F001 DD VOL=REF=*.TAPE01,DISP=OLD,LABEL=(7,NL,,IN),
//    DSN=PU241,DCB=*.FT20F001
//G.SYSIN DD DUMMY
```

As shown by the EXEC card SCØRE is part of the library LOAD.NUSYS.

4.2.1.3 Input for SCØRE at the IBM 2250 terminal

A summary of the SCØRE input is appended under A3. below.

4.2.1.2.1 Input modes

Three different input modes exist:

1. light pen display on the screen,
2. input via keybord as alphameric text, fixed- or floating-point numbers,
3. input via the programmed-function keyboard controlling program execution.

Note: Key no. 1 of the programmed-function keyboard has the same function as the END key of the keyboard.

4.2.1.3.2 Input for SCØRE, mainly for KEDAK data

Reproductions of many of the Figures displayed on the screen can be found in Ref. /12/.

First figure: The user can choose between two options: KEDAK and NEUDADA (cf. 4.2.3.1). The option selected by light pen is displayed in large size /9/ on the screen. The information is transferred to the main program when the END key is pressed.

On the second figure SCØRE requests user and case identification (cf. /1/, Vol. I, p. 16).

KEDAK option

If the KEDAK option was selected on the first figure a list of the alphameric names /5/ of the isotopes available in the KNDF library /3/, /4/ is displayed in the third figure. Fifteen names are shown per figure. Selecting the option PAGE with the light pen the user can "turn the page", i.e. get the next 15 names displayed. He chooses the isotope with the light pen. The chosen isotope is shown in large size. If the KNDF library does not contain any reaction types for the chosen isotope that can be processed by SCØRE, pressing of the END key causes display of the first 15 isotopes again. Otherwise a list of reaction types available for SCØRE processing follows in the next figure. Again the choice is made by light pen. The chosen reaction type is displayed in large size in the list, and also a request for the energy limits. The energy limits must be typed in. Pressing of the END key causes display of the cross section curve in "Basic SCØRE Data Display" (/1/, Vol. I, p. 16). Exceptions are the reaction types SGNC (differential elastic-scattering cross section) and SGIZ (partial inelastic-scattering cross section) for which a list of incident (SGNC) or excitation (SGIZ) energies is shown first, with PAGE having the same function as described above. After selecting the appropriate energy with the light pen the user can get the cross section curve displayed by pressing the END key.

The following selection of the original options (cf. /1/, Vol. I, p. 27) are valid, i.e. can be chosen with the light pen:

MAIN 1/3	option list 1 -
RESTART	
PRINT	
EXPAND	
QUIT	
MAIN 2/3	option list 2 -
KEDAK	

Meaning:

MAIN 1/3	display option list MAIN 2/3,
RESTART	jump back to the start of the program, i.e. display first picture again,
PRINT	print the data displayed on the screen,
EXPAND	Expand the graph portion of the display to full screen size with omission of comment and option lists, (EXPAND can be turned off again by pressing the END key),
QUIT	end terminal session ,
MAIN 2/3	display option list MAIN 3/3 ,
REPLOT	change co-ordinate system according to input furnished by user ,
MAIN 3/3	display option list MAIN 1/3 ,
KEDAK	display KEDAK (evaluated) data in curve form together with experimental data points.

NEUDADA option

If the NEUDADA option is chosen in the first figure the description in Ref. /1/ is valid for subsequent input. Selecting KEDAK from option list 3/3 one obtains the (evaluated) KEDAK data in curve form together with the (experimental) NEUDADA data /7/.

4.2.1.3.3 Summary of light-pen input

Figure 1	(1) select KEDAK or NEUDADA option by light pen, (2) press END key.
Figure 2	user identification etc., cf. Ref. /1/, Vol. I, pp. 16, 17
Figure 2	list of isotope names (1) turn page, if necessary, by touching PAGE with the light pen, (2) select isotope, (3) press END key.
Figure 4	list of reaction types (1) select reaction type with the light pen, (2) supply energy limits using the key board, (3) press END key.
Figure 5	list of energies (for SGNZ, SGIZ only) (1) turn page, if necessary, by touching PAGE with the light pen, (2) select energy, (3) press END key.

4.2.1.4 Job control cards and input for NAP

For NAP input see also Ref. /1/, Vol. II, pp. 19-21. The following example shows the job control cards for reformatting of a NEUDADA tape (to be requested from CCDN, Saclay, in "internal expanded format") in SCØRE format and creation of a SCØRE library of experimental data in direct-access form on disc (cf.4.2.2.3).

```
//INRC48NA JOB (0048,101,P6M10),LANGNER,CLASS=A,TIME=3
/*FORMAT PU,DDNAME=FT04F001,FORMS=STANZ
/*SETUP DEVICE=TAPE9,ID=(SCORE1,,,NL)
/*SETUP DEVICE=TAPE9,ID=(DV0645,NORING,,SL)
/*SETUP DEVICE=2314,ID=GFK016
// EXEC FHG,LIB=NUSYS,NAME=NAP
//G.FT07F001 DD SYSOUT=B
//G.FT08F001 DD UNIT=TAPE9,VOL=SER=DV0645,DSN=NEUDA1,
//   LABEL=1,DISP=OLD
//G.FT09F001 DD UNIT=TAPE9,VOL=SER=SCORE1,LABEL=(8,NL),DISP=(,PASS),
//   DSN=NAP.LABEL,DCB=(RECFM=VBS,LRECL=68,BLKSIZE=3744)
//G.FT09F002 DD UNIT=TAPE9,VOL=SER=SCORE1,LABEL=(9,NL),DISP=(,PASS),
//   DSN=FE000,DCB=*.FT09F001
//G.FT09F003 DD UNIT=TAPE9,VOL=SER=SCORE1,LABEL=(10,NL),DISP=(,PASS),
//   DSN=FE056,DCB=*.FT09F001
//G.FT17F001 DD UNIT=2314,VOL=SER=GFK016,DSN=NEUDAD.DA,
//   SPACE=(3520,520),DISP=(NEW,KEEP)
//G.SYSIN DD *
16/11/71   1            1            1           -1            0
SCOREC02  SECOND SCORE TAPE
FE000  NEUDADA CCDN
26000              55.847            2            1
FE056  NEDADA CCDN
26056              55.935            2            2
```

The second example shows the job control cards needed to print a table of content of a SCØRE tape and to set up a direct-access file (if no such file exists):

```
//INR048NA JOB (0048,101,P6M10),LANGNER,CLASS=A,TIME=3
/*FORMAT PU,DDNAME=FT04F001,FORMS=STANZ
/*SETUP DEVICE=2314,ID=GFK016
/*SETUP DEVICE=TAPE9,ID=(SCORE1,NORING,,NL)
// EXEC FHG,LIB=NUSYS,NAME=NAP
//G.FT07F001 DD SYSOUT=B
//G.FT10F001 DD UNIT=TAPE9,VOL=SER=SCORE1,LABEL=(8,NL,,IN),
//   DSN=NAP.LABEL,DISP=OLD,DCB=(RECFM=VBS,LRECL=68,BLKSIZE=3744)
//G.FT10F002 DD UNIT=TAPE9,VOL=SER=SCORE1,LABEL=(9,NL,,IN),
//   DSN=FE000,DISP=OLD,DCB=*.FT10F001
//G.FT10F003 DD UNIT=TAPE9,VOL=SER=SCORE1,LABEL=(10,NL,,IN),
//   DSN=FE056,DISP=OLD,DCB=*.FT10F001
//G.FT17F001 DD UNIT=2314,VOL=SER=GFK016,DSN=NEUDAD.LAN,
//   SPACE=(3520,520),DISP=(NEW,KEEP)
//G.SYSIN DD *
18/11/71   0            0            1            1            3
           8            2
SCOREOC1     SCOREOO2
       11023      1C10000 7/19/67             2230
       26000      1C20000 7/19/67             7485
       26054      1C30000 7/20/67              238
        3000      1C40000 7/20/67             1123
        3007      1050000 7/20/67             1037
       94241      1C60000 11/27/67           16544
       26000      2C1000018/10/71            17373
       26056      2C2000018/10/71              128
```

Short input description

The structure of the experimental-data SCØRE library and the related terminology are explained in Ref. /1/, Vol. II, chap. II.

first card, format (2A4,1X,I3,5I12)

date	2 words
NF	number of files to be converted on the NEUDADA tape, = 0 if no conversion is requested
IFILE	= 0: do not write an identification file. = 1: write an identification file for a new SCØRE tape
IDACS	= 0: do not create a direct-access library = 1: create a direct-access library
IUPD	= -1: create table of tape identifications and isotope table (these tables are neither printed nor punched if NF=0) = 0: the tables are to be transferred from the direct-access library disc = 1: the tables are to be read in
NPRINT	number of files on the SCØRE tape for which the tables are to be printed.

tape identification card, format (16A4)

(missing if IFILE=0)

TAPEX	2 words, tape identification,
HEAD	14 words, arbitrary information.

tape identification and isotope table card, format (2I12)

(missing if IFILE=1)

NIS	number of isotopes
NTS	number of tapes

tape identification card, format (2A4,4X,2A4,4X etc.)

TAPE(J,I) NTS tape identifications are to be read, six per card (J= 1,2, I=1,NTS). I is the tape number.

isotope table card, format (2I12,2A4,4X,I12)

ISOTOP(I) isotopic identification in the form Z*1000+A,

ISNAM(I) tape and file number of the actual and preceding data sets in the form $(NT*100+NF)_{act.}+10000+(NT*100+NF)_{prec.}$,

DATEIS(J,I) date, 2 words (J=1,2)

NRECS(I) total number of sets on the file

The last two cards must be read in for all NIS isotopes.

Title card, format (A2,A3,7X,E12.8,2I12)

ZNUM 2 alphameric characters, proton number Z

ANUM 3 alphameric characters, nucleon number A

AMASS atomic weight (^{12}C system)

NT tape number

NF file number

Title card and isotope card are to be given for each isotope.

Sorting of NEUDADA data (cf. 4.2.2.3) with SØRT/MERGE Utility /9/

Job control cards and input:

```
//INR048SO JOB (0048,101,P6M1A),LANGNER,MSGLEVEL=(1,1),CLASS=A
/*SETUP DEVICE=TAPE9,ID=DV0645
/*SETUP DEVICE=TAPE9,ID=(ND0429,NORING,,NL)
// EXEC PGM=SORT,PARM='MSG=AP'
//SORTLIB DD DSNAME=SYS1.SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=A
//SYSLIN DD UNIT=SYSDA,SPACE=(80,(10,10))
//SYSLMOD DD UNIT=SYSDA,SPACE=(3600,(20,20,1))
//SYSUT1 DD UNIT=(SYSDA,SEP=(SORTLIB,SYSLMOD,SYSLIN)),
//   SPACE=(1000,(60,20))
//SYSPRINT DD DUMMY
//SORTIN DD UNIT=TAPE9,VOL=SER=ND0429,LABEL=(2,NL),
//   DCB=(RECFM=FB,LRECL=75,BLKSIZE=750),DISP=(OLD,PASS)
//SORTOUT DD DSN=LNDSOR,UNIT=TAPE9,VOL=SER=DV0645,
//   LABEL=(1,SL),DISP=(,PASS),
//   DCB=(RECFM=FB,LRECL=75,BLKSIZE=750)
//SORTWK01 DD UNIT=SYSDA,SPACE=(TRK,(100),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(TRK,(100),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(TRK,(100),,CONTIG)
//SYSIN DD *
 SORT FIELDS=(1,9,CH,A,48,4,FL,A,60,4,FL,A),SIZE=E16000
```

Note: SIZE=E... depends on the number of data sets in the file that is to be sorted.

4.2.2 Modifications in SCØRE

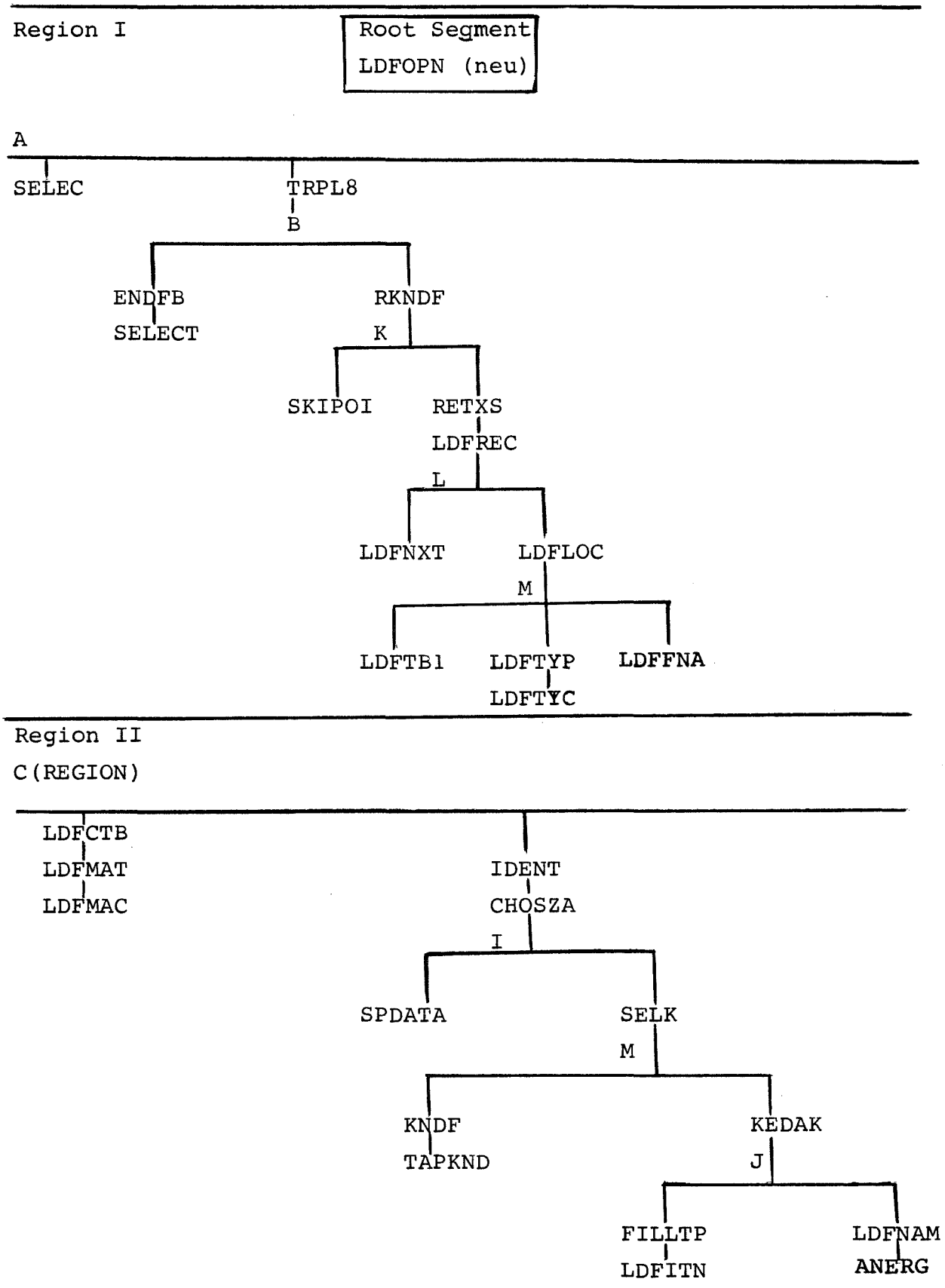
The source programs for the SCØRE subroutines exist as "members" of a "partitioned data set" /10/. At present this data set consists of 23 members.

New additions are the two "members" NAP and KNDF. Changes were made in the members OVERLAY, ONE, FOUR, FIVE, SIX and EIGHT.

4.2.2.1 Changes in the overlay structure

The object program of SCØRE needs 300 kBytes of memory space without, 126 kBytes with overlay. In the overlay structure the memory space is subdivided into four regions. The changes and the addition of subroutines for processing of KEDAK data resulted in the following modifications in Region I and II.

Changes in Overlay Structure



The "root segment" /11/ contains the programs MAIN, GIO, GTVDD, WTOR, DACON, CMBCD, CENTER, INFILQ, ADDBLK, EXINGR, STIMPT, PLOTC, PLOTE, CHGCHR, STOBUF, GREAD, SETKEY, ENLP, INCUR, IFNDBF, RESTSCL, SCALAR and LDFOPN.

List of all SCØRE overlay statements:

```
OVERLAY A
INSERT SELEC
OVERLAY A
INSERT FIG2
OVERLAY B
INSERT SORT
OVERLAY B
INSERT DISPAR
OVERLAY A
INSERT DELETE,CORECT
OVERLAY A
INSERT DALIST,PRIND
OVERLAY A
INSERT DAFIT
OVERLAY B
INSERT MODNOD,SOND
OVERLAY B
INSERT CURVFT
OVERLAY B
INSERT LEGCAL
OVERLAY A
INSERT RESFIT
OVERLAY A
INSERT TRPL8
OVERLAY B
INSERT ENDFB,SELECT
OVERLAY B
INSERT RKNDF
OVERLAY K
INSERT SKIPOI
OVERLAY K
INSERT RETXS,LDFREC
OVERLAY L
INSERT LDFNXT
OVERLAY L
INSERT LDFLOC
OVERLAY M
INSERT LDFTB1
OVERLAY M
INSERT LDFTYP,LDFTYC
OVERLAY M
INSERT LDFFNA
OVERLAY A
INSERT  WAITR,GT2250,IHCFEXIT
OVERLAY C(REGION)
INSERT LDFCTB,LDFMAT,LDFMAC
OVERLAY C
INSERT IDENT,CHOSZA
```

```
OVERLAY I
INSERT SPDATA,INPUT,LABEL,RECORD,PCINT
OVERLAY I
INSERT SELK
OVERLAY H
INSERT KNDF,TAPKND
OVERLAY H
INSERT KEDAK
OVERLAY J
INSERT FILLTP,LOFITN
OVERLAY J
INSERT LDFNAM,ANERG
OVERLAY C
INSERT RETREV,RECORE,ICS,TAPER,FL3
OVERLAY D
INSERT TAPERS,RITDSK,REACT,TAB1,TABLE,TABLE3,TABLE4,TRANL
OVERLAY D
INSERT FL5
OVERLAY D
INSERT FL4,XERIES,MLTERP
OVERLAY C
INSERT BLK10
OVERLAY D
INSERT DISPLA,BORDER,GRIDT,GRID,MPLOT,NOPT
OVERLAY D
INSERT BSETUP,SPLSUP,INTERP
OVERLAY E
INSERT LEGEND,SERIES
OVERLAY E
INSERT SPLINE,CHISQR
OVERLAY E
INSERT ENGSPL
OVERLAY D
INSERT COLPSE,ANERES,EXPAND,RESISC
OVERLAY E
INSERT SELETY,SELRES,SELDOP,ROWSET
OVERLAY E
INSERT SELZA,ANEWZA
OVERLAY E
INSERT RESMOD,RSETUP
OVERLAY E
INSERT  PKFIT,VALFIT
OVERLAY D
INSERT SAP,SIGMAS
OVERLAY E
INSERT COMBCC,LAIN
OVERLAY E
INSERT ZERDIN,PD,DOPBR
OVERLAY D
INSERT ADLER,SIG,WIDTH,CHIPSI,IHCLEXP,IHCLSQRT
OVERLAY D
INSERT DELCUR,LIMITS,CHIP,MULP,UPDTIL,ESHIFT
OVERLAY F(REGION)
INSERT  DISPLAY,PFKEYS,WRTXTM
```

```
OVERLAY F
INSERT IHCSEXP,IHCERXPR,IHCERXPI,IHCSSCN,IHCLSCN
OVERLAY G(REGION)
INSERT ERRCOR,HELP,CENTER,CMBCD
OVERLAY G
INSERT TRACK
OVERLAY G
INSERT PLINE,PLOTP
OVERLAY G
INSERT  FIG1
OVERLAY G
INSERT ROWISC,ROWTAB
OVERLAY G
INSERT CADALM,CALIM,LCML
ENTRY MAIN
NAME SCORE(R)
```

4.2.2.2 SCØRE routines

4.2.2.2.1 MAIN PROGRAM SCØRE

The changes permit access to KEDAK data /3/, /4/ from the KNDF library and display of the cross sections as curves on the screen. Call of the subroutine LDFOPN /6/ makes the KNDF library (Karlsruhe Nuclear Data File) on unit 1 available to SCØRE and fills COMMON/CDFMT/ with the material names /5/ of the isotopes stored in KEDAK format. The newly added call of subroutine SELEC permits selection between the KEDAK and NEUDADA option. The option KEDAK was added to the option list MAIN 3/3 /1/. Call of the subroutine KNDF makes the KEDAK data available for display in curve form simultaneously with NEUDADA data. Call of the subroutine KEDAK yields KEDAK data also if experimental data are not available or not transferred to core storage.

4.2.2.2.2 Subroutines ERRCØR, SPDATA, CHOSZA

An error message for reading of KEDAK data was added to subroutine ERRCØR. In subroutine SPDATA the section for the figure "Case Identification" (cf. /1/, Vol. II, p. 16) was replaced by subroutine IDENT (cf. 4.2.3.1.2). Subroutine CHØSZA, which provides for interactive choice of isotope and reaction type on the screen, was changed so that also the 8-byte alphameric KEDAK isotope and reaction type names /5/ can be processed.

4.2.2.3 Program NAP - compilation of SCØRE library from experimental (NEUDADA) data

NAP (NEUDADA Adaptation Program) was rewritten and tested.

Experimental data from other centers reach Karlsruhe via CCDN (OECD/NEA Centre de Compilation de Données Neutroniques), Saclay. The data are transmitted on magnetic tape in NEUDADA format /7/.

The user must sort the data received on a NEUDADA tape in ascending order of Z, A and energy with the SØRT/MERGE utility program /8/ (see 4.2.1.4), if necessary. The NAP program reformats the data thus prepared and writes than onto a library tape. This library tape contains an identification record and a directory of addresses for the content of the tape. The data themselves are organized in such a way that one data block corresponds to one record of the direct-access library which is also generated by NAP. This library contains a table of available isotopes and the tape names and file numbers of the corresponding library tapes. This table is updated by NAP. Transfer of the data from tape to direct-access library is performed by SCØRE /1/ if necessary.

4.2.3 New subroutines for SCØRE

4.2.3.1.1 SELEC - choice between NEUDADA and KEDAK

Subroutine SELEC, called from MAIN PROGRAM SCØRE, generates the first figure for SCØRE. The following text is displayed on the screen:

SCØRE

CROSS SECTION EVALUATION SYSTEM
ADAPTED TO NEUDADA AND KEDAK DATA

SELECT BY LIGHT PEN DATA TYPE
TO BE RETRIEVED FIRST

KEDAK

NEUDADA

Thus the user can choose between KEDAK or NEUDADA data. If he chooses the NEUDADA option the SCØRE version described in /1/ is at his disposal with the additional possibility to display KEDAK data in curve form together with the experimental NEUDADA data - provided that data for the chosen isotope, reaction type and energy range exist in the KNDF library /3/, /4/. If he chooses the KEDAK option he gets the cross sections from the KNDF library displayed on the screen, regardless whether NEUDADA data are available for the chosen isotope, reaction type and energy range, or not.

The option chosen is displayed in large size on the screen. Pressing of the END key or the equivalent key no. 1 on the programmed-function keyboard causes transmittal of the information to the main program. Erroneous pressing of the END key without prior selection of the format has no consequence, SELEC will still wait for the light pen.

SELEC is called by

```
CALL SELEC(IOPT),
```

where IOPT is set by SELEC. (IOPT=1: KEDAK, IOPT=2: NEUDADA.)

SELEC utilizes the following subroutines from the "graphic subroutine package" of SCØRE: INPICT,WRTXTM,SETKEY,ENDKEY,STPICT,EXINGR,CHGCHR whose functions are as follows:

INPICT	initializes the figure,
WRTXTM	writes a character string onto a specified place on the screen,
SETKEY	indicates the beginning of a series of graphic commands associated with a key,
ENDKEY	indicates the end of the series,
STPICT	brings a block of graphic commands into the buffer of the IBM 2250 and starts the figure,

EXINGR controls the input to the IBM 2250,

CHGCHR changes the mode of a character string from protected to unprotected and vice versa

4.2.3.1.2 IDENT - user and case identification

The subroutine IDENT is essentially a program section taken out of subroutine SPDATA so as to be available also to KEDAK. More specifically it is the part that generates the first figure for SCØRE (cf. /1/, Vol. I, p. 16) in which the case identification is requested.

IDENT is called by

```
CALL IDENT(IER),
```

where IER is an error flag:

IER = 1 : no error, continue;
IER = 2 : jump back to the start of the program;
IER = 3 : error, terminate execution.

IDENT calls the SCØRE routines FIG1, DACON, ERRCOR and utilizes CØMMØN/BLK1/:

FIG1 generates the figure,

DACØN converts numbers from the internal fixed- or floating-point representation to alphameric or vice versa,

ERRCØR is a routine for error messages and error response.

CØMMØN/BLK1/ABCD(5,4),ICASE,CNUM
The array ABCD contains user identification and date, ICASE is the case number and CNUM the figure number for this case.

4.2.3.2 Subroutines for the organization of KEDAK data

4.2.3.2.1 KNDF - KEDAK data for curve display

Subroutine KNDF organizes cross sections from the KNDF library /3/, /4/ for curve display together with experimental data. Proton number and reaction type number are taken over from the experimental data (option NEUDADA), likewise the energy range is fixed by the experimental data. For differential elastic-scattering cross sections the data for the requested energy are calculated, if necessary, by linear interpolation between the angular distributions existing on KEDAK.

Three quantities are taken from CØMMØN/BLK4/:

ENER	incident energy for the differential elastic-scattering cross section,
QX	excitation energy for the partial inelastic-scattering cross section,
NREC	KEDAK number of the reaction type chosen.

The energy limits EMIN, EMAX are taken from CØMMØN/BLK5/. Subroutine KNDF stores cross sections from the KNDF library in CØMMØN/BLK8/NPØINT,X,Y,NKNDF, ZLIM(4), where

NPØINT	is the number of energies,
X	is the abscissa (energy) array,
Y	is the ordinate (cross section) array,
NKNDF	is taken as equal to NPØINT,
ZLIM	are the minimum and maximum energy and cross section values calculated by subroutine CALIMA.

Subroutine KNDF utilizes subroutines TAPKND,RKNDF (cf. 4.2.3.2.2, 4.2.3.2.3) and the SCØRE subroutines CALIMA and ERRCØR. It is called by

```
CALL KNDF(IER)
```

where IER is an error flag as explained in 4.2.3.1.2 above.

4.2.3.2.2 TAPKND - calculation of numeric KEDAK names

Subroutine TAPKND calculates numeric KEDAK names /5/ for the isotope and the reaction type selected for the NEUDADA data for which the KEDAK curve is to be displayed together with the experimental values. The isotope name is calculated from the proton number (IZ) and the nucleon number (IA) as follows

NAMES(1) = IZ*10000+IA.

The reaction type number is calculated as

NAMES(2) = MF*10000+MT*10 with MF = 3

according to the ENDF conventions /2/ for all reaction types listed in table MTS (see below), with the exception of the differential elastic-scattering cross section and the inelastic-scattering cross section for a specific excitation of the residual nucleus (NREC = 14, NREC = 15). For these latter reaction types one has

NAMES(2) = MF*10000+MT*10+2 with MF = 4.

MT is determined by NREC (number of the reaction type in table MTS) from CØMMØN/BLK4/ and table MTS. The following reaction types are available in SCØRE at present:

MTS ENDF/B	KEDAK numeric	KEDAK alphameric	Error message from ERRCØR on screen[+)]
107	31070	SGALP	SGAL
104	31040	SGD	SGD
19	30190	SGF	SGF
102	31020	SGG	SGG
106	31060	SGHE3	SHE3
103	31030	SGP	SGP
105	31050	SGH3	SGH3
4	30040	SGI	SGI
16	30160	SG2N	SG2N

[+)] For error messages on screen the type name must be shortened to four characters where necessary.

MTS ENDF/B	KEDAK numeric	KEDAK alphameric	Error message from ERRCØR on screen+)
17	30170	SG3N	SG3N
2	30020	SGN	SGN
5	30050	SGIZ	SGIZ
1	30010	SGT	SGT
2	40022	SGNC	SGNC
5	40052	SGICZ	SICZ

+) For error messages on screen the type name must be shortened to four characters where necessary.

TAPKND is called by

```
CALL TAPKND(NAMES,NZNAM,MF,IQ,IEF)
```

where

NAMES is a 5-word array with the KEDAK names inserted in words 1 and 2 by the subroutine,

NZNAM is the number of KEDAK names as set by TAPKND,

IQ is the additional KEDAK identifier for angular distributions or inelastic-scattering cross sections,

IEF is an error code (IEF=0 without errors).

4.2.3.2.3 RKNDF - reading of KEDAK data

Subroutine RKNDF prepares the KEDAK data for the energy range EMIN...EMAX in the arrays X and Y. If there are more than 150 energy values all those curve points are deleted by subroutine SKIPØI which are not required for a 1 % accuracy of the cross section curve. The cross sections for the lower and upper energy limits EMIN, EMAX are calculated by linear interpolation in the SCØRE subroutine TRPL8. Differential elastic-scattering cross sections are prepared by subroutine SELK (see 4.2.3.2.5). Reading of KEDAK data from the KNDF file is controlled by subroutine RETXS /6/. Thus the subroutines called by RKNDF are SELK,RETXS,SKIPØI,TRPL8.

RKNDF is called by

```
CALL RKNDF(NAMES,NZNAM,MF,EMIN,EMAX,X,Y,NPØINT,IEF)
```

where

NAMES	is the field for the KEDAK names of isotope and reaction type,
NZNAM	is the number of KEDAK names,
MF	= 3 or 4 according to ENDF conventions /2/,
EMIN	is the lower limit of the requested energy range,
EMAX	is the upper limit of the requested energy range,

Note that these arguments must be available, whereas the following arguments are supplied by RKNDF:

X	1-dimensional array of KEDAK energies,
Y	1-dimensional array of KEDAK cross sections,
NPØINT	number of data points transferred from the KNDF library
IEF	error code for subroutine ERRCØR (IEF=0 without error).

4.2.3.2.4 SKIPØI - deletion of data points

Subroutine SKIPØI deletes energy grid points of the KEDAK data if their number exceeds a limit that depends on the array dimensions in SCØRE (150 at present). Only those points are deleted which are not required for a curve which represents the KEDAK values with ε = 1 % accuracy.

SKIPØI is called by

```
CALL SKIPØI(XX,YY,NUMX,X,Y,NPØINT,LENX,EPS)
```

where

XX	is a 1-dimensional array for the energy values,
YY	is a 1-dimensional array for the cross section values,
NUMX	is the number of XX values (or YY values),

X	is the energy array after deletion of superfluous values,
Y	is the cross section array after deletion of superfluous values,
NPØINT	is the number of X (or Y) values,
LENX	is the maximum number of X (or Y) values
EPS	is the accuracy limit ε.

SKIPØI is repetitively called until the X (and Y) array is filled completely.

4.2.3.2.5 SELK - calculation of the differential elastic-scattering cross section

The differential elastic-scattering cross sections are calculated from the angle-integrated elastic-scattering cross section $\sigma_n(E)$ and the elastic angular distributions (center-of-mass system) $p(\mu,E)$ that are stored in the KEDAK library /3/, /4/. For a given incident energy E the angle-integrated cross section is calculated by linear interpolation, if necessary. The angular distribution is also determined by linear interpolation between the distributions stored on KEDAK. Here the interpolation is between scattering-angle cosines μ on one hand and incident energies on the other. The angular grid is taken from the lowest incident energy on KEDAK. The differential elastic-scattering cross section is then

$$\frac{d\sigma_n(E,\Omega)}{d\Omega} = \sigma_n(E) \cdot p(\mu,E),$$

in units of mb/sterad. Linear interpolation is performed by the SCØRE subroutine TRPL8. The subroutines RETXS, IDFLOC, IDFNXT /6/ are used for reading of KEDAK data.

SELK is called by

```
CALL SELK(NARG,NAMEN,EMI1,EMI2,XX,YY,MAXNUM,NUM1,IEF)
```

where

NARG	is the number of KEDAK names for the requested isotope,
NAMEN	is an array for the KEDAK names in numeric representation,
EMI1	is the lower limit of the requested energy range,
EMI2	is the upper limit of the requested energy range,

XX	is the array of scattering-angle cosines,
YY	is the array for the differential elastic-scattering cross sections,
MAXNUM	is the maximum number of cosines,
NUMI	is the actual number of cosines,
IEF	is an error code for the SCØRE routine ERRCØR (IEF=0 without errors)

4.2.3.2.6 KEDAK - KEDAK data without experimental (NEUDADA) data

Representation of KEDAK data independent of NEUDADA data (option KEDAK) is organized in subroutine KEDAK.

First KEDAK calls subroutine IDENT which supplies user and case identification. The list of isotope names on KNDF is written into CØMMØN/LDFMT/ by LDFØPN /6/ and thus transferred to the modified SCØRE routine CHØSZA via entry CHØSMA. This permits display on the screen and selection of the desired isotope by light pen.

Subroutine FILLTP /6/ yields the alphameric and numeric /5/ reaction type names for the selected isotope. The alphameric names are displayed on the screen via entry CHØSRE in CHØSZA so that a reaction type can be chosen. In case SGNC (the differential elastic-scattering cross section) or SGIZ (the partial inelastic-scattering cross section) is selected subroutine LDFNAM /6/ is called. This subroutine supplies the incident energies (for SGNC) or excitation energies (for SGIZ) for which the KNDF file contains data. Subroutine ANERG lists these energies on the screen allowing for selection of the appropriate energy by light pen.

Subroutine RKNDF (cf. 4.2.3.2.3) supplies the KEDAK data for the chosen isotope, reaction type, energy range (and - for SGNC - incident energy or - for SGIZ - excitation energy).

Subroutine CALIMA provides minimum and maximum energy and cross section values for the plot. Isotope and reaction type names and case number are written into the appropriate field for figure caption and printout.

Subroutine KEDAK utilizes the subroutines FILLTP,LDFNAM /6/, subroutine IDENT (cf. 4.2.3.1.2), RKNDF (cf. 4.2.3.2.3) and ANERG (cf. 4.2.3.2.7) and the SCØRE subroutines CHØSZA, DACØN, CENTER and ERRCØR. It is called by

```
CALL KEDAK(IER)
```

where IER is an error code (see 4.2.3.1.2).

The KEDAK data are stored in CØMMØN/BLK8/ (cf. 4.2.3.2.1).

4.2.3.2.7 ANERG - listing of energies on the screen

Subroutine ANERG displays a list of either

- incident energies for differential elastic-scattering cross sections (reaction type SGNC), or

- excitation energies for partial inelastic-scattering cross sections (reaction type SGIZ),

as available on KEDAK. The desired energy can then be selected by light pen. The display contains isotope and reaction type name, 15 energies and the word PAGE. Touching the word PAGE with the light pen the user can switch to the next 15 energies. If there are no further energies for the requested isotope the first 15 energies are shown again etc. (At present up to 200 different energies can be displayed in this way.)

The selected energy is displayed in large size. The information is transferred to subroutine KEDAK by pressing of the END key.

ANERG utilizes the following subroutines of the "graphic subroutine package" of SCØRE: INPICT,WRTCTM,SETKEY,ENDKEY,STPICT,EXINGR,CHGCHR and the SCØRE routine DACØN. It is called by

```
CALL ANERG(NAM1,N2,XNAM1,NNAM,IER)
```

where

NAM1	is a field of 2 double words which must contain the alphameric KEDAK isotope and reaction type names /5/,
N2	is the number of energies,
XNAM1	is an array containing the energies,
NNAM	is the running number of the selected energy as determined by ANERG,
IER	is an error code (IER=0 without errors).

References

/1/ C.L. Dunford, SCØRE, Vol. I und II, Atomics International, AI-AEC-12994, 1971

/2/ M.K. Drake, Data Formats and Procedures for the ENDF Neutron Cross Section Library National Neutron Cross Section Center, BNL 50274 (T-601), 1970

/3/ J.J. Schmidt, Neutron Cross Sections for fast Reactor Materials, 1966, KFK 120

/4/ B. Goel, report KFK 2234 (1975)

/5/ D. Woll, Card Image Format of the Karlsruhe Evaluated Nuclear Data File 1968, KFK 880 (1968)

/6/ Present Compendium, Chs. III.2, III.4

/7/ NEUDADA System Description, NEA/CCDN Newsletter CCDN/SYS-2, 1969

/8/ IBM SYSTEM/360, Operating System SORT/MERGE, Form C28-6543

/9/ IBM SYSTEM/360 Component Description IBM 2250 Display Unit Model 1, Form A27-2701-2

/10/ IBM SYSTEM/360 Operating System Supervisor and Data Management Services, Form GC28-6646

/11/ IBM SYSTEM/360 Operating System Linkage Editor, Form C28-6538

/12/ H. Alter and C.L. Dunford, Proc. 2nd Intern. Conf. on Nuclear Data for Reactors, Helsinki, 1970; Vol. II, p. 643

Appendix:

A1. List of changes in SCØRE routines

```
      MEMBER ONE - MAIN PROGRAM

./ DELETE SEQ1=01C036CC,SEQ2=01003600
      CATA TASK3     /'MAIN 3/3RESTART KEDAK   DATA FITENDF/B  RES CALC 01003600
./ NUMBER SEQ1=01004750,NEW1=01004800,INCR=100,INSERT=YES
      CALL LDFOPN(1,NDAT,&1000)
./ NUMBER SEQ1=01CC7CCC,NEW1=01007100,INCR=100,INSERT=YES
      CALL SELEC(IER)
      ICATA=IER
      GO TO (21,22),IER
C
   21 CALL KEDAK(IER)
      GO TO (30,10,10CC),IER
./ DELETE SEQ1=01CC73CC,SEQ2=01007300
   22 CALL SPDATA(IER)                                                  01007300
./ DELETE SEQ1=01CC95CC,SEQ2=01009500
   60 GO TO   (35,10,36,300,350,400,35,120,35,1000),INDM                01009500
./ DELETE SEQ1=01012900,SEQ2=01012900
  135 CALL PRIND(ICATA)                                                 01012900
./ NUMBER SEQ1=01C177CC,NEW1=01017800,INCR=100,INSERT=YES
C
C      KNDF OVERLAY
C
   36 IER=1
      IF(ICATA.EQ.1) GO TO 21
      CALL KNDF(IER)
      GO TO (425,1C,1CCC),IER


      MEMBER ONE - BLOCK DATA

./ DELETE SEQ1=01020500,SEQ2=01020500
      CATA VERDAT/'AUGUST  1972'/,ABCD/20*'    '/,NRECP,IMT/13,0/,       01020500


      MEMBER ONE - SUBROUTINE ERRCOR

./ NUMBER SEQ1=0105280C,NEW1=01052900,INCR=100,INSERT=YES
      CATA EFR13/'* NO DATA FOUND FOR                    ON KNDF
     8                '/
      IF(IEF.EQ.13) GO TO 11
      GO TO 12
   11 ERR13(6)=RTYPE
      CALL IDACON (ERR13(7),DUM1(14),2,1,1,1,IER)
   12 IER=1


      MEMBER FOUR - SUBROUTINE GRID

./ NUMBER SEQ1=0402580C,NEW1=04025900,INCR=100,INSERT=YES
      IF(JEXP.LT.10) GO TO 21
./ NUMBER SEQ1=0402580C,NEW1=04025900,INCR=100,INSERT=YES
      GO TO 22
   21 WRITE (99,23) BL,TEN,S,JEXP,BL
```

```
   23 FORMAT (A1,A4,A1,I1,A1)
   22 CALL INFILQ (A,I2)


      MEMBER FIVE - SUBROUTINE SPDATA

./ NUMBER SEQ1=050034CC,NEW1=05003500,INCR=100,INSERT=YES
      IDATA=IER
./ DELETE SEQ1=050044CC,SEQ2=050C6000
     2  'ATOMICS INTERNATIONAL AND IBM - PALO ALTO'////29X,'GFK - KARLSR05004400
     3UFE   VERSION III-K',5X,3A4)                                        05004500
C                                                                         05004600
      CALL IDENT (IER)                                                    05004700
      IF(IER.NE.1) GO TO 1000                                             05004800
./ DELETE SEQ1=050C8700,SEQ2=050C8700
   12 CALL CHOSZA (IZT,IAT,NISO,I,IDATA)                                  05008700
./ DELETE SEQ1=050233CC,SEQ2=05023300
   75 CALL CHOSRT (IRT,NRT,BCDRT,RTYPE,NREC,EL,IDATA)                     05023300


      MEMBER FIVE - SUBROUTINE CHOSZA

./ DELETE SEQ1=050471CC,SEQ2=05047100
      SUBROUTINE CHOSZA (IZ,IA,N,I,IDAT)                                  05047100
./ DELETE SEQ1=050483CC,05048300
      DIMENSION TITLE(9),TITLE1(12),TITLE2(8,3),IZ(1),IA(1),XKED(3),      05048300
./ NUMBER SEQ1=050484CC,NEW1=05048500,INCR=100,INSERT=YES
      REAL*8 NAME,SGIZ,XZ(1),R1,XB(1)
      EQUIVALENCE (TITLE1(9),NAME)
./ DELETE SEQ1=050485CC,SEQ2=05048500
      DATA  TITLE/'ISOTOPES AVAILABLE FROM NEUDADA ARE '/,TITLE1/'REACTI05048500
./ DELETE SEQ1=050487CC,SEQ2=05048700
     2X1,X4,Y1,Y4/2.,2.,10.75,2./,Y0,DY1,X5/9.5,.5,4./,X00,Y00/4.5,10./ 05048700
./ NUMBER SEQ1=05049CCC,NEW1=05049000,INCR=100,INSERT=YES
      DATA XKED/'KEDAK ARE    '/,BLANK/'     '/,SGIZ/'SGIZ     '/
C
      DIMENSION XEUD(3),TIT1(5)
      DATA XEUD/'NEUDADA ARE '/,TIT1/' Z =      A =      ARE'/
      DO 40 L=1,3
   40 TITLE(L+6)=XEUD(L)
      DO 50 L=1,5
   50 TITLE1(L+7)=TIT1(L)
./ NUMBER SEQ1=050494CC,NEW=05049500,INCR=100,INSERT=YES
      ENTRY CHOSMA (XZ,IA,N,I,IDAT)
      IST=(I-1)/15
   25 CALL INPICT
      IF(IDAT.EQ.2) GO TO 1
      TITLE(7)=XKED(1)
      TITLE(8)=XKED(2)
      TITLE(9)=XKED(3)
    1 CALL WRTXTM(2,36,TITLE,X1,Y1)
      IF(IDAT.EQ.1) GO TO 2
./ NUMBER SEQ1=05052050,NEW1=05052100,INCR=100,INSERT=YES
      GO TO 3
      DISPLAY MATERIAL NAMES FROM KEDAK
    2 Y2=Y0
      IL=IST*15+1
      IF (IL.LE.N) GO TO 4
      IL=1
```

```
      IST=0
    4 IU=IL+14
      IU=MINO(IU,N)
      NN=IU-IL+1
      CALL SETKEY (7,1,8)
      DO 10 K=IL,IU
      ISZ=1
      IF(I.EQ.K) ISZ=2
      NAME=XZ(K)
      TITLE1(8)=BLANK
      CALL WRTXTM(ISZ,8,TITLE1(9),X5,Y2)
   10 Y2=Y2-DY1
    3 CALL ENDKEY
./ NUMBER SEQ1=050552CC,NEW1=05055300,INCR=100,INSERT=YES
      IF(IDAT.EQ.1) GO TO 11
./ NUMBER SEQ1=050555CC,NEW1=05055600,INCR=100,INSERT=YES
   11 NAME=XZ(I)
      GO TO 1000
./ NUMBER SEQ1=050557CC,NEW1=05055800,INCR=100,INSERT=YES
      ENTRY CHOSRE (IR,NR,XB,R1,NREC,XX,IDAT)
./ NUMBER SEQ1=050582CC,NEW1=05058300,INCR=100,INSERT=YES
      L=4
      IF(IDAT.EQ.1) L=8
      CALL SETKEY (7,1,L)
./ NUMBER SEQ1=050585CC,NEW1=05058600,INCR=100,INSERT=YES
      NRE=16
      IF(IDAT.EQ.1) NRE=NR
./ DELETE SEQ1=05058800,SEQ2=05058800
      DO 250 I=1,NRE
./ NUMBER SEQ1=050593C0,NEW1=05059400,INCR=100,INSERT=YES
      IF(IDAT.EQ.1) GO TO 5
./ NUMBER SEQ1=050595C0,NEW1=05059600,INCR=100,INSERT=YES
    5 CALL WRTXTM (1,8,XB(I),X3,Y3)
      GO TO 6
./ NUMBER SEQ1=050614C0,NEW1=05061500,INCR=100,INSERT=YES
      IF(IDAT.EQ.1) GO TO 7
./ NUMBER SEQ1=050616CC,NEW1=05061700,INCR=100,INSERT=YES
    7 R1 = XB(I)
      GO TO 380
./ NUMBER SEQ1=050637C0,NEW1=05063800,INCR=100,INSERT=YES
      IF(R1.EQ.SGIZ) I2=3


      MEMBER SIX - SUBROUTINE PRIND

./ DELETE SEQ1=06060700,SEQ2=06060700
      SUBROUTINE PRIND (IDAT)                                    06060700
./ DELETE SEQ1=06065400,SEQ2=06065400
   30 IF(IDAT.EQ.1) GO TO 240                                    06065400
      WRITE (6,35)    ABCRE                                      06065500
./ DELETE SEQ1=06069000,SEQ2=06069000
  240 IF(NENDFB.EQ.0) GO TO 1000                                 06069000
      IF(IDAT.EQ.1) GO TO 270                                    06069100
./ NUMBER SEQ1=06069200,NEW1=06069200,INCR=100,INSERT=YES
      GO TO 260
  270 WRITE (6,280)
  280 FORMAT (//////30X,'KEDAK DATA'//)
  260 IT=1
```

```
      MEMBER EIGHT - SUBROUTINE SOND

./ DELETE SEQ1=C80306C0,SEQ2=C8030700
      CALL ROUND (ZX,1,5)                                                    08030600
      CALL ROUND (ZX,1,5)                                                    08030700



      MEMBER NAP - PROGRAM NAP

./ DELETE SEQ1=51701300,SEQ2=51701300
      COMMON     /BLK1/A(16,55),C(16),IV,ITSC,IR,IC,NPRINT,RECT(175),        51701300
./ NUMBER SEQ1=51702300,NEW1=51702400,INCR=100,INSERT=YES
      DATA BCON/'N'/
C
      DEFINE FILE 17(520,880,U,IV)
./ DELETE SEQ1=51702700,SEQ2=51702700
      READ (IN,500) DATX,NF,IFILE1,IDACS,IUPD,NPRINT                         51702700
./ DELETE SEQ1=51704900,SEQ2=51705900
./ DELETE SEQ1=51707800,SEQ2=51708200
      WRITE (17'1) NIS,NTS,(ISOTOP(I),ISNUM(I),DATEIS(1,I),DATEIS(2,I),      51707800
     1 NRECS(I),I=1,NIS),(TAPE(1,J),TAPE(2,J),J=1,NTS)                       51707900
./ DELETE SEQ1=51708800,SEQ2=51709100
    5 READ (17'1) NIS,NTS,(ISOTOP(I),ISNUM(I),DATEIS(1,I),DATEIS(2,I),       51708800
     1 NRECS(I),I=1,NIS),(TAPE(1,J),TAPE(2,J),J=1,NTS)                       51708900
./ NUMBER SEQ1=51712400,NEW1=51712500,INCR=100,INSERT=YES
      BLK=PAC(BT(7),BL,BL,BL)
./ DELETE SEQ1=51734800,SEQ2=51735100
      WRITE (17'1) NIS,NTS,(ISOTOP(I),ISNUM(I),DATEIS(1,I),DATEIS(2,I),      51734800
     1 NRECS(I),I=1,NIS),(TAPE(1,J),TAPE(2,J),J=1,NTS)                       51734900


      MEMBER NAP - SUBROUTINE PRINT

./ DELETE SEQ1=51741700,SEQ2=51741700
./ DELETE SEQ1=51745100,SEQ2=51745100
   38 IU=55                                                                  51745100


      MEMBER NAP - SUBROUTINE CREATE

./ DELETE SEQ1=51754000,SEQ2=51754000


      MEMBER NAP - SUBROUTINE RDISK

./ DELETE SEQ1=51756200,SEQ2=51756200


      MEMBER NAP - SUBROUTINE DPOINT

./ DELETE SEQ1=51757900,SEQ2=51760900

./ ENDUP
```

Appendix:

A2. Source program lists of the new subroutines

SELEC

IDENT

KNDF

RKNDF

TAPKND

SELK

RETXS

SKIPOI

KEDAK

ANERG

FILLTP

```
      SUBROUTINE SELEC (IOPT)
C
C     PERMITS THE SELECTION OF NEUDADA OR KEDAK DATA

C
      DIMENSION TEXT(14),TEXT1(2,2),TITLE(19)
      DATA Y1/5./,DY/.75/,X1/2./,TEXT1/'KEDA','K   ','NEUD','ADA '/
      DATA TITLE/'SCORE   CROSS SECTION EVALUATION SYSTEM ADAPTED TO NEU
     1DADA AND KEDAK DATA   '/
      DATA TEXT/'SELECT BY LIGHT-PEN DATA TYPE   TO BE RETRIEVED FIRST
     1 '/
C
      ICPT=0
      IC = 0
C
C     INITIALIZE DISPLAY AND WRITE TEXT
C
    1 CALL INPICT
      CALL WRTXTM (2,8,TITLE(1),5.,10.75)
      CALL WRTXTM (2,32,TITLE(3),2.,10.)
      CALL WRTXTM (2,36,TITLE(11),2.,9.25)
      CALL WRTXTM (2,32,TEXT(1),2.,7.75)
      CALL WRTXTM (2,24,TEXT(9),2.,7.)
      CALL SETKEY (7,1,8)
      Y=Y1
      DO 10 I=1,2

      ISZ=1
      IF(IO.EQ.I) ISZ=2
      CALL WRTXTM(ISZ,8,TEXT1(1,I),X1,Y)
   10 Y=Y-DY
      CALL ENDKEY
C
C     GENERATE DISPLAY AND AWAIT USER ACTION
C
      CALL STPICT
   20 CALL EXINGR (IKEY,IVAL,IOPT)
      GO TO (3,2,1,1000,20,20),IKEY
C
C     PROCESS USER ACTION AND RETURN
C
    3 IF(IVAL.GT.2) GO TO 20
      IF(IOPT.EQ.0) GO TO 1
      GO TO (1000,1),IVAL
    2 IF(IOPT.GT.2) GO TO 20
      IF(IO.LT.1.OR.IO.GT.2) GO TO 5
      CALL CHGCHR (IVAL,IO,1)
    5 CALL CHGCHR (IVAL,IOPT,2)
      IC=IOPT
      GO TO 20
 1000 RETURN
      END
```

```
C
      SUBROUTINE IDENT (IER)
C
      DIMENSION T1(4,4)
      COMMON/BLK1/ABCD(5,4),ICASE,CNUM,DUM1(43)
      DATA T1/'LABORATORY:      EVALUATOR:       DATE:            PROBLEM I
     1.D.:    '/
C
C     USER SUPPLIES IDENTIFICATION FOR PROBLEM
      NOUT=6
    1 CALL FIG1 (4,16,20,T1,ABCD,IKEY)
      CALL DACON (ABCD(1,4),CASE,5,1,1,0,2,IE)
      IF(IE.EQ.0) GO TO 2
      CALL ERRCOR (8,IER)
      GO TO (1,1000,1000),IER
    2 ICASE=100*IFIX(CASE+0.1)
      CALL IDACON (CNUM,ICASE,1,1,1,1,IE)
      WRITE (NOUT,101) ICASE
  101 FORMAT(1H1,//' CASE NUMBER',2X,I8//)
      DO 10 I=1,3
      WRITE (NOUT,102) (T1(J,I),J=1,4),(ABCD(K,I),K=1,5)
  102 FORMAT (1H0,5X,4A4,5X,5A4)
   10 CONTINUE
 1000 RETURN
      END
```

```
MEMBER NAME  KNDF
      SUBROUTINE KNDF (IER)                                              25400100
C                                                                        25400200
C     THIS IS THE CONTROL ROUTINE FOR THE HANDLING OF THE KNDF-DATA      25400300
C                                                                        25400400
      COMMON /BLK4/ DUM1(3),ENER,QX,NREC,NRECP,IZ,IAND,IZA,DUM2(5)       25400500
      COMMON /BLK5/ DUM5(4),XPLOTP(3),DUM6(7)                            25400600
      COMMON /BLK8/ NPOINT,X(150),Y(150),NKNDF,ZLIM(4)                   25400700
      DIMENSION NAMES(5)                                                 25400900
      EQUIVALENCE (Q,IQ)                                                 25401000
C                                                                        25401100
      IEF=0                                                              25401200
      IER=1                                                              25401300
      NPOINT=0                                                           25401400
C                                                                        25401500
      IF(NREC.GT.NRECP) GO TO 1                                          25401600
      EMIN=XPLOTP(1)*1.0E+06                                             25401700
      EMAX=XPLOTP(2)*1.0E+06                                             25401800
      GO TO 2                                                            25401900
C                                                                        25402000
    1 EMIN=ENER*1.0E+06                                                  25402100
      EMAX=0.                                                            25402200
    2 Q=QX*1.0E+06                                                       25402300
C                                                                        25402400
      CALL TAPKND  (NAMES,NZNAM,MM,IQ,IEF)                               25402500
      IF(IEF.NE.0) GO TO 55                                              25402600
C                                                                        25402700
    3 CALL RKNDF(NAMES,NZNAM,MM,EMIN,EMAX,X,Y,NPOINT,IEF)                25402800
      IF(IEF.NE.0) GO TO 55                                              25402900
C                                                                        25403000
C     CHANGE ENERGY UNIT FROM EV TO MEV                                  25403100
C                                                                        25403200
      IF(MM.EQ.4) GO TO 6                                                25403300
      DO 20 I=1,NPOINT                                                   25403400
   20 X(I)=X(I)*1.0E-06                                                  25403500
C                                                                        25403600
    6 IF(X(2).LT.1.000001*X(1).AND.X(2).GT.0.999999*X(1)) GO TO 5        25403700
      IF(Y(1).NE.0.0) GO TO 4                                            25403800
    5 DO 30 I=2,NPOINT                                                   25403900
      I1=I-1                                                             25404000
      X(I1)=X(I)                                                         25404100
   30 Y(I1)=Y(I)                                                         25404200
      NPOINT=NPOINT-1                                                    25404300
C                                                                        25404400
    4 CONTINUE                                                           25404500
C                                                                        25404600
      IF(X(1).LT.XPLOTP(1).AND.NREC.LE.NRECP) X(1)=XPLOTP(1)             25404700
 1000 NKNDF=NPOINT                                                       25404800
C                                                                        25404900
      CALL CALIMA (NPOINT,X,Y,ZLIM)                                      25405000
C                                                                        25405100
  100 RETURN                                                             25405200
   55 CALL ERRCOR (IEF,IER)                                              25405300
      GO TO 100                                                          25405400
      END                                                                25405500
C                                                                        25405600
      SUBROUTINE TAPKND (NAMES,NZNAM,MF,IQ,IEF)                          25405700
C                                                                        25405800
C     THIS ROUTINE CALCULATES THE KNDF-NAMES                             25405900
```

```
MEMBER NAME  KNDF
C      OF THE MATERIAL AND THE REACTION TYPE                                    25406000
C                                                                               25406100
       INTEGER DUM3                                                             25406200
       COMMON /BLK4/ DUM1(4),IQX,NREC,NRECP,IZ,IA,IZA,DUM3(4),RTYPE             25406300
       DIMENSION NAMES(5),NNAM(16),MTS(16),TYPNA(16)                            25406400
C                                                                               25406500
       DATA NNAM/11*2,1*3,1*2,1*3,1*4,1*0/                                      25406600
C                                                                               25406700
       DATA TYPNA/'SGAL','SGD ','SGF ','SGG ','SHE3','SGP ','SGH3','SGI '25406800
      2,'SG2N','SG3N','SGN ','SGIZ','SGT ','SGNC','SIGZ','    '/              25406900
C                                                                               25407000
       DATA MTS /107,104,19,102,106,103,105,4,16,17,2,5,1,2,5,29/               25407100
C                                                                               25407200
       DO 30 I=1,5                                                              25407300
    30 NAMES(I)=0                                                               25407400
       NAMES(1)=IZ*10000+IA                                                     25407500
       NAMES(3)=IQ                                                              25407600
C                                                                               25407700
C                                                                               25407800
       DO 20 I=1,16                                                             25407900
       MT=MTS(I)                                                                25408000
       ITY =I                                                                   25408100
       NZNAM=NNAM(I)                                                            25408200
       IF(NREC.EQ.1) GO TO 2                                                    25408300
C                                                                               25408400
    20 CONTINUE                                                                 25408500
       IEF=13                                                                   25408600
       GO TO 1000                                                               25408700
C                                                                               25408800
     4 MF=4                                                                     25408900
       GO TO 3                                                                  25409000
     2 IF(NREC.GT.NRECP) GO TO 4                                                25409100
       MF=3                                                                     25409200
     3 NAMES(2)=MF*10000+MT*10                                                  25409300
       IF(NREC.EQ.14.OR.NREC.EQ.15) NAMES(2)=NAMES(2)+2                         25409400
       RTYPE=TYPNA(ITY )                                                        25409500
       DUM3(4)=NAMES(1)                                                         25409600
       IEF=0                                                                    25409700
  1000 RETURN                                                                   25409800
C                                                                               25409900
       END                                                                      25410000
       SUBROUTINE RKNDF(NAMES,NZNAM,MF,EMIN,EMAX,X,Y,NPOINT,IEF)                25410100
C                                                                               25410200
C      THIS ROUTINE FILLS THE ARRAYS X AND Y WITH KNDF-DATA FROM THE            25410300
C      REQUIRED ENERGY RANGE                                                    25410400
C                                                                               25410500
       DIMENSION X(150),Y(150),NARG(3),NAMES(5),XX(150),YY(150)                 25410600
       DATA MAXNUM,EPS/150,0.01/                                                25410700
C                                                                               25410800
       IEF=0                                                                    25410900
       NOUT=6                                                                   25411000
       IF(MF.EQ.4) GO TO 400                                                    25411100
       IF(MF.EQ.3) GO TO 300                                                    25411200
       GO TO 1                                                                  25411300
  400  CALL SELK (NARG,NAMES,EMIN,EMAX,X,Y,MAXNUM,NPOINT,IEF)                  25411400
       GO TO 8                                                                  25411500
  300  NPOI1=0                                                                  25411600
       NARG(1)=NZNAM                                                            25411700
```

```
MEMBER NAME  KNDF
C                                                                       25411800
      CALL RETXS(NARG,NAMES,EMIN,EMAX,XX,YY,NUMX,MAXNUM,NR)             25411900
C                                                                       25412000
      I1=2                                                              25412100
   21 IF(NR.EQ.0) GO TO 11                                              25412200
      IF(NR.EQ.1) GO TO 12                                              25412300
      IF(NR.EQ.2) GO TO 13                                              25412400
      IF(NR.EQ.3) GO TO 1                                               25412500
      IF(NR.EQ.4) GO TO 1                                               25412600
      IF(NR.EQ.5) GO TO 1                                               25412700
      GO TO 16                                                          25412800
C                                                                       25412900
   11 I1=2                                                              25413000
      IF(NPOI1.NE.0) GO TO 18                                           25413100
   18 DO 10 I=1,NUMX                                                    25413200
      X(I)=XX(I)                                                        25413300
   10 Y(I)=YY(I)                                                        25413400
      NPOINT=NUMX                                                       25413500
      GO TO 17                                                          25413600
C                                                                       25413700
   19 CALL REPXS(NARG,NAMES,EMIN,EMAX,XX,YY,NUMX,MAXNUM,NR)             25413800
      GO TO 21                                                          25413900
C                                                                       25414000
   16 WRITE (NOUT,102) NR                                               25414100
  102 FORMAT(1X,'THE RETURN-CODE FROM RETXS = ',I10)                    25414200
      GO TO 1                                                           25414300
C                                                                       25414400
   12 IF(NPOI1.EQ.0) GO TO 18                                           25414500
   13 CALL SKIPOI(XX,YY,NUMX,X,Y,NPOI1,MAXNUM,EPS)                      25414600
      IF(NPOI1.LT.MAXNUM.AND.NR.EQ.2) GO TO 19                          25414700
      NPOINT=NPOI1                                                      25414800
      IF(NPOINT.LT.2) GO TO 1                                           25414900
C                                                                       25415000
C     INTERPOLATE CROSS SECTION TO LOWER ENERGY BOUNDARY                25415100
C                                                                       25415200
   17 IF(EMIN.GT.0.999999*X(1).AND.EMIN.LT.1.000001*X(1)) X(1)=EMIN     25415300
      IF(EMIN.LE.X(1)) GO TO 7                                          25415400
      CALL TRPL8(X(1),X(2),EMIN,I1,Y(1),Y(2),Z)                         25415500
      X(1)=EMIN                                                         25415600
      Y(1)=Z                                                            25415700
    7 IF(EMAX.EQ.0.0) GO TO 8                                           25415800
      IF(EMAX.GE.X(NPOINT))GO TO 8                                      25415900
C                                                                       25416000
C     INTERPOLATE CROSS SECTION TO UPPER ENERGY BOUNDARY                25416100
C                                                                       25416200
      CALL  TRPL8 (X(NPOINT-1),X(NPOINT),EMAX,I1,Y(NPOINT-1),Y(NPOINT),Z25416300
     1)                                                                 25416400
      X(NPOINT)=EMAX                                                    25416500
      Y(NPOINT)=Z                                                       25416600
      IEF=0                                                             25416700
      GO TO 8                                                           25416800
    1 IEF=13                                                            25416900
    8 RETURN                                                            25417000
      END                                                               25417100
      SUBROUTINE SELK (NARG,NAMEN,EMI1,EMA1,XX,YY,MAXNUM,NUM1,IEF)      25417200
C                                                                       25417300
C     NEUTRON ELASTIC SCATTERING ANGULAR DISTRIBUTIONS FROM KNDF        25417400
C                                                                       25417500
```

```
MEMBER NAME  KNDF
      DIMENSION XX(150),YY(150),NARG(3),NAMES(5),X1(101),Y1(101),X2(101)25417600
     1,Y2(101),XN(2),YN(2),F(10),NAMEN(1)                              25417700
      INTEGER*4 SGN/30020/,SGNC/40022/                                 25417800
      EQUIVALENCE(NAMES(3),XNAM)                                       25417900
C                                                                      25418000
      IEF=0                                                            25418100
      NOUT=6                                                           25418200
      MXN=101                                                          25418300
      NAMES(1)=NAMEN(1)                                                25418400
      NAMES(2)=SGN                                                     25418500
      NARG(1)=2                                                        25418600
      EMIN=EMI1                                                        25418700
C                                                                      25418800
      CALL RETXS(NARG,NAMES,EMIN,EMIN,XN,YN,NUMX,2,NR)                 25418900
      IF(NR.GT.2) GO TO 13                                             25419000
C                                                                      25419100
      IF(XN(1).EQ.EMIN) GO TO 11                                       25419200
      IF(XN(2).EQ.EMIN) GO TO 12                                       25419300
      CALL TRPL8(XN(1),XN(2),EMIN,2,YN(1),YN(2),Z)                     25419400
      WSGN=Z                                                           25419500
      GO TO 14                                                         25419600
C                                                                      25419700
   11 WSGN=YN(1)                                                       25419800
      GO TO 14                                                         25419900
C                                                                      25420000
   12 WSGN=YN(2)                                                       25420100
   14 ESGN=EMIN                                                        25420200
      WRITE (NOUT,105) ESGN,WSGN                                       25420300
C                                                                      25420400
      IGO=0                                                            25420500
      NAMES(1)=NAMEN(1)                                                25420600
      NAMES(2)=SGNC                                                    25420700
      XNAM=0.                                                          25420800
      NARG(1)=3                                                        25420900
    7 CALL IDFLOC (NERR,NARG,NAMES,F)                                  25421000
      IF(XNAM.EQ.0.) GO TO 19                                          25421100
C                                                                      25421200
      NAMES(1)=NAMEN(1)                                                25421400
      NAMES(2)=SGNC                                                    25421500
      NARG(1)=3                                                        25421600
      EMAX=1.                                                          25421700
      EMIN=-1.                                                         25421800
      E1=XNAM                                                          25421900
      XNAM1=XNAM                                                       25422000
      CALL RETXS(NARG,NAMES,EMIN,EMAX,XX,YY,NUMX,MAXNUM,NR)            25422100
      IF(NR.GT.2) GO TO 23                                             25422200
  105 FORMAT(1X,10E13.5)                                               25422400
      LEVEL=1                                                          25422500
      NUM1=NUMX                                                        25422600
      XNAM2=XNAM                                                       25422700
      E2=XNAM                                                          25422800
C                                                                      25422900
   15 IF(E1.LT.ESGN) GO TO 16                                          25423000
      IF(E1.GT.ESGN) GO TO 19                                          25423100
      LEVEL=2                                                          25423200
      IF(IGO.NE.0) GO TO 18                                            25423300
      IEF=0                                                            25423400
      GO TO 1000                                                       25423500
```

```
MEMBER NAME  KNDF
C                                                                       25423600
   13 WRITE (NOUT,101) NAMES(1),EMIN                                    25423700
  101 FORMAT(1X,' FOR ',I10,' SGN, EMIN= ',E16.8,' NO DATA FOUND ON KNDF25423800
     1')                                                                25423900
      GO TO 19                                                          25424000
   23 WRITE (NOUT,102) NAMES(1),E1                                      25424100
  102 FORMAT(1X,' FOR',I10,' SGNC LEVEL ENERGY= ',E16.8,' NO DATA FOUND 25424200
     1ON KNDF')                                                         25424300
   19 IEF=13                                                            25424400
      GO TO 1000                                                        25424500
C                                                                       25424600
   16 IF(E2.LT.ESGN) GO TO 17                                           25424700
      IF(E2.GT.ESGN) GO TO 18                                           25424800
   17 CALL IDFNXT(NERR,NARG,NAMES,F)                                    25424900
      IF(NERR.NE.0) GO TO 17                                            25425000
      IF(XNAM2.EQ.XNAM) GO TO 18                                        25425100
      XNAM1=XNAM2                                                       25425200
      E1=E2                                                             25425300
      XNAM2=XNAM                                                        25425400
      E2=XNAM                                                           25425500
      IGC=1                                                             25425600
      GO TO 15                                                          25425700
C                                                                       25425800
   18 XNAM=XNAM1                                                        25425900
      NAMES(1)=NAMEN(1)                                                 25426000
      NAMES(2)=SGNC                                                     25426100
      CALL RETXS(NARG,NAMES,EMIN,EMAX,X1,Y1,NUMX,MXN,NR)                25426200
      IF(NR.GT.2) GO TO 23                                              25426300
      NUM2=NUMX                                                         25426500
      NAMES(1)=NAMEN(1)                                                 25426600
      NAMES(2)=SGNC                                                     25426700
      XNAM=XNAM2                                                        25426800
      IF(LEVEL.EQ.2) GO TO 24                                           25426900
      CALL RETXS (NARG,NAMES,EMIN,EMAX,X2,Y2,NUMX,MXN,NR)               25427000
      IF (NR.GT.2) GO TO 21                                             25427100
      LEVEL=3                                                           25427300
      NUM3=NUMX                                                         25427400
C                                                                       25427500
   24 NUM1=NUM2-1                                                       25427600
      DO 10 I=1,NUM1                                                    25427700
      DO 20 J=1,NUM1                                                    25427800
      IF(X1(J).LT.XX(I).AND.X1(J+1).GT.XX(I)) GO TO 25                  25427900
      IF(X1(J).EQ.XX(I)) GO TO 32                                       25428000
   20 CONTINUE                                                          25428100
      IF(X1(NUM2).LE.XX(I)) GO TO 21                                    25428200
      GO TO 29                                                          25428300
   32 YY(I)=Y1(J)                                                       25428400
      GO TO 10                                                          25428500
C                                                                       25428600
   21 J=NUM1                                                            25428700
   25 CALL TRPLB (X1(J),X1(J+1),XX(I),2,Y1(J),Y1(J+1),YY(I))            25428800
   10 CONTINUE                                                          25428900
      IF(LEVEL.EQ.2) GO TO 31                                           25429000
C                                                                       25429100
   26 NUM1=NUM3-1                                                       25429200
      DO 30 I=1,NUM1                                                    25429300
      DO 40 J=1,NUM1                                                    25429400
      IF (X2(J).LT.XX(I).AND.X2(J+1).GT.XX(I)) GO TO 27                 25429500
```

```
MEMBER NAME  KNDF
      IF(Z(1).LE.EMIN) GOTO 5                                          25435400
      IF(Z(1).GE.EMAX) GOTO36                                          25435500
   11 I=I+1                                                            25435600
      X(I)=W(1)                                                        25435700
      Y(I)=W(2)                                                        25435800
   12 I=I+1                                                            25435900
      X(I)=Z(1)                                                        25436000
      Y(I)=Z(2)                                                        25436100
      GOTO NST,(20,200)                                                25436200
   20 CALL IDFNXT(NERR,NARG,NAMES,Z)                                   25436300
      IF(NERR.EQ.0) GOTO 22                                            25436400
      IF(Z(1).GE.EMAX) GOTO 24                                         25436500
      IF(I.EQ.MAXNUM) GOTO 34                                          25436600
   21 I=I+1                                                            25436700
      X(I)=Z(1)                                                        25436800
      Y(I)=Z(2)                                                        25436900
      GOTO 20                                                          25437000
C                                                                      25437100
      ENTRY REPXS(NARG,NAMES,EMIN,EMAX,X,Y,NUMX,MAXNUM,NR)             25437200
      I=0                                                              25437300
      NAMZ=NARG(1)                                                     25437400
      IF(NAMZ.LE.2) GOTO 21                                            25437500
      DO 19 J=3,NAMZ                                                   25437600
   19 NAMSV(J-2)=NAMES(J)                                              25437700
      GOTO 21                                                          25437800
C                                                                      25437900
   22 IF(I.LT.1) GOTO 23                                               25438000
      NR=1                                                             25438100
      GOTO 198                                                         25438200
   23 NR=5                                                             25438300
      I=1                                                              25438400
      X(I)=Z(1)                                                        25438500
      Y(I)=Z(2)                                                        25438600
      GOTO 198                                                         25438700
C                                                                      25438800
   24 IF(I.EQ.MAXNUM) GOTO 34                                          25438900
      I=I+1                                                            25439000
      X(I)=Z(1)                                                        25439100
      Y(I)=Z(2)                                                        25439200
      NR=0                                                             25439300
      GOTO 200                                                         25439400
   26 I=1                                                              25439500
      X(I)=W(1)                                                        25439600
      Y(I)=W(2)                                                        25439700
      NR=5                                                             25439800
      GOTO 198                                                         25439900
C                                                                      25440000
   30 NR=3                                                             25440100
      GOTO 200                                                         25440200
C                                                                      25440300
   32 NR=4                                                             25440400
      I=1                                                              25440500
      X(1)=Z(1)                                                        25440600
      Y(1)=Z(2)                                                        25440700
      GOTO 200                                                         25440800
C                                                                      25440900
   34 NR=2                                                             25441000
      GOTO 200                                                         25441100
```

```
MEMBER NAME  KNDF
   36 ASSIGN 200 TO NST                                                  25441200
      NR=0                                                               25441300
      GOTO 7                                                             25441400
   38 ASSIGN 200 TO NST                                                  25441500
      NR=0                                                               25441600
      GOTO 11                                                            25441700
  196 IF(NAMZ.LE.2) GOTO 200                                             25441800
      DO 199 J=3,NAMZ                                                    25441900
  199 NAMES(J)=NAMSV(J-2)                                                25442000
  200 NUMX=I                                                             25442100
      RETURN                                                             25442200
      END                                                                25442300
      SUBROUTINE SKIPOI(XX,YY,NUMX,X,Y,NPOINT,LENX,EPS)                  25442400
      DIMENSION XX(1),YY(1),X(1),Y(1)                                    25442500
C                                                                        25442600
C         SKIPOI SKIPS POINTS NOT NECESSARY TO REPRESENT CURVE TO AN     25442700
C         ACCURACY OF EPS.                                               25442800
C                                                                        25442900
      J=1                                                                25443000
      I=0                                                                25443100
      IF(NPOINT.EQ.0) GOTO 10                                            25443200
      I=NPOINT                                                           25443300
      IF(NUMX.GT.1) GOTO 11                                              25443400
   10 I=I+1                                                              25443500
      IF(I.GT.LENX) GOTO 99                                              25443600
      X(I)=XX(J)                                                         25443700
      Y(I)=YY(J)                                                         25443800
      IF(J.GE.NUMX) GOTO 100                                             25443900
   11 IF(J+1.GE.NUMX) GOTO 98                                            25444000
      JA=J+1                                                             25444100
      JB=J+2                                                             25444200
      DO 30 JK=JB,NUMX                                                   25444300
      XK=(YY(JK)-Y(I))/(XX(JK)-X(I))                                     25444400
      JC=JK-1                                                            25444500
      IS=0                                                               25444600
      DO 20 JJ=JA,JC                                                     25444700
      YI=XK*(XX(JJ)-X(I))+Y(I)                                           25444800
      IF(ABS(YI-YY(JJ)).LT.EPS*YY(JJ)) GOTO 18                           25444900
      GOTO 22                                                            25445000
   18 IF(ABS(YI-YY(JJ)).GT.1.E-5*YY(JJ)) GOTO 20                         25445100
      IS=IS+1                                                            25445200
   20 CONTINUE                                                           25445300
      JA=JA+IS                                                           25445400
      GOTO 30                                                            25445500
   22 J=JC                                                               25445600
      GOTO 10                                                            25445700
   30 CONTINUE                                                           25445800
      J=NUMX                                                             25445900
      GOTO 10                                                            25446000
   98 J=J+1                                                              25446100
      GOTO 10                                                            25446200
   99 I=I-1                                                              25446300
  100 NPOINT=I                                                           25446400
      RETURN                                                             25446500
      END                                                                25446600
      SUBROUTINE KEDAK (IER)                                             25446700
C                                                                        25446800
C     RETRIEVES KEDAK DATA FOR SCORE                                     25446900
```

```
MEMBER NAME  KNDF
C                                                                       25447000
      COMMON /LDFMT/ IDUM(4),MAT(70),IMA(70),DUM(105),NMAT              25447100
      COMMON /BLK1/ ABCD(5,4),ICASE,ISHIFT,E(5),REAC(10),BGTIL(18),U(10)25447200
      COMMON /BLK2/ NULL,P(1500),II,DU(1004)                            25447300
      COMMON /BLK3/ NR,BCDR(5,30),TCHAR(30),ISEL(30)                    25447400
      COMMON /BLK4/ DUM1(4),QX,ID(5),IUM3(4),RTYPE                      25447500
      COMMON /BLK5/ LI(4),XY3(6),XY4(4)                                 25447600
      COMMON /BLK6/ NUL1,XYS(90),NUL2                                   25447700
      COMMON /BLK7/ NUL3,XY(300),NUL4,NUL5,YX(300),NUL6                 25447800
      COMMON /BLK8/ NPOINT,X(150),Y(150),NKNDF,ZLIM(4)                  25447900
      COMMON /BLK13/ NEAD,EAD(15),GAMD(15),NPE(16)                      25448000
C                                                                       25448100
      REAL*8 MAT,MATER,TYPES(70),RTYP,NAM1(2)                           25448200
      DIMENSION ITYP(70),IR(70),XX(3),NAMES(5),IAT(70),NDAT(70),REA(10) 25448300
      DIMENSION XMAT(5),DUMC(2),BGTIL2(18),XNAM1(200)                   25448400
      DATA IR/70*2/,BLANK/'    '/                                       25448500
      EQUIVALENCE (REA(1),RTYP),(XMAT(1),MATER),(NAMES(3),XNAM)         25448600
C                                                                       25448700
      NULL = 0                                                          25448800
      NUL1=0                                                            25448900
      NUL2=0                                                            25449000
      NUL3=0                                                            25449100
      NUL4=0                                                            25449200
      NUL5=0                                                            25449300
      NUL6=0                                                            25449400
      NZ=2                                                              25449500
      NPOINT=0                                                          25449600
      NOUT=6                                                            25449700
      IDAT=IER                                                          25449800
      IER=1                                                             25449900
      ID(1)=1                                                           25450000
      NR=0                                                              25450100
      DO 120 I=1,10                                                     25450200
      REA(I)=BLANK                                                      25450300
  120 U(I)=BLANK                                                        25450400
      DO 130 I=1,18                                                     25450500
      BGTIL2(I)=BLANK                                                   25450600
  130 BGTIL(I)=BLANK                                                    25450700
      DO 10 J=1,30                                                      25450800
      DO 10 I=1,5                                                       25450900
      XMAT(I)=BLANK                                                     25451000
   10 BCDR(I,J)=BLANK                                                   25451100
C                                                                       25451200
C     USER IDENTIFICATION                                               25451300
C                                                                       25451400
      CALL IDENT(IER)                                                   25451500
      IF(IER.NE.1) GO TO 1000                                           25451600
    1 IMAT=0                                                            25451700
C                                                                       25451800
C     SELECT ISOTPE                                                     25451900
C                                                                       25452000
      CALL CHOSMA(MAT,IAT,NMAT,IMAT,IDAT)                               25452100
      IF(IMAT.LT.1.OR.IMAT.GT.70) GO TO 1                               25452200
      NAMES(1)=IMA(IMAT)                                                25452300
      IUM3(4)=NAMES(1)                                                  25452400
      MATER=MAT(IMAT)                                                   25452500
      IATER=IMA(IMAT)                                                   25452600
      ID(3)  =IATER/10000                                               25452700
```

```
MEMBER NAME  KNDF
      WRITE (NOUT,101) MATER                                              25452800
  101 FORMAT (1H0,5X,A3)                                                  25452900
C                                                                         25453000
C     FILL LIST WITH REACTION TYPES FOR SELECTED ISOTOPE                  25453100
C                                                                         25453200
      CALL FILLTP (NER,MATER,TYPES,ITYP,NTYP)                             25453300
      IF(NER.NE.1) GO TO 33                                               25453400
      IF(NTYP.EQ.0) GO TO 1                                               25453500
      XX(3)=0.                                                            25453600
C                                                                         25453700
C     CHOOSE REACTION TYPE                                                25453800
C                                                                         25453900
      CALL CHOSRE (IR,NTYP,TYPES,RTYP,NREC,XX,ICAT)                       25454000
      IF(NREC.LT.1.OR.NREC.GT.32) GO TO 1                                 25454100
      RTYPE=RTYP                                                          25454200
      ITYPE=ITYP(NREC)                                                    25454300
      NAMES(2)=ITYPE                                                      25454400
      MM=ITYPE/10000                                                      25454500
      IF(MM.EQ.3) GO TO 11                                                25454600
      IF(ITYPE.EQ.40022) GO TO 12                                         25454700
      GO TO 1000                                                          25454800
C                                                                         25454900
   12 ID(1)=14                                                            25455000
   11 EMIN=XX(1)*1.0E+6                                                   25455100
      EMAX=XX(2)*1.0E+6                                                   25455200
      XNAM=XX(3)*1.0E+6                                                   25455300
      DUM1(1)=EMIN                                                        25455400
      DUM1(2)=EMAX                                                        25455500
      DUM1(3)=XNAM                                                        25455600
C                                                                         25455700
      IF(ITYPE.EQ.40022.OR.ITYPE.EQ.30050) GO TO 13                       25455800
      GO TO 200                                                           25455900
C                                                                         25456000
   13 NAM1(1)=MATER                                                       25456100
      NAM1(2)=RTYP                                                        25456200
      MAX=200                                                             25456300
      NZ=3                                                                25456400
C                                                                         25456500
      CALL LOFNAM (NER,NAM1,XNAM1,N1,N2,MAX,2)                            25456600
C                                                                         25456700
      IF(NER.NE.1) GO TO 33                                               25456800
C                                                                         25456900
      CALL ANERG(NAM1,N2,XNAM1,NNAM,IER)                                  25457000
C                                                                         25457100
      IF (IER.NE.0) GO TO 55                                              25457200
      IER=1                                                               25457300
      IF (ITYPE.EQ.40022) GO TO 110                                       25457400
      QX=XNAM1(NNAM)*1.0E-6                                               25457500
      XNAM=XNAM1(NNAM)                                                    25457600
      DUM1(3)=XNAM                                                        25457700
      GO TO 200                                                           25457800
  110 EMIN=XNAM1(NNAM)                                                    25457900
      DUM1(1)=EMIN                                                        25458000
      DUM1(3)=EMIN                                                        25458100
      NEAD=1                                                              25458200
      EAD(1)=EMIN                                                         25458300
C                                                                         25458400
C     RETRIEVE CROSS SECTION FROM KEDAK                                   25458500
```

```
MEMBER NAME  KNDF
C                                                                       25458600
  200 CALL RKNDF (NAMES,NZ,MM,EMIN,EMAX,X,Y,NPOINT,IEF)                 25458700
      IF(IEF.NE.0) GO TO 55                                             25458800
      IF (MM.EQ.4) GO TO 6                                              25458900
C                                                                       25459000
      DO 20 I=1,NPOINT                                                  25459100
   20 X(I)=X(I)*1.E-6                                                   25459200
    3 IF(X(2).LT.1.000001*X(1).AND.X(2).GT.0.999999*X(1)) GO TO 5       25459300
      IF(Y(1).NE.0.0) GO TO 4                                           25459400
C                                                                       25459500
    5 DO 30 I=2,NPOINT                                                  25459600
      I1=I-1                                                            25459700
      X(I1)=X(I)                                                        25459800
   30 Y(I1)=Y(I)                                                        25459900
      NPOINT=NPOINT-1                                                   25460000
    4 CONTINUE                                                          25460100
      NKNDF=NPOINT                                                      25460200
C                                                                       25460300
      CALL CALIMA(NPOINT,X,Y,ZLIM)                                      25460400
C                                                                       25460500
      DO 70 I=1,4                                                       25460600
      XY4(I)=ZLIM(I)                                                    25460700
   70 DL(I)=ZLIM(I)                                                     25460800
C                                                                       25460900
      DO 40 I=1,10                                                      25461000
      BGTIL(I+6)=REA(I)                                                 25461100
   40 REAC(I)=REA(I)                                                    25461200
      DO 50 I=1,5                                                       25461300
      BGTIL(I)=XMAT(I)                                                  25461400
   50 E(I)=XMAT(I)                                                      25461500
      ICASE=ICASE+1                                                     25461600
      CALL IDACON (DUMC,ICASE,2,1,1,1,IE)                               25461700
      CNUM=DUMC(2)                                                      25461800
      ICNUM=17                                                          25461900
      BGTIL(17)=CNUM                                                    25462000
      DO 90 I=1,16                                                      25462100
   90 BGTIL2(I)=BGTIL(I)                                                25462200
      CALL CENTER(BGTIL2,16,BGTIL,ISHIFT)                               25462300
      ISHIFT= ISHIFT+4*(ICNUM-1)+1                                      25462400
C                                                                       25462500
 1000 RETURN                                                            25462600
   55 CALL ERROR(IEF,IER)                                               25462700
      GO TO 1000                                                        25462800
   33 IEF=13                                                            25462900
      GO TO 55                                                          25463000
      END                                                               25463100
C                                                                       25463200
      SUBROUTINE FILLTP(NR,MAT,TYP,ITYP,NT)                             25463300
C                                                                       25463400
      REAL*8 TYPS(32)/'SGT','SGN','SGX','SGI','SGIZC','SGIZ','SG2N','SG3225463500
     1N','SGIA','SGI3A','SG2NA','SG3NA','SG1P','SGNI','SGA','SGF','SGG',25463600
     2'SGP','SGD','SGH3','SGHE3','SGALP','SG2HE','SGTR','MUEL','ETA','AL25463700
     3PHA','NUE','NUEP','CHIF','CHIFD','SGNC'/,TYP(70),MAT,NAMES(3)     25463800
      INTEGER NP(70),MTYP/70/,MTYPS/32/,NARG(3)                         25463900
      DIMENSION ITYP(70)                                                25464000
      COMMON/LDFTT/MC(3)                                                25464100
C                                                                       25464200
C          FILL TABLE OF SINGLE VALUED,ENERGYDEPENDENT TYPES.           25464300
```

```
MEMBER NAME  KNDF
      KOUT=6                                                                25464400
      CALL LDFITN(NR,MAT,TYP,NP,NT,MTYP,2)                                  25464500
      IF(NR.EQ.0.AND.NT.EQ.0) GOTO 100                                      25464600
      IMAT=MO(3)                                                            25464700
      CALL LDFITN(NR,IMAT,ITYP,NP,NT,MTYP,1)                                25464800
      IF(NR.EQ.0.AND.NT.EQ.0) GOTO 99                                       25464900
      J=0                                                                   25465000
      DO 12 N=1,NT                                                          25465100
      DO 10 K=1,MTYPS                                                       25465200
      IF(TYP(N).NE.TYPS(K)) GOTO 10                                         25465300
      J=J+1                                                                 25465400
      TYP(J)=TYP(N)                                                         25465500
      ITYP(J)=ITYP(N)                                                       25465600
      NP(J)=NP(N)                                                           25465700
      GOTO 12                                                               25465800
   10 CONTINUE                                                              25465900
   12 CONTINUE                                                              25466000
      NT=J                                                                  25466100
      GOTO 100                                                              25466200
   99 WRITE(KOUT,600)                                                       25466300
  600 FORMAT(///'   PROGRAMMING-ERROR IN LDFITN.CALL PROGRAMMER.')          25466400
      STOP                                                                  25466500
  100 CONTINUE                                                              25466600
      RETURN                                                                25466700
      END                                                                   25466800
C                                                                           25466900
      SUBROUTINE ANERG (NAM1,N2,XNAM1,NNAM,IER)                             25467000
C                                                                           25467100
C     PERMITS THE SELECTION OF THE LEVEL ENERGY FOR INELASTIC SCATTERING25467200
C     OR THE INCIDENT ENERGY FOR ELASTIC SCATTERING                         25467300
C                                                                           25467400
      REAL*8NAM1(2)                                                         25467500
      DIMENSION XNAM1(1),BCD(3)                                             25467600
      DATA PAGE/'PAGE'/,Y1,X00,Y00,DY1/10.75,4.5,10.,.5/                    25467700
C                                                                           25467800
      I=0                                                                   25467900
      NNAM=0                                                                25468000
      IST=0                                                                 25468100
C                                                                           25468200
  100 CALL INPICT                                                           25468300
      CALL WRTXTM (2,16,NAM1, 3.,Y1)                                        25468400
      X3=X00                                                                25468500
      Y3=Y00                                                                25468600
      IL=15*IST+1                                                           25468700
      IF(IL.LT.N2) GO TO 111                                                25468800
      IL=1                                                                  25468900
      IST=0                                                                 25469000
  111 IU=IL+14                                                              25469100
      IL=MINO(IU,N2)                                                        25469200
      NN=IU-IL+1                                                            25469300
C                                                                           25469400
      CALL SETKEY (7,1,12)                                                  25469500
      DO 10 K=IL,IU                                                         25469600
      ISZ=1                                                                 25469700
      IF (I.EQ.K) ISZ=2                                                     25469800
      DNUM=XNAM1(K)                                                         25469900
      CALL DACON (BCD,DNUM,3,1,1,6,1,IER)                                   25470000
      IF(IER.NE.0) GO TO 1000                                               25470100
```

```
MEMBER NAME  KNDF
      CALL WRTXTM (1SZ,12,BCD,X3,Y3)                                    25470200
   10 Y3=Y3-DY1                                                         25470300
      CALL ENDKEY                                                       25470400
C                                                                       25470500
      CALL SETKEY (1,1,4)                                               25470600
      CALL WRTXTM (2,4,PAGE,5.5,1.)                                     25470700
      CALL ENDKEY                                                       25470800
C                                                                       25470900
      CALL STPICT                                                       25471000
  150 CALL EXINGR(IKEY,IVAL,ISEQ)                                       25471100
C                                                                       25471200
      GO TO (120,200,120,120,150,150),IKEY                              25471300
  200 IF (IVAL.EQ.2) GO TO  110                                         25471400
      IF (ISEQ.GT.NN)GO TO  150                                         25471500
      IF (I.LT.IL.OR.I.GT.IU) GO TO 105                                 25471600
      ICH=I-IL+1                                                        25471700
      CALL CHGCHR (1,ICH,1)                                             25471800
  105 CALL CHGCHR (1,ISEQ,2)                                            25471900
      I=IL+ISEQ-1                                                       25472000
      GO TO 150                                                         25472100
  110 IST=IST+1                                                         25472200
      GO TO 100                                                         25472300
C                                                                       25472400
  120 NNAM=I                                                            25472500
 1000 RETURN                                                            25472600
      END                                                               25472700
```

A3. Input Summary for Karlsruhe Version of SCØRE

Input from the IBM 2250 terminal.

Note: END key means key no. 1,
CANCEL key means key no. 2 of the programmed-function keyboard.

Figure 1:

(1) Chose between options KEDAK and NEUDADA by the light pen.
(2) Press END key.

Figure 2: User and case identification

(1) Type in user identification, date and running number of problem (case identification) from keyboard.

(2) Press END key.

Figure 3: List of isotope names

(1) Selection of PAGE with light pen turns page
(2) Select isotope with light pen
(2) Press END key

Figure 4: List of reaction types

(1) Select reaction type with light pen
(2) Type in energy limits from keyboard
(3) Press END key

Figure 5: List of energies for SGNC or SGIZ (only after selection of option KEDAK in Figure 1)

(1) Selection of PAGE by light pen turns page
(2) Select energy with light pen
(3) press END key

Figure 6: Option lists MAIN 1/3, 2/3, 3/3

Options can be chosen by light pen or from programmed-function keyboard.
Notation used:

k nn : programmed-function key no. nn,
N : option valid for NEUDADA only,
K : option valid for KEDAK only,
N+K : option valid for both NEUDADA and KEDAK,

where NEUDADA and KEDAK refer to the choice made in Figure 1.

Option list MAIN 1/3

N+K	MAIN 1/3	k 11	Display option list MAIN 2/3.
N+K	RESTART	k 12	Jump back to start of program, Figure 1.
N+K	PRINT	k 13	Print data displayed on screen.
N	LETTERS(PØINTS)	k 14	Replace curve points by letters distinguishing references (or not).
N	HATS(NØ HATS)	k 15	Add error bars (or not).
N+K	EXPAND	k 16	Expand the graph portion of the display to full screen size, omitting caption, references and option lists. Pressing of the END key restores the unexpanded picture.
N	NØ CURVE	k 17	Delete curves of evaluated data.
N	LISTDATA	k 18	Tabulate energy and cross section values (plus references), 20 data points at one time on the screen. Choosing one of the numbers appearing in the right-

			most column under the + (or -) sign by light pen causes forward (or backward) shifting of the list by so many energies. Light pen selection of (+) produces (-). Pressing of the END key terminates the LISTDATA option.
N+K	QUIT	k 20	End terminal session.

Option list MAIN 2/3

N+K	MAIN 2/3	k 11	Display option list MAIN 3/3.
N+K	RESTART	k 12	Jump back to start, Figure 1.
N + K	REPLØT	k 13	Change co-ordinate system.
N	AUTØ LIM	k 14	Recalculate plot scale with modified data (after CØRRECT or DELETE).
N	SØRT	k 15	Delete data points belonging to the references selected from the following Figure 7.
N	DELETE	k 17	Delete points indicated with light pen on expanded graph (curve points being represented by letters associated with references). Deletion of the last point can be canceled with the CANCEL key. Pressing of the END key terminates the DELETE option.
N	RESTØRE	k 18	Restore all points previously deleted.
N	CØRRECT	k 19	Earmark points indicated by light pen for correction. Erroneous assignment can be canceled with the CANCEL key. After selecting all points to be corrected one gets them listed upon pressing the END key in Figure 8.

N+K	QUIT	k 20	End terminal session.

Option list MAIN 3/3

N+K	MAIN 3/3	k 11	Display option list MAIN 1/3.
N+K	RESTART	k 12	Jump back to start, Figure 1.
N	KEDAK	k 13	Display KEDAK curve together with experimental data (for option NEUDADA from Figure 1).
K	KEDAK	k 13	Jump back to Figure 2: Treat next case (for option KEDAK from Figure 1).
N	DATA FIT	k 14	Call module for data fit.
N	ENDF/B	k 15	Display ENDF/B curve together with experimental data.
N	RES CALC	k 16	Call module for resonance fits.
N	NØ CURVE	k 18	Delete curves or points. Requested option must be indicated by light pen and is then displayed in large size. Cancellation possible with CANCEL key. Option is terminated with END key.
N+K	QUIT	k 20	End terminal session.

Figure 7: List of references

This figure appears after the SØRT option has been chosen from option list MAIN 2/3.

If more than 13 references exist for the selected data type and energy range one can turn the page by pressing the END key. Points from the references indicated by light pen are discarded, the selected references are displayed in large size. They can be restored by pointing to them once more. One can also type in

a reduced energy interval and get a corresponding new co-ordinate system by pointing to 'YES' with the light pen.

Figure 8: List of uncorrected and corrected data

This figure tabulating the first 15 points to be corrected appears after the CØRRECT option has been chosen from option list MAIN 2/3. One can indicate after LIGHT PEN DATA whether a value is to be multiplied by a factor or increased (decreased) by a given amount which then must be typed in. The values (energies or cross sections) to be corrected in this way, can then be indicated by light pen. Erroneous selection can be canceled with the CANCEL key. Alternatively one can set a cursor on the energy value of the incorrect point (right-hand pair of columns) and then type in the correct values. After completion of all corrections in the right-hand columns one must press the END key to get a display of the next 15 points earmarked for correction, etc., until eventually the corrected plot appears on the screen.

DATA FIT: Module for fitting non-resonance data

For this module there are two option lists,
DATA 1/2 and 2/2.

Option list DATA 1/2

N	DATA 1/2	k 11	Display option list DATA 2/2.
N	LETTERS(POINTS)	k 12	Replace curve points by letters distinguishing references (or not).
N	HATS(NO HATS)	k 13	Add error bars (or not).
N	EXPAND	k 14	Expand the graph portion of the display to full screen size, cf. option list MAIN 1/3.
N	CØMPARE	k 15	Show curves fitted with different methods for comparison.
N	AUTO LIM	k 16	Recalculate plot scale with modified data (after CØRRECT or DELETE).

N	PRINT	k 17	Print fitted data.
N	LEG CØEF	k 18	Analyse differential angular distributions in terms of Legendre coefficients. END key is to be pressed after completion.
N	NØ CURVE	k 19	Delete curve or points. Requested option must be indicated by light pen and is then displayed in large size. Cancellation possible with CANCEL key. Option is terminated with END key.
N	MAIN ØPT	k 20	Jump back to main program SCØRE.

Option list DATA 2/2

N	DATA 2/2	k 11	Display option list DATA 1/2.
N	CHG NØDE	k 12	Nodes can be added, deleted or moved. On the expanded graph the curve points are shown with error bars together with a tracking pattern (3 concentric circles) which facilitates manipulation of nodes. Three programmed function keys can be activated:
	(ADD NØDE)	k 11	Add node with light pen. Each time the END key is pressed the center of the tracking pattern becomes a node of the fit curve, represented by an asterisk. Up to 30 node points can be generated. Pressing of the CANCEL key terminates this option.
	(REMØVE)	k 12	Delete nodes with light pen. Pressing of the CANCEL key terminates this option.

	(MØVE)	k 13	Move a point: The point is indicated by light pen whereupon it is replaced by the tracking pattern which can be moved around with the light pen. Upon pressing of the END key the center is replaced by an asterisk which fixes a node. Pressing of the CANCEL key terminates this option.
N	COMPARE	k 15	Show curves fitted with different methods for comparison.
N	LINEAR	k 16	Interpolate linearly between nodes.
N	SPLINE	k 17	Interpolate with cubic spline between nodes.
N	AUTØ SPL	k 18	Interpolate with cubic spline determined by least-squares method.
N	NØ CURVE	k 19	Delete curve or points. Requested option must be indicated by light pen and is then displayed in large size. Cancellation is possible with CANCEL key. Option is terminated with END key.
N	MAIN OPT	k 20	Jump back to main program SCØRE.

RES_CALC: Module for resonance fitting

For evaluation in the resonance region three resonance formalisms are available:

Breit-Wigner single-level formalism,
Reich-Moore multi-level formalism,
Adler-Adler multi-level formalism.

The fit module for the resonance region is called by selection of the option RES CALC in option list MAIN 3/3. A series of special figures is displayed:

Figure 1: Selection of resonance formalism

(1) Select one of the three available formalisms by light pen.

Figure 2: User information

(1) Any time a new resonance formalism is selected Doppler kernel, resolution function and sample composition must be respecified.
(2) The selected options are displayed in large size.
(3) After each selection the END key must be pressed.

Figure 3: Input of Doppler-broadening parameters

(1) Select Doppler kernel with the light pen.
(2) The chosen kernel is displayed in large size together with a list of parameters needed.
(3) Type in the appropriate parameters.
(4) Press END key.

Figure 4: Input of resolution-broadening parameters

(1) Select resolution function with the light pen.
(2) The chosen resolution function is displayed in large size together with a list of parameters needed.
(3) Type in the appropriate parameters.
(4) Press END key.

Figure 5: Specification of isotopic mixture

(1) Select isotopes by light pen from a list of available isotopes. The chosen isotope is displayed in large size.
(2) A list of needed parameters is shown. Type in the appropriate values.
(3) By pointing with the light pen to DELETE the user can delete the selected isotope.
(4) Press END key to terminate input for the isotope.
(5) Renewed pressing of the END key calls Figure 6.

Figure 6:

(1) Selection of YES by light pen calls Figure 7.
(2) Selection of NØ by light pen terminates input of new isotopes.

Figure 7: Isotope identification

(1) Type in identification parameters for the isotope.
(2) Press END key. If resonance parameters for the isotope are already in the library Figure 5 reappears, otherwise Figure 8.

Figure 8: Resonance parameter input

(1) Type in resonance parameters
(2) After completing input for one resonance press END key. Figure 8 reappears for the next resonance. Up to 10 new resonances can be entered.
(3) Press CANCEL key to terminate resonance parameter input.

Two options exist for modification of resonance parameters during the calculation without changing them in the disc library:

- ADD RES permits addition of new parameters
- MOD RES permits modification of parameters in core storage.

Both options start with Figure 9.

Figure 9: Selection of isotope for resonance modification

(1) A list of isotopes in core storage is displayed allowing light-pen selection of those isotopes for which resonance parameters are to be added or modified.
(2) Selection of ALL ZA means that changes are to be made for all isotopes.
(3) Press END key after selection is complete.

If ADD RES was chosen, Figure 8 is displayed for each isotope selected for modification.

If RES MØD was chosen, Fig. 10 appears with a list of resonance energies.

Figure 10: Input of modified resonance parameters (RES MØD option)

(1) Select resonance to be modified with the light pen. A picture for the chosen resonance similar to Figure 8 is displayed. Parameters must be typed in. After completion press END key and modify next resonance etc. After last modification press END key twice to terminate RES MØD option.

Figure 11:

This figure appears after selection of option E SHIFT from option list RES 3/3. Type in the letter identifying the reference for which the energy scale is to be modified, and also the coefficients c_1 and c_2 of the modification formula (see below). After pressing of the END key corrected energies E_c are calculated from the uncorrected energies E according to

$$E_c = \frac{c_1}{(1+c_2\sqrt{E})^2} E.$$

The E SHIFT option is terminated with the CANCEL key.

For module RES CALC there are three option lists,
RES 1/3, 2/3 and 3/3.

Option list RES 1/3

MODEL	Display Figure 1.
CHANGES	Display Figure 2.
ADD RES	Display Figure 9 for isotope selection, then Figure 8 for addition of resonances.
MØD RES	Display Figure 9 for isotope selection, then Figure 10 for resonance modification.
LØAD RES	Load selected parameters into core memory.
CØMPARE	Display results of preceding and present calculation for comparison.
FIT	Apply iterative algorithm for resonance parameter adjustment thrice.
CALC	Calculate with selected resonance formalism and display resulting curve (no fit).
MAIN OPT	Terminate RES CALC option, jump back to main program SCØRE.

Option list RES 2/3

LETTERS (POINTS) Represent experimental points by the letters distinguishing references (by dots).

HATS(NO HATS) Add error bars (or not).

EXPAND Expand the graph portion of the display to full screen size, cf. option list MAIN 1/3.

STØRE RES Update direct-access library on disc (after completion of fit procedure).

AUTØ LIM Recalculate plot scale with modified data.

CØMPARE Display results of preceding and present calculation simultaneously for comparison.

FIT Apply iterative algorithm for resonance parameter adjustment thrice.

CALC Calculate with selected resonance formalism and display resulting curve (no fit).

MAIN ØPT Terminate option RES CALC, jump back to main program SCØRE.

Option list RES 3/3

P PØINTS Set "peak points" on expanded graph. Three points per resonance must be specified for the fit by means of the programmed-function keys:

- k 1 END key.
- k 2 CANCEL key.
- k 3 Delete all points.
- k 4 Delete previously selected points.
- k 11 Go 1 point further for selection of next point.
- k 12 Go 2 points further for selection of next point.
- k 13 Go 5 points further for selection of next point.
- k 14 Go 10 points further for selection of next point.
- k 15 Go to first point
- k 16 Add or modify "peak points" with light pen and tracking pattern. Select one point at the peak, press END key, select a point on the left-hand slope, press END key, select a point on the right-hand slope, press END key. Three such points are required per resonance.

V PØINTS	Additional "valley points" can be set on the expanded graph to improve the fit in the valleys between resonances. Use programmed-function keys in the same way as for P PØINTS.
E SHIFT	Display Figure 11 to enable energy scale for points from a given reference to be shifted.
NØ CURVE	Delete curves
CØMPARE	Display results of preceding and present calculation for comparison.
FIT	Apply iterative algorithm for parameter adjustment thrice.
CALC	Calculate with selected resonance formalism and display curve.
MAIN ØPT	Terminate RES CALC option, jump back to main program SCØRE.

5. PROGRAMS FOR ESTIMATION OF NUCLEAR-MODEL AND CROSS-SECTION PARAMETERS

A good evaluation of cross section data is not possible without utilization of nuclear models and nuclear reaction theory. Codes which serve this purpose are briefly described in the following sections. Since these codes are essentially independent of any evaluated-data format and most of them can be used as stand-alone programs they do not belong, strictly speaking, to the KEDAK software proper. To a certain degree, however, they reflect the methods employed in KEDAK evaluation work. Therefore, and because the programs as such may interest potential users elsewhere, program abstracts were included in the present compendium.

5.1 Multi-level shape analysis program for transmission data (FANAL)

Name of program:

FANAL2

Name and establishment of author:

F.H. Fröhner
INR
Kernforschungszentrum
D-7500 Karlsruhe, West Germany

Problem solved:

Resonance parameter extraction from several sets of high-resolution time-of-flight transmission data simultaneously.

Method of solution:

Multi-level shape analysis with two-channel Reich-Moore formalism. Picket-fence approximation for distant levels. Resolution broadening. Doppler broadening of narrow ("p-, d-..wave") levels, but not of s-wave levels (the code was written for structural materials). Iterative least-squares fit. Resulting fit is plotted after each step.

Restrictions:

Doppler broadening of s-wave levels is neglected (but not of "p-, d-...wave" levels, cf. above). Up to 5 transmission data sets with a total of not more than 5120 data points can be fitted simultaneously. The number of cross section parameters (channel radii, distant-level strangth functions, resonance energies and partial widths) must not exceed 200 (corresponding to about 45 resonances) of which not more than 50 can be adjusted. The sample material must not contain more than 5 different nuclides.

Unusual features:

Card input is organized for maximum user convenience (no flags, no bookkeeping information), modular structure, no tricky programming.

Related and auxiliary programs:

Non-KFK users must replace the Karlsruhe plotter subroutine PLØTA following the instructions given in the program.

Programming language:

FØRTRAN IV

Computer:

IBM/360-65,/360-91,/370-168 etc.

Machine requirements:

260 kbytes of core storage, card reader, printer, plotter.

Typical running time:

1-20 minutes on IBM/370-168 under OS, depending on number of data points, number of parameters adjusted, quality of first guesses etc.

References

F.H. Fröhner, report KFK 2129 (1976)

5.2 Multi-level shape analysis program for capture data (FANAC)

Name of program:

FANAC

Name and establishment of author:

F.H. Fröhner
INR
Kernforschungszentrum
D-7500 Karlsruhe, West Germany

Problem solved:

Resonance parameter extraction from several sets of high-resolution time-of-flight capture data simultaneously.

Method solved:

Multi-level shape analysis with two-channel Reich-Moore formalism. Picket-fence approximation for distant levels. Resolution broadening. Doppler broadening of narrow ("p-, d-...wave") levels, but not of s-wave levels(the code was written for structural materials). Multiple-collision events are treated by Monte-Carlo simulation. Iterative least-squares fit. Resulting fit is plotted after each step.

Restrictions:

Doppler broadening of s-wave levels is neglected (but not of "p-, d-...wave" levels, cf. above). Up to 5 capture data sets with a total of not more than 512 data points can be fitted simultanously. The number of cross section parameters (channel radii, distant-level strangth functions, resonance energies and partial widths) must not exceed 200 (corresponding to about 45 resonances) of which not more than 20 can be adjusted. The sample material must not contain more than 5 different nuclides.

Unusual features:

Inportance sampling and Russian roulette are used in the Monte Carlo calculations to increase efficiency and to avoid bias. Card input is organized for maximum user convenience (no flags, no bookkeeping information). Modular structure, no tricky programming.

Related and auxiliary programs:

Non-KFK users must replace the Karlsruhe plotter subroutine PLØTA following the instructions given in the program. Non-IBM users must replace the random-number generator RANDU by an equivalent subroutine.

Programming language:

FØRTRAN IV

Computer:

IBM/360-65,/360-91,/370-168 etc.

Machine requirements:

458 kbytes of core storage, card reader, printer, plotter.

Typical running time:

2-20 minutes on IBM/370-168 under OS, depending on number of data points, number of parameters adjusted, quality of first guesses etc.

References:

F.H. Fröhner, Proc. 4th Conf. on Nuclear Cross Sections and Technology, Washington D.C., 1975, Vol. 2, p. 929

5.3 Estimation of level-statistical parameters from individual resonance parameters (STARA)

Name of program:

STARA

Name and establishment of author:

F.H. Fröhner
INR
Kernforschungszentrum
D-7500 Karlsruhe, West Germany

Problem solved:

Estimation of level-statistical cross section parameters (average level spacing, average reduced neutron width, strength function, average radiation width) from individual resonance parameters (level energies, neutron widths, radiation widths), with due account of missing levels.

Method of solution:

Maximum-likelihood estimation of mean spacing and average reduced neutron width simultaneously, based on the "crystalline" regularity of level spacings (Dyson and Mehta) and the Porter-Thomas distribution of reduced neutron widths (per channel). The coupled system of 2 maximum-likelihood equations is solved by iteration. Uncertainties are also estimated. Results are plotted.

Restrictions:

The maximum number of resonances is 1000.

Unusual features:

No estimate of the observability threshold or its energy dependence is required, in contrast to the widely used Fuketa-Harvey method.

Related and auxiliary programs:

Non-KFK users must replace the Karlsruhe plotter subroutine PLØTA following the instructions given in the program.

Programming language:

FØRTRAN IV

Computer:

IBM/360-65,/360-91,/370-168 etc.

Machine requirements:

234 kbytes of core storage, card reader, printer, plotter.

Typical running time:

5-10 s per resonance series (on IBM/370-168 under OS)

References:

-

5.4 Estimation of level-statistical parameters from average cross sections (FITACS)

Name of program:

FITACS

Name and establishment of author:

F.H. Fröhner
INR
Kernforschungszentrum
D-7500 Karlsruhe, West Germany

Problem solved:

Estimation of level-statistical cross section parameters (average spacings, strength functions, average radiation widths) from average total and capture cross sections.

Method of solution:

Least-squares fit to average total, or average total and capture cross section data simultaneously, with due account of inelastic scattering. Data are weighted

according to uncertainty and/or point density. Average cross sections are calculated with Hauser-Feshbach theory including width fluctuations. Results are plotted.

Restrictions:

Energy dependence of strength function is neglected and only s-, p-, d-wave interactions are considered. Fission cannot be handled by present code version. This restricts applicability to energies up to 100 - 200 keV and excludes fitting of partial cross sections of fissile nuclei.Up to 200 total and up to 200 capture cross section values can be fitted. Only pure nuclides can be handled.

Unusual features:

As the code was written mainly for the interpretation of evaluated average cross section data weighting of points can be chosen according to point density (i.e. inversely propertional to energy interval "covered" by a given point) as an alternative to the more common weighting by inverse variance.

Related and auxiliary programs:

Non-KFK users must replace the Karlsruhe plotter subroutine PLØTA following the instructions given in the program.

Programming language:

FØRTRAN IV

Computer:

IBM/360-65,/360-91,/370-168 etc.

Machine requirements:

102 kbytes of core storage, card reader, printer, plotter.

Typical running times:

About 1 s per nuclide on IBM/370-168 under OS.

References:

-

5.5 INCH3 - a coupled-channels program for inelastic and elastic exchange reactions with automatic parameter optimization

INCH3 is a modified version of the program INCH written by A.D. Hill and D.J. Edens.

Name of program:

INCH3

Name and establishment of author:

B. Goel
INR
Kernforschungszentrum
D-7500 Karlsruhe, West Germany

Problem solved:

Estimation of interaction parameters of coupled-channels problem by fitting angle-integrated and differential elastic and inelastic cross section and polarization data at a given incident energy.

Method of solution:

Weighted least-squares fit. Numerical integration of the coupled-channels equations is performed with the full interaction out to a cut-off radius R_1 for nuclear forces. Beyond R_1 only long-range Coulomb terms (proportional to r^{-L-1}) are used up to a radius R_2 where the asymptotic series expansions of the Coulomb wave functions are valid and can be matched.

Restrictions:

Antisymmetrization between incoming particle and target nucleons is neglected and also excitation of the scattered particle. The interaction potential must be local and velocity-independent. No restrictions (other than practicability) limit the number of coupled channels, the spin of the scattered particle or the number of excited target-nuclear states.

Unusual features:

Standard potentials (Coulomb, Saxon-Woods, derivative Saxon-Woods, centrifugal potential) are available as options, Modifications are possible in subroutine NUCLIN following author's instructions /1/.

Related and auxiliary programs:

-

Programming language:

FØRTRAN IV

Computer:

ATLAS, CDC 6600, IBM/370-168

Machine requirements:

240 kbytes of core storage

Typical running time:

40 sec on IBM/370-168 for a parameter optimisation problem of 10 search cycles

References:

/1/ Unpublished description available from author.

5.6 DWBA: Program calculating inelastic-scattering and charge-exchange cross sections

Name of program:

DWBA

Name and establishment of author:

B. Goel
INR
Kernforschungszentrum
D-7500 Karlsruhe, West Germany

Problem solved:

Calculation of differential cross sections for inelastic scattering and charge exchange reactions such as those for (n,p) or (^{3}He,t) reactions.

Method of solution:

Distorted-wave Born approximation. Numerical integration of Schrödinger equation. A combination of Saxon-Woods form and its derivative for real and imaginary nuclear potential is used. Calculated and experimental data can be plotted together (line printer plot).

Restrictions:

Only angular distribution is calculated.

Unusual features:

A formalism for the calculation of knock-on processes is also included.

Related and auxiliary programs:

-

Programming language:

FØRTRAN IV

Computer:

ATLAS, IBM/370-168

Machine requirements:

128 kbytes of core storage

Typical running time:

about 1 min per problem or angular distribution on IBM/370-168

Reference:

B. Goel, Thesis, University of Freiburg i.Br., 1970

5.7 SECDIST: Program for calculation of energy distributions of secondary neutrons produced by inelastic scattering

Name of program:

SECDIST

Names and establishment of authors:

I. Broeders, C. Broeders
INR
Kernforschungszentrum
D-7500 Karlsruhe, West Germany

Problem solved:

Calculation of angle-integrated energy spectra of secondary neutrons produced by inelastic scattering with account of equilibrium and pre-equilibrium processes /1/, /2/. Comparison with experimental data.

Method of solution:

Equilibrium processes are calculated using a special version of the Hauser-Feshbach expression for continuous channels, which is obtained by angle integration of the corresponding expression used in the computer code HELENE /3/.

Preequilibrium reactions are calculated using Blann's theory /4/ and modified version of Blann's code /5/. Absorption cross sections and transmission coefficients needed for the equilibrium and preequilibrium calculations are obtained by the nonlocal optical model developped by Perey and Buck /6/ and a modified version of their code /7/.
Experimental data for energy distributions of secondary neutrons are often given as functions of the scattering angle. In order to obtain quantities that are comparable with the computer results the angular distributions are fitted and extrapolated by squares of spherical Bessel functions or by Legendre polynomials and integrated.

Restrictions:

Only elastic and inelastic neutron scattering channels are considered in the Hauser-Feshbach formula.

Unusual features:

Absorption cross sections and transmission coefficients which are needed for the Hauser Feshbach and for the preequilibrium calculations are obtained by the nonlocal-optical-model code /6,7/ and transferred within the system by densely stored tables. The transfer of two-dimensional tables (cross sections versus energy or energy distributions of secondary neutrons versus energy) is performed according to PLOTEASY /8/ formats. In this way automatic plotting of these quantities is made possible. In order to minimize user input the NAMELIST input option is used by all modules.

Related and auxiliary programs:

SECDIST consists of the subprograms ØPTMØD, PREEQ, SIGMIX and FITMØD and utilizes the plot package PLØTEASY /8/.

Programming language:

IBM FØRTRAN IV

Computer:

IMB/360 and /370 series (e.g. IBM 370 model 168). Plotting devices: IBM-1627, CALCOMP-Plotter, STATOS-Plotter, SYNETICS-Plotter, CRT-display at Karlsruhe.
The program is executed under ØS 370.

Machine requirements:

One disk is needed for transferring data. Without overlay all modules can be run with 122 k core storage with one exception: OPTMOD needs 200 k.

Typical running time:

As example the CPU times on a IBM 370/168 are given for the calculations for 14 MeV neutrons on Fe^{56} /1,2/

1. Preparation of the optical model data with the code OPTMOD ≈ 5 min 30 sec CPU-Time
2. Preequilibrium calculation with PREEQ 2 sec.
3. Equilibrium and composite data with SIGMIX 5 sec.
4. Fitting of experimental data 5 - 13 sec.

References:

/1/ H. Jahn, Contributed paper to the IAEA consultants meeting on the use of nuclear theory in neutron nuclear data evaluation, Trieste 8-12 December 75.

/2/ H. Borgwaldt, C.H.M. Broeders, I. Broeders, H. Jahn, M. Lalović, D. Rusch, Reaktortagung Nürnberg, 1975.

/3/ S.K. Penny, "HELENE", ORNL-TM-2590 (1969)

/4/ M. Blann, Phys, Rev. Lett. 28 (1972), 757 and Nucl. Phys. A213, (1973), 570

/5/ M. Blann, COO_3494-14

/6/ F.G. Perey, B. Buck, Nucl. Phys. 32, (1962), 353

/7/ F.G. Perey, ORNL-3429

/8/ C.H.M. Broeders, PLOTEASY , (1976) unpublished