KFKI-1991-15/G

Z. SZATMÁRY

USER'S MANUAL OF PROGRAM RFIT

Part 3. The data files

Hungarian Academy of Sciences
CENTRAL
RESEARCH
INSTITUTE FOR
PHYSICS

B U D A P E S T

KFKI-1991-15/G
PREPRINT

USER'S MANUAL OF PROGRAM RFIT

Part 3. The data files

Z. Szatmáry

Central Research Institute for Physics
H-1525 Budapest 114, P.O.B. 49, Hungary

HU ISSN 0368 5330

Z. Szatmáry: User's manual of program RFIT Part 3. The data files. KFKI-1991-15/G

ABSTRACT

The present Part 3 of the user's manual of program RFIT is devoted to a detailed description of all files of experimental data:

1. PDF - file of raw experimental data,
2. EDF - file of evaluated data,
3. CLIB - file of calibration factors.

In relation to these files, it is described:

- the information stored;
- structure;
- rules of input preparation for including new data;
- rules of correcting the stored data;
- options available for manipulating the file (e.g. listing, deleting, copying, restoring the whole file or parts of it).

Finally, it is discussed how the data stored in the files can be evaluated.

Chapter 2 treats the physical contents of PDF while Chapters 3 and 4 specify the input formats for new and corrected PDF data sets, respectively. Chapter 5 is devoted to subfile operations. The evaluation of the data stored in PDF is discussed in Chapter 6. Files EDF and CLIB are treated in Chapter 7.

З. Сатмари: Справочник по использованию программы RFIT. 3. том: Файлы данных. KFKI-1991-15/G

АННОТАЦИЯ

Настоящий, третий том справочника по использованию программы RFIT дает подробное изложение файлов данных, а именно:

1. PDF - файл первичных экспериментальных данных,
2. EDF - файл оцененных данных,
3. CLIB - файл калибровочных факторов.

В отношении каждого файла излагается:

- характер данных, хранимых в файле,
- структура файла,
- задание новых данных,
- исправление имеющихся в файле данных,
- операции различных видов, как, напр., распечатка, копирование, вычеркивание хранимых данных и т.п.
- обработка хранимых в файле данных.

Szatmáry Z.: Az RFIT program felhasználói kézikönyve. 3. rész. Az adatfile ok. KFKI 1991 15/G

KIVONAT

Az RFIT program felhasználói kézikönyvének jelen, harmadik része a következő adatfile okat tárgyalja részletesen:

1. PDF - a primér kísérleti adatok file ja,
2. EDF - a kiértékelt adatok file ja,
3. CLIB - a kalibrációs tényezők file ja

Mindegyik file ra vonatkozóan leírjuk:

- a tárolt információt;
- a file szerkezetét;
- az új adatok megadásának a módját;
- a tárolt adatok javítását;
- a file lal végzett különböző műveleteket (listázást, törlést, másolást, a sérült file visszaállítását stb.);
- a tárolt adatok kiértékelését.

Table of contents

Part 3. The data files

1. Introductory remarks on storing experimental data

Since 1973, experiments have been performed at the ZR-6 zero power reactor by an international team of physicists called TIC (Temporary International Collective) with the participation of ten countries. The aim of the measurements was to provide basic reactor physics data for light water moderated low enriched fuel lattices in order to validate reactor codes and group-constant libraries. The ZR-6 critical assembly, the experiments and their results are published in a special series of ZR-6 reports and summarized in Ref. [1]. In addition to this, the experimental results have been documented in several computerized data storage systems. The purpose of the present part of this user's manual is to describe them. It contains all information given in Refs. [2] and [3] completed with the developments introduced in program RFIT since their publication.

The largest and most important member of this data storage system is the file of raw experimental data referred to in this user's manual as PDF (abbreviation derived from its previous name Primary Data File). This file is a careful and complete documentation of not only the raw experimental results but also of all the additional information showing as truly as possible how the individual experiments have been done. A file structure which is able to satisfy the needs of every type of measurement is rather complex and the quantity of the data stored in it is substantial. Therefore, some explanation is necessary why it was needed. The main reason of recurring to the use of such a sophisticated system follows mainly from the international character of the work carried out by TIC: the individual experimenters participating in the common experimental program do their measurements not only for their own country but also for the whole community of TIC. At the very beginning of the experimental program, it turned out that special efforts and means are necessary for making all experimental results available for all parties of TIC. Now, PDF was meant primarily as a tool of storing and disseminating the experimental results obtained in the framework of this joint program.

Why just the raw experimental data are stored, this makes further explanations necessary since one can object that the above

mentioned series of ZR-6 reports containing the evaluated results of the measurements would do almost the same as PDF does if the problem was only the documentation and dissemination of the experimental information. This is unquestionably true and this way of data documention is also applied: volumes 1 and 3 of the Final Report of TIC (see Ref. [1]) are edited just for this purpose. However, the very first year of the international cooperation within TIC has shown that TIC cannot be managed without storing also the raw experimental data. In order to make this point clear, let us consider the following two examples.

Extensive experiments have been carried out for the determination of deep subcritical reactivities with the help of the pulsed neutron source method. They have been evaluated according to methods like those of Sjostrand, Gozani and others, consequently, the original task has been done. The measured decay curves, however, contain some further, not less useful information: decay constants of the fundamental and higher harmonics which, when completed by a detailed information concerning the reactor state in which they were determined, lend themselves for further analysis. If someone wants to perform such a work, he will look for the raw experimental data. This is in fact not a hypothetical case: a real interest has been formulated in extrapolating the above mentioned decay constants to the critical states but, unfortunately, it turned out that the raw data of these measurements have not been recorded carefully enough. Hence such an analysis would be possible only after repeating the whole series of measurements (which is no more possible for financial reasons).

The other example concerns the determination of the material buckling from macroscopic distributions of different reaction rates (or in a not too correct expression: on the basis of measured macrofluxes). As discussed in Section 1.2.5 of Part 2, this is based on the asymptotic parts of the measured distributions where they can be fitted by cosine or Besselian functions. It is natural that the experimenters tend to throw away the non-asymptotic parts of their macrofluxes (for what they cannot be blamed very much). However, the non-asymptotic parts of the measured macrofluxes are valuable for those who wish to validate their programs computing neutron fluxes in reactors consisting of different material regions (e.g. core and reflector). Such diverging needs can be best satisfied by keeping the raw experimental data and thus allowing everybody to treat and evaluate them as he (or she) likes.

One could continue with further examples demonstrating the following everyday experience: all measurements contain much more information than the experimenter originally desired to obtain. That is the main reason of storing the raw experimental data. In additon to this, there are other points of view, too, which are summarized as follows:

- Only the raw data represent the full experimental information. We always lose information by data evaluation.
- Only the raw data may be considered as experimental facts. Evaluated data always contain some subjective judgment.

- The details of the experimental method can be understood only when the raw data are available.
- We restrict the possibility of any reevaluation of the measurements if the raw data are not kept.
- The effort and money spent for measuring the data are too much for allowing to throw away any peace of them.

The total number of raw experimental data to be stored has reached several millions by the end of 1990, consequently, only a computerized data storage comes into question. There is, however, a more decisive argument favouring such a system: the data evaluation is done by a computer anyhow whence any other way of storing the data would lead to complications. Originally, the programs handling the data storage system and fitting program RFIT were separate programs. Later, they were united into one program but the structure of PDF remained practically the same.

The data storage sytem has two further members: the library of calibration factors (referred to as CLIB) and EDF mentioned above already. Both data files are much smaller in size than PDF. They are also handled by program RFIT and they are described in Chapter 7. CLIB is a complementer file to PDF: it contains the characteristics of the activation detectors used in the measurements. In this sense, it is necessary for evaluating the data stored in PDF. EDF contains some results obtained by program RFIT. On the other hand, the data stored in EDF can be input data of further evaluations (also done by program RFIT). All these relations and connections are made clear in the present part of this user's manual. It is worthwhile to note that most of the data published in Ref. [1] are stored in EDF.

2. PDF: the file of raw experimental data

The file of raw experimental data (PDF) contains the results of the measurements just as they were recorded i.e. without any data handling. It is subdivided into subfiles. As a matter of fact, it is hard to give a general definition of what a subfile is since, as it will become clear from the following, the experimenters have a relatively large freedom in defining subfiles. Therefore, we state only the guideline: a subfile normally corresponds to one single measurement performed for a well defined state of the reactor. In case of a macroflux measurement for example, one subfile contains, as a rule, all data which correspond to one irradiation. There may be exceptions: it may sometimes be expedient to split the data belonging to the same irradiation into several subfiles. But the opposite may not be the case: the data measured for several irradiations must be stored in separate subfiles. In case of reactivity measurements, the definition of the subfiles depends on the consideration of the experimenter to a larger extent. Further details of this problem can be discussed only in terms of the subfile structure. Therefore, we stop this line of reasoning for the time being.

Who describes the subfile, describes PDF as well, consequently, the first thing to do is to discuss the structure of the subfile. Before preceeding to this, it is worthwhile to note that the name "subfile" used to be a technical term at the ICL-1905 computer at which PDF was originally created. Although it has no such significance at IBM computers where program RFIT is running now, this expression has been kept and is used to designate a natural unit of the experimental information stored in PDF.

2.1. Subfile identification

We need 12 characters for an unambiguous identification of the subfile. They are the following:

- the first 4 characters define the type of the measurement,
- the next 8 characters give the date when the measurement was

performed, and
- the last 2 characters are used for distinguishing between subfiles for which the first 10 characters would be identical.

The notations which may be used for specifying the type of the measurement are summarized in Table 2.1. No other notation will be interpreted as a subfile identification by program RFIT. The date is specified by giving the day, month and year using 2 characters for each. (As to the year, only the last two digits are given.) For example,

MA03120575

identifies a subfile corresponding to a radial macroflux measurement (cf. Table 2.1) performed on 12 May 1975. In cases when several MA03 type measurements are performed on the same day, additional characters are needed, e.g.

MA03120575-1
MA03120575-2
etc.

As this example shows, the last 2 characters of the subfile identifiers are optional. When they are given, any characters may be chosen for them.

It follows from what was told above that the subfile identifiers hold some useful information concerning the measurement. The indication of the type of measurements will be of great help for program RFIT when the data stored in the subfile are evaluated (or just interpreted). The date can be a rather useful tool for subfile selection (see Chapter 5).

2.2. The structure of a subfile

2.2.1. Generalities on the subfile structure

The following main goals were pursued with the actual subfile structure:

- not to lose any peace of information,
- to store only the relevant information in PDF,
- to make data evaluation flexible, and
- to help the work of experimenters as much as possible.

There are two further, maybe less trivial but nonetheless important points of view:

- it must be easy to complement existing subfile with additional information and
- it must be possible to develop the subfile structure (i.e. to extend it with new elements of information).

These last two points require that one must have the possibility to develop PDF itself and to complement any subfiles in it. This, however, may not have any influence on the information recorded previously. This means in plain words that experimenters who are not interested in the new developments could afford not to take notice of them: what they could do ten years ago, they must be able to do it now and that exactly in the same way.

The easiest way of explaining the subfile structure is by considering it a questionary: the information to be stored in the subfile is divided into groups and each group is considered as an answer to a definite question. The questions are identified by integer numbers ranging from 1 to 99. Both in this user's manual and in the program output, reference is made consequently to the question numbers. The questions which are defined for the program version described now are listed in Table 2.2 and their contents will be explained in the next two sections. But, prior to this, we have to discuss some further generalities.

As to the type of information required, the questions can be diveded into two classes: those which require verbal and those which require numeric answers. Question numbers 1 to 60 refer to the former while those from 61 to 99 refer to the latter class. (Questions 5, 7, 8, 9, and 22 are exceptions to this: they contain half numeric, half verbal information, see below.) There is a difference of principle between these classes. When the data stored in the subfile are evaluated, the program can work only with numbers i.e. only the second class is relevant for data evaluation. In this respect, the verbal answers bear only auxiliary information. It would be a mistake, however, to think that one can manage without verbal information. All experimenters know that in however beautiful numeric tables their experimental results are presented, the latter are useless if their copybook notes are lost where such things have been recorded as:

- which detectors have been used,
- which their calibration factors are,
- what was the position of these detectors within the reactor,
- which measurements are repetitions of others and which should be considered as independent ones,
- how the apparatus used was set, etc.

Such a kind of information is needed only when something goes wrong with the evaluation or when someone wants to reevaluate the subfile. That is why the structure of PDF subfiles allows to record text information, too. From the experimenters' point of view, providing for verbal information is not only a courtesy for the eventual ultimate users of their data but it is their own interest, too: what they have at hand in the moment of performing the measurement, it may seem trivial for them but it may turn out to be indispensable several years later when everything will have been lost and forgotten and when only the subfile will be available.

Questions 9, 63, and 78 are marked by asterisks below and in Table 2.2. They play an important role in specifying the subfile structure in that they are necessary for the formulation of other

questions. Consequently, the subfile usually cannot be created if any one of them is omitted. Questions 79 and 80 are also marked since the subfile cannot be evaluated if both are omitted. These interdependences will become clear from the explanations given in Sections 2.2.2 and 2.2.3.

There is an important part of the experimental information for the storage of which a computerized system is not suitable: core maps, drawings, explanations to be given only in papers or reports. Therefore, PDF makes reference to such things frequently. There is a further important complement to PDF: this is CLIB which contains the calibration factors of the activation detectors used. As the same activation foils are used repeatedly, it is convenient to give only the foils numbers in the subfiles and to store their calibration factors separately. That is done in CLIB. It is the subject of Chapter 7 how program RFIT handles it.

2.2.2. Text questions

The numbers of text questions range from 1 to 80. Those which are defined for the described version of program RFIT are explained below. When the input data make reference to an undefined text question, an error message will be generated but the subfile can be created.

1. Purpose of the measurement

   The experimenter is supposed to tell here what he/she had in mind when he/she did the measurement. If there is no better idea than to put sentences such as "macroflux measurement by activating fuel rods", it is better to omit this question since this is expressed by the subfile identification and by Question 9. Ideas for Question 1:

   - relation of the subfile to the experimental program of TIC;
   - relation to other subfiles (e.g. whether the measurement is a mere repetition of another one or it is independent);
   - whether the measurement is to solve some methodical problem, or is to check some instr or idea, etc.

2. The name of the person who prepared the measurement
3. The name of the person who performed the measurement
4. The name of the operator in charge

   Questions 2 to 4 tell to whom to turn when something is not clear about the subfile or the measurement. In addition to this, the name (or names) given at Question 3 are frequently used for selecting subfiles (see Chapter 4).

5. Parameters characterizing the reactor state

   The value of the following parameters should be given for the reactor state at which the measurement was performed (the units to be used are given in brackets):

No. of core certificate,
No. of core map,
critical moderator level (cm),
concentration of boric acid (g/liter),
nominal reactor power (watts).

Although Question 5 requires numeric information, it is only partly a numeric question in the sense discussed above because only the numbers of the core certificate and core map are used when program RFIT evaluates the subfile. In case of DRDH and DRDT type measurements, there may be some ambiguity here: the reactivity is measured either as a function of the water level (DRDH) or as a function of the boric acid concentration or the temperature (DRDT). In both cases, one or more of the reactor parameters change during the measurement. In order to avoid ambiguity, let us introduce the following convention: the starting critical state is defined at Question 5 and the values of the variable state parameters should be given at later questions (79 or 80 or both). The core map number will be used by task EVAL for the determination of the lattice ptich (p) if it is not given at Question 67. (Refer to Section 6.3.1.3 for more details.)

6. Sketch number

If some explanatory drawing or sketch is necessary for understanding the information given in the subfile, its number (or some identifier) should be given here. The sketches themselves are collected in Ref. [4].

7. Start and end of the irradiation in ZR-6
8. Start and end of the irradiation in the thermal column

Both Questions 7 and 8 require the specification of the times in the format:

hh/mm/ss

i.e. hours/minutes/seconds relative to the day indicated in the subfile identification. These time data serve only for information. However, do not underestimate their importance: it costs little to specify them but they can be of great help in some cases. Practice has shown that these "only informative" time data can be of great help if one tries to find out what was done simultaneously with other measurements or separately. (Such relations can determine the way of applying eventual corrections.) Special cases:

- For static measurements, ususally both Questions 7 and 8 are relevant and all four time data are within the day given in the subfile identification.
- Things are different in case of reactivity measurements for several reasons. Firstly, the thermal column is not used whence Question 8 may be omitted. Secondly, the whole measurement can last more than one day. If the duration does not exceed 99 hour (i.e. roughly 4 days), this can be

adjusted to the format of Question 7. For example, the data

7 10/30/00 64/45/00

indicate a measurement which started at 10.30 of the day given in the subfile identification and ended at 6.45 two days later. If this seems to be inconvenient, we may use any other time specification for the end of the measurement because we have Question 11 where we can explain our approach and make our comments.

*8. Detector identifications

For each detector type or detector material used in the measurement, two data should be known:

- the identification of the data set in CLIB which contains the calibration factors of the detector material at hand
- and some identification of the detector type.

This information is basic from the point of view of the subfile structure: all subsequent numeric data will be grouped according to the detector types defined at Question 9. Therefore, no numeric data can be given without an answer to Question 9. The total number of detector materials may not exceed 20. Remarks:

a/ Program RFIT accepts any string as a detector material identification (but only the eight first characters are taken into account). This string will appear in all printouts which are related to the subfile. The same string may be used for the identification of several detector materials. At ZR-6, standard notations have been established for the detector materials used in activation measurements. They are listed in Table 2.3. Their use is recommended but not obligatory. If no detector identification is specified in input, program RFIT generates names like ANONYM01, ANONYM02, etc.

b/ At the stage of creating the subfile, any (integer) number is accepted as a calibration data set identifier. Table 2.3 gives also recommendations for the introduction of new identifiers. The identifier will play a role only when evaluating the subfile: the calibation factors belonging to the foils (or other detectors) mentioned in the subfile will be taken from the data set identified. Error messages will be generated in two cases: firstly, when reference is made to a non-existing data set and, secondly, when the subfile refers to a foil for which no calibration factor is available in the identifed data set. Of course, this error message appears only at the stage of the evaluation. If zero is specified as a calibration data set identifier, this means that (according to the opinion of the experimenter) no calibration factors apply for the detector material at hand. CA01 type subfiles are special cases (see below).

c/ Special cases:

- In case of reactivity measurements, the experimenter has a great freedom in defining "detector materials": he/she may assign all his/her data to one detector but he/she may

split them according to his/her neutron counters as well. If it seems expedient to him/her, he/she may introduce any other splitting of the data. Any choice he/she makes, the calibaration data set identifier should be always 0 in case of reactivity measurements.

- There are cases when it is expedient to define the background measurements as pertaining to a special detector material. Whether this is the case it follows from Question 68. It is trivial that the calibration data set number should be 0 for such an artificial "detector material".
- There is an option (see Questions 70, 72 and 75) that monitor data are specified as pertaining to a special material. (This is analogous to the background data.) It is trivial that the calibration data set identifier should be 0 for such artificial "detector materials".
- When the activity of the same foil is measured by different counters, different calibation factors can belong to the corresponding data. In such cases, it is also expedient to define different detector materials for the different counters in order to be able to specify different calibration date set identifiers. In that case, it is not necessary to use different detector material symbols.
- There is a possibility to give the calibration factors at Questions 66 or 79. If this option is applied for some of the materials, the program ignores what is given at Question 9 for the corresponding calibration data set identifiers. (For this reason, it may be 0.)
- In case of CA01 type subfiles, the calibration data set identifier plays a special role: when such subfiles are evaluated, it specifies not the data set where the calibration factors are to be taken from but rather the data set which will be created as a result of evaluating the subfile (for details of this, see Chapters 6 and 7).

10. Comment on the foils or detectors used

It follows from the remarks made at Question 5 that, to a large extent, the definition of the "detector materials" depends on the considerations of the experimenter. Now, Question 10 allows to explain the points of view according to which the materials have been defined at Question 9. In that sense, Question 10 can help to make clear what the "material notations" mean. In particular, it would be rather useful to explain which detector materials' notations are used for background and which ones to some monitor specification.

11. Comment on the irradiation conditions

As a matter of fact, only Questions 5 and 6 serve for characterizing the state of the reactor during irradiation. There are a lot of things which could be necessary to tell what happened during the measurement. In order to have an idea of what is meant here, the following examples are given as to the information expected at Question 11:

- the reactor power could have changed during the irradiation; such an event could make some corrections necessary for the subfile at hand;
- both for axial and radial macroflux measurements, the axial position of the intermediate grid plate can play an important role; now, Question 11 allows to tell about this; etc.

12. Comment on the sKetch

There may be undefined parameters in the sKetch mentioned at Question 6. Of course, any other comment on the sKetch is possible at this question.

13. Comment on the boron concentration determination

At ZR-6, two methods were applied for the determination of the concentration of the boric acid dissolved in the moderator: a chemical method which had an accuracy of 0.5 % and a physical method based on ratios of diluting the moderator with pure distilled water. Now, some information can be given concerning the value which is specified at Question 5. In case of reactivity measurements, Question 13 can be an important source of information as to how the boron concentration was changed during the measurement (especially in cases of DRDH or DRDT type subfiles).

14. to 18. Undefined questions

19. General remark

If all previous questions have been answered as it has been explained above, no more information is needed for understanding the subfile. If some piece of information does not fit into the previous questions, it can be mentioned at Question 19. For example: when some quantities are not given in the standard units, this should be noted here.

20. Pertinent references

If reports, papers, books which are connected to the subfile at hand, their references can be given at Question 20.

21. Description of the activity measuring device

If a standard device was used, it can be interesting to know how it it was set. If a non-standard device was used, its description would be useful here.

22. Start of the activity measurement

There are activation measurements in which the activity of the activated foils is not measured immediately after the activation. Question 22 is defined for the sake of such measurements. The start of the activity measurement is given in the following form:

day/hour/minute/second.

Here, day = 0 if the measurements has been started on the same day as the irradiation has been started. Similarly, day = 1 means that it has been started the next day and so on. This time will be taken as t = 0 for all running time data specified in columns 5 to 7 of the matrix "79". If Question 22 is not answered, the initial moment (i.e. the moment taken as t = 0) will be defined as if

22 0/0/0/0

were given i.e. day = hour = minute = second = 0 is taken. The number of lines given for Question 22 can be 1 or more. The program applies the following convention when evaluating the subfile (see Chapter 6):

- If only one line is given, this defines the initial moment for all materials.
- If the number of lines is n22 > 1, the initial moment for material j will be defined as follows:
  * for $j \leq n22$, the initial moment will correspond to line j; if the day is negative, the initial moment will be 0 (cf. Section 6.32);
  * for $j > n22$, the initial moment will be defined for material j as if Question 22 were not answered at all.

It follows from this that the program need not check whether the number of lines (i.e. n22) is equal to the number of materials or not.

23. Comment on the conditions or the devices of the activity measurement

Which devices were used for the determination of the foil activities, this should be described in reports like Ref. [1]. If something is deviating from the standard, this sould be told at Question 23. If anything is not convenient to specify at Question 21, it may be done at this question, too.

24. Positions within the core

It was explained in Section 2.2.1 that there are data which do not play any role in the evaluation of the subfile but might play an important role in its interpretation. As an example of this, the axial coordinate can be cited at which radial measuremets have been performed. Such data have no significance during data evaluation but they might be essential when radial neutron fields are interpreted (especially when macroflux corrections are applied, see at Question 82).

25. Comment concerning foil calibration

It follows from the explanations given to Question 9 that it can sometimes be useful to explain how the calibration data sets and their identifiers have been chosen.

26 to 38. Undefined questions

39. CAMON data

In measurements performed after 1980, a computer controled activity measuring device has been used the control sotfware of which was called CAMON [5]. As an answer to Question 39, this program automatically generates all the settings of the devices.

40 to 60. Undefined questions

2.2.3. Numeric questions

Questions with numbers 61, 62, etc. are all numeric and their formats are more or less strictly determined. Those which are defined for the described version of program RFIT are explained below. If the input data happen to make reference to an undefined numeric question, an error message will be generated and the subfile can not be created. The format of the input data are given in Chapter 3. Thus, only the meaning of the questions is explained in the present section. Those who are familiar with this, need only Chapter 3 when working with PDF.

The most essential information is at Question 79 where a matrix should be specified separately for each of the detector materials defined at Question 9. Throughout this user's manual, they are referred to as matrices "79". Within a subfile, the actual size and structure of the matrices "79" may be different for different materials but the physical meaning of the individual columns of the matrices are the same. Consequently, it is sufficient to define here the matrix only for one material. All previous questions serve only as complementary data:

- questions defining the structure of the matrices: 63 and 78;
- questions which can simplify the matrix specification: 62, 64, 65, and 66;
- questions defining the corrections (see Chapter 2 of Part 2): 61, 68, 70, 72, 75, and 82;
- questions giving the options for calculating the variances (see Chapter 2 of Part 2): 69, 71, 73, 74, and 76;
- additional information: 67, 80, 81, and 83.

Questions 61 to 80 (incl.) are related to truly raw experimental data while Questions 81, 82, and 83 are connected with the evaluation or are related to results of evaluations. Furthermore, in the explanations given below concerning the physical meaning of the information required by Questions 61 to 83 (incl.), we shall be concerned also with the problem of how the data stored in the subfile are converted into variables and correction factors of the fitting function. However, this will be done only to an extent which is necessary for understanding the subfile structure. A complete discussion of this is given in Chapter 6.

Most of the numeric questions can be understood only if the structure and the physical content of the matrix "79" are known. Therefore, at first reading of this user's manual, the reader is advised to go right to Question 79 and to return to Question 61 only after having been acquainted with the structure of the matrix "79". The amount of the data to be given at Question 79 can be very large. The structure of PDF is defined in a way to reduce this as much as possible. Such simplifications are possible if the measurement shows some regularity in the data structure. In that case, some of the columns of this large matrix can be omitted because the corresponding information can be given in a simpler way at other questions. However, these simplifications are not compulsory but only allowed by program RFIT. In PDF, the following regularities can be turned to advantage as means of simplification:

- Constant quantities: the counting time is frequently constant for all materials. Therefore, it is sufficient to give its value at Question 62 and to omit column 4 of the matrix "79" for some of the materials.
- Cycles: it is a frequent case that the activity of the activated foils is measured several times in the same way and order. In such cases, cycles can be observed in some columns of the matrix. (This is mainly the case when automatic devices do the activity measurement.) Then it is sufficient to characterize one cycle.
- Constant change of some quantitities: the running time and the axial coordinate (columns 3 and 8, respectively) can have such a regularity that they change by the same amount from count to count. There are cases in which this can be observed only within the cycles. All this is taken into account at Questions 64 and 65.
- In certain cases, Question 66 can be a further help in saving the specification of some columns of the matrix "79" (see there).

In the following, the physical meaning of the data to be given at the individual questions is discussed. This might appear hard to follow at first sight but it will become clearer if the following is taken into account. When the input data are specified for fitting, one has to take care of the following variables (see Chapters 1 and 2 of Part 2):

- fitted values y (which can be given at Questions 79 or 80);
- the x variable of the fitting function (which can be given at Questions 65, 66, 79, and 80, depending on the actual case);
- the t variable i.e. the running time variable needed for the calculation of the corrections (which can be given at Questions 64 or 79);
- the corrections and the error options.

When evaluating the subfile, program RFIT looks for the values of the x, t, and y variables at Question 79 (eventually at Question 80) first. When the data are not available there, it turns to other questions. In this sense, there is a certain hierarchy among the questions which will be defined below, too.

61. Dead time τ of the activity measuring device

One has to specify pairs of data (K, τ_K) for each detector material where K is its number and τ_K is the dead time (in seconds). Remarks:

- the dead time correction will be carried out according to Equs. (2.1.) and (2.9) of Part 2;
- if the dead time correction is negligible for all materials, this question should be omitted;
- if the dead time is not given for some of the materials, program RFIT assumes that $\tau_K = 0$.

62. Counting time T (if it is constant)

The specification of T is analogous to that of the dead time. If it is not given at Question 62, this means that it is variable and its values will be given in column 4 of the matrix "79". The value of T is used

- for the dead time correction if Question 61 is answered, and
- as a multiplicative correction according to Equ. (2.1.b) of Part 2 if it is variable.

Program RFIT turns to Question 62 only if no data are given in column 4 of the matrix "79".

*63. Cycles

Most measurements show some cyclic regularity. As an example, let us take the case of N foils the activities of which are measured M times in such a way that each of the M runs is an exact repetition of the others. The specification of the large data matrix "79" can be simplified a lot for such measurements if this cyclic regularity is taken into account. Now, Question 63 requires to give (N_K, M_K) for all materials K. Remarks:

- Question 63 may not be omitted if further data are given;
- the number of rows of matrix "79" will be equal to $N_K M_K$;
- if there is no cyclic regularity, $M_K = 1$ and N_K should be equal to the number of rows of matrix "79";
- the cycles may be different for different materials;
- background data may be parts of a cycle if they were measured according to the cyclic regularity.

64. Equidistant running time

Let t_i be the running time i.e. the time when the data in row i of matrix "79" were measured. We call it equidistant if

$$t_i = t_1 + (i-1)\Delta t. \qquad (2.1)$$

If this is not the case, the running can be given either in column 3 or in columns 5, 6, and 7 of the matrix. Control variable MT serves for distinguishing between the following two cases:

MT = 1 means that formula (2.1) holds for the whole matrix; in that case, subscript i runs continuously through the cycles i.e. its value does not go back to i = 1 at the beginning of each cycle;

MT = 2 means that either t_1 or Δt (or both) depend on the cycle i.e. we have

$$t_{ji} = t_{j1} + (i - 1)\Delta t_j \qquad (2.2)$$

for cycle j and measurement i within the cycle where now i = 1 at the beginning of each cycle.

The program turns to Question 64 only if no data are given in columns 3 or 5 to 7 of the matrix "79".

65. Equidistant axial coordinate

Let z_i be the axial coordinate inside the reactor at which the data in row i of the matrix "79" were measured. Its values are normally given in column 8 of the matrix. Things are simpler if it is equidistant i.e. if

$$z_i = z_1 + (i - 1)\Delta z. \qquad (2.3)$$

Control variable MX serves for distinguishing between the following two cases:

MX = 1 means that formula (2.3) holds for all cycles; subscript i returns to i = 1 at the beginning of each cycle;

MX = 2 means that either z_1 or Δz (or both) depend on the cycle i.e. we have

$$z_{ji} = z_{j1} + (i - 1)\Delta z_j \qquad (2.4)$$

for cycle j and measurement i within the cycle where i = 1 at the beginning of each cycle.

The z_i values calculated in this way will become the values of the x variable of the fitting function (see e.g. Chapter 1 of Part 2). The program turns to Question 65 only if no data are given neither in column 8 of the matrix "79" nor in column 3 of the matrix "66". When evaluating spectra given at Question 80, Question 65 is ignored since the values of the x variable will be taken from Question 80.

Questions 64 an 65 are formally identical for both values of the control variables. Note, however, that options MT = 1 and MX = 1 mean different things if there are several cycles.

66. Data for the individual foils

Question 66 can be rather useful when simplifying the specification of the matrix "79" in case of activation measurements. In matrix "79", several rows can be related to the same foil. Data like the coordinates (columns 8 and 9) or calibration

factors (column 12) are unambiguously determined by the identifier of the foil whence it is sufficient to give only the identifier at Question 79, and to give the correspondence between the other data and the identifiers somewhere else. Now, this is the role of Question 66. Here also, we give a matrix which can have 5 columns at maximum. The correspondence of the individual columns of matrices "66" and "79":

column in matrix "66"	column in matrix "79"
1	1
2	2
3	8
4	9
5	12

The most frequent case is that conditional notations (column 1) are used at Question 79 which are then interpreted on the basis of the matrix "66". Remarks:

- as to the physical meaning of the individual columns, see the remarks at Question 79;
- all foils should be mentioned at Question 66 which appear at Question 79 else an error message will be generated when evaluating the subfile;
- the program turns to Question 66 only if some information is missing in the matrix "79";
- if, for some material K, all columns which are given at Question 66 are missing at Question 79, the program considers matrix "66" as a minor of matrix "79" which enters for each cycle repetitively; a necessary condition of this is that the number of rows of matrix "66" must be equal to N_K.

The use of matrix "66" is illustrated by the following simple example. Assume that matrices "66" and "79" are the following for some material:

matrix "66"

col. 1	col. 2	col. 5
1	47	0.97
2	53	1.07
3	28	1.15

matrix "79"

col. 1	col. 10
1	3541
2	4139
3	4511
1	3223
2	4295
3	4479
1	3698
2	4062
3	4671

This is equivalent to the following matrix "79" what shows that we can really spare a substantial amount of input data with the help of a matrix "66":

col. 1	col. 2	col. 10	col. 12
1	47	3541	0.97
2	53	4139	1.07
3	28	4511	1.15
1	47	3223	0.97
2	53	4295	1.07
3	28	4479	1.15
1	47	3699	0.97
2	53	4062	1.07
3	28	4671	1.15

67. Hexagonal coordinates of the core centre

Five numbers are required at Question 67:

(h10, h20) - the hexagonal coordinates of the core centre,
p - the lattice pitch (in cm), and
(h11, h21) - the hexagonal coordinates of a lattice point needed for fitting function No. 43 (see Figure 1.4 of Part 2).

If Question 67 is omitted, program RFIT takes the following default values:

h10 = h20 = 35

p = 1.27 cm.

If Question 67 is answered, one may give either 2 numbers (namely, the core centre data h10 and h20) or 3 numbers (namely, the core centre data and the lattice pitch) or 5 numbers (all those listed above). If only 2 are given, the default value is taken for p. These data will be used for the computation of radial coordinate according to Equs. (1.20), (1.21), and (A.12) of Part 2 (cf. also Section 6.3.1.3).

68. Laboratory background
70. Decay correction
72. Correction for detector efficiency

In Questions 68, 70, and 72, control variables KORA, KORD, and KORE, respectively, serve for distinguishing between the options listed below. If the corrections are computed by some formula, this is based on running time t (column 3 or columns 5, 6, and 7 of the matrix "79" or the values specified at Question 64). The options are the following:

0 - the correction is not applied;
1 - the correction is given in matrix "79";
2 - the correction is given by the formula

$$\mu_i = A \exp(-\lambda t_i) \qquad (2.5)$$

where constants A and λ are the same for all cycles;

3 - the correction is given by the formula

$$\mu_i = A_j \exp(-\lambda_j t_i) \qquad (2.6)$$

where constants A_j and λ_j vary with the cycle (i.e. with subsrcipt j);

4 - interpolation of monitor data where the monitor is a special foil;

5 - interpolation of monitor data which are given as belonging to a special material;

6 - averaging of monitor data where the monitor is a special foil, fuel element, etc.;

7 - averaging of monitor data which are given as belonging to a special material;

8 - the same as option 4 but the monitor is identified not by the foil number but by a given value of the coordinate.

These options are summarized in Table 2.4 where it is also indicated which input data are required when the subfile is created (see Chapter 3). The use and role of monitors are explained in Chapter 2 of Part 2. Note that the options determined by control variables KORA and KORD are not the same here as in Part 2: we have more options here than there. When the subfile is evaluated, program RFIT will do the necessary conversions (see Chapter 6).

Remarks concerning the laboratory background (Question 68):

- for a constant background, use options 2 or 3 with $\lambda = 0$ or $\lambda_j = 0$, respectively;
- in case of otpions 2 and 3, the background is related to the same counting time (T) as the fitted values y: if T is variable, A should be given for unit counting time while it should be related to T if the latter is constant (refer to Sections 6.3.5 and 6.3.8 for further details).
- options 4 and 5 are needed when the background changes with time according to a function which is determined experimentally;
- options 6 and 7 are relevant only for the background;
- it is a convention in program RFIT that the foil identifier for bakcground foil-monitor data should be 0 (in case of options 4 and 6).

Remarks concerning the decay correction (Question 70):

- mainly options 2, 4, and 5 are used for the decay correction;
- option 3 is a principal possibility of approximating a decay correction which is not strictly exponential; of course, the cycles should be defined accordingly;
- the decay correction is special for fitting functions NNo. 30 and 38: the details are discussed in Section 6.3.7.

Remarks concerning the detector efficiency (Question 72):

- options 4 and 5 have been foreseen for this correction;
- program RFIT accepts data for Question 72 but this has not been used in the practice of the ZR-6 experiments; consequently, their use is not elaborated and these data are not taken into account during the evaluation of the subfiles. (This remark applies to Question 73, too.)

69. Error of laboratory background
71. Error of decay correction
73. Error of correction for detector efficiency
74. Error of counts
76. Error of x data

Here, we speak consequently on experimental errors although the program works with variances (i.e squares of the errors). In Questions 69, 71, 73, 74, and 76, control variables MWA, MWD, MWE, MWY, and MWX serve, respectively, for distinguishing between the options listed below:

0 - no correction;
1 - Poissonian random variable (variance approximated by the count);
2 - constant variance;
3 - variance given to each measured value (this option is not to be used in PDF);
4 - constant relative variance;
5 - Poissonian random variable (variance approximated by the fitted function - this option is not to be used in PDF);
6 - the counting time is measured by a preset scaler while the preset count is constant (see Appendix 2 of Part 2);
7 - the counting time is measured by a preset scaler while the preset count varies with the cycles.

These options are summarized in Table 2.5 where it is also indicated which input data are required when creating the subfile (see Chapter 3). Remarks:

- if a correction is obtained by interpolating or averaging some monitor data, the error option of the monitor data determines the error option of the correction; in such a case, it is useless to specify the error option for the correction; the error option specified at Question 69 is taken into account only if the option is 1 at Question 68 (cf. Sections 6.3.5); since option 1 is not allowed at Question 70, the error option eventually specified at Question 71 is always ignored (cf. Section 6.3.7);
- if no option is given for the error, the program takes the default option which is option 1;
- a vanishing variance can be specified via option 2 with C = 0;
- when option 2 is specified at Question 69, parameter C should be related to unit counting time if T is variable while C should be related to T if the counting time is constant (cf. Sections 6.3.5 and 6.3.8, and the remarks made

to Question 68);
- the role of option 0 varies with the respective variable:
  * MWA = 0 is interpreted as a Poissonian variable if KORA = 1 at Question 68;
  * MWY = 0 is interpreted as a Poissionian y;
  * MWX = 0 means that the variance of the x variable is zero;
  * MWD = 0 and MWE = 0 are irrelevant (see remarks above).

75. Remanent activity

There are detectors whose activities do not decay completely until the next irradiation. This is the case e.g. when the fuel rods are used as detectors of the fission rate. Their remanent activities are measured immediately before they are activated. Such remanent activities have to be given as data corresponding to a special material. One should not forget to give background data for the measurement of the remanent activity at Questions 68 and 69 (cf. Equ. (2.13) of Part 2). The correspondence of the remanent activity data and the other activities is determined through the foil identifiers (i.e. columns 2 of the matrices "79" or columns 2 of the matrices "66").

77. Undefined question

*78. Columns of the matrices "79"

Question 78 defines the columns of the matrices "79". For each material for which a matrix will be given at Question 79, it should given at Question 78:

- how many columns are there in the matrix "79" and
- which are the numbers of these columns.

As explained below, the physical meanings of the individual columns of the matrix "79" are strictly defined but this does not necessarily mean that the input data should contain the columns in that order. The order of naming the columns at Question 78 indicates the input order of the columns. The program will restore the natural permutation when the subfile is created. This feature of program RFIT is sometimes of great help in the practice. As an example, let us assume that we have a large matrix consisting of columns 2, 8, 9, 10, and 12 which is punched ready for input but we are faced with the following problem: the hexagonal coordinates (i.e. columns 8 and 9) are given systematically in reversed order and the calibration factors (i.e. column 12) have been punched immediately after the foil numbers (i.e column 2). Then, at Question 78, it is sufficient to give the following order of the columns for that matrix:

2, 12, 9, 8, 10

upon which the program will restore the natural order of the columns.

*79. Data

The results of all measurements performed at critical assembly ZR-6 can be represented in the form of a matrix the columns of which contain (see explanations later):

1 - conditional identifier of the foil,
2 - identifier of the foil,
3 - running time,
4 - counting time,
5 - day of starting the measurement,
6 - hour of starting the measurement,
7 - minute of starting the measurement,
8 - coordinate of the detector,
9 - coordinate of the detector,
10 - fitted variable y,
11 - background,
12 - calibration factor.

These data form the rows of the matrix "79" which is usually given for every material defined at Question 9. In case of DRDH, DRDT, and R281 (R282 and R283) type measurements, a special approach is necessary which is explained at Question 80. General remarks concerning Question 79:

- if a material is mentioned at Question 79 for which data are missing either at Questions 63 or 78, this will generate an error message and the subfile will not be created;
- normally, a matrix "79" is assumed to be created for each material defined at Question 9 but it need not be necessarily the case; omission of the matrix "79" can lead to an error meassage only when the subfile is evaluated;
- not all columns of this matrix need to be given but only those which apply for the material at hand; however, column 10 is essential if the matrix is given at all.

Comments on the individual columns of the matrix:

- The conditional identifiers of the detectors (or activation foils) are used mainly when the activities are measured by some automatic foil exchanger (or some analogous device) in the output of which the identification of the foils can be different from their own identifiers. In this case, one has to specify at Question 66 which foil identifiers correspond to the conditional ones. If data are missing at Question 66 for a conditional foil identifier mentioned at Question 79, this leads to an error message when the subfile is evaluated.
- The foil identifiers given in column 2 of matrix "79" (or column 2 of matrix "66") must coincide with the numbers appearing in the calibration data set eventually given at Question 9 for the material at hand. The foil identifiers will become the values of the x variable when CA01 type (i.e. foil calibration) measurements are evaluated by task EVAL (see Chapter 6). In addition to this, they are needed for task EVAL also for the determination of:

* the foil calibration factors (if they are not given in column 12 of the matrix "79");
* the remanent activity (see at Question 75);
* the decay and/or background corrections (when KORD = 4, KORA = 4 or 6, see at Questions 70 and 68, respectively);
* the macroflux correction (see Question 82); this use of the foil identifiers is restricted only to those which are given at Question 66;
* the x variable in case of spectral measurements if the position numbers are not given in column 9 of the matrix "79".

If a foil identifier is negative, this can lead to errors when evaluating the subfile. Therefore, it is advisable to avoid the use of negative numbers as foil identifiers.

- The running time can be given in column 3. If this is the case, columns 5, 6, 7 are not necessary. In most cases, however, the running time is a calculated quantity the calculation of which is based on columns 5, 6, 7 and on Question 22. (Normally, not the running time itself but the data to be given in columns 5, 6, 7 are the raw data.) If the running time is missing at Question 79, the program looks for it at Question 64. (In this latter case, the data eventually given at Question 22 are ignored.) Anyhow, the running time is needed only when some corrections apply. As to the day (i.e. column 5), day = 0 means the same day as specified in the subfile identifier. Would all data be 0 in column 5, the whole column may be omitted since this is the default value for it. Anyhow, adding multiples of 24 hours to the values given in column 6 always allows to omit column 5 completely. (All this is up to the consideration of the experimenter.)
- Column 4 contains the length of the time interval during which the fitted value given in column 10 was measured. The role of the value given here is twofold: firstly, it will be used for the calculation of the dead time correction factor (if a dead time different from zero is given for this material at Question 61) and, secondly, it will become a multiplicative correction factor (cf. Chapter 2 of Part 2) since the program assumes the measuring time variable if it is given at Question 79 (without checking whether all elements of column 4 happen to be equal). When it is not given here, the program looks for it at Question 62 if a positive dead time was specified at Question 61 for the material at hand.
- The physical meaning of columns 8 and 9 varies with the type of the measurement and, to some extent, also with the fitting function. (All what is stated below for these columns of matrix "79" apply to columns 3 and 4 of matrix "66" mutatis mutandis.) Anyhow, the data given here will yield the x variable of the fitting function. Originally, columns 8 and 9 were foreseen for the first and second (i.e. the roman and arabic, respectively) hexagonal coordinates of the core position in which the foils (detectors) specified in column 2 were irradiated. This interpretation remains valid for radial macroflux measurements (cf. Table 2.2).

Program RFIT tolerates the following carelessness here:

* there is a common convention that hexagonal coordinates h1 and h2 may be united in one single variable as

$$h = h1 + h2/100 \qquad (2.7)$$

the value of which may be put in any one of columns 8 and 9: the program will find it and will recognize that this convention has been applied;
* in case of some measurements, only the radial coordinate r (cf. Section A.1.2 of Part 2) is given in the subfile; *now*, the program will recognize this, too, and the corresponding data may be given either in column 8 or column 9. (As to how do program differentiates between these cases, refer to Section 6.3.1.3.)

There are further special cases:

* For axial measurements (cf. Table 2.2), the axial coordinate is supposed to be given in column 8. If it is not available at Question 79, the program turns to Questions 66 and 65 (in this order).
* For azimuthal (MI06 type) measurements, the azimuthal angle is supposed to be given in column 9. If data are not available there, the program turns to Questions 66 and 65 (in this order).
* For DRDH type measurements, the moderator level data are supposed to be given in column 8 but they are not used when the subfile is evaluated since the directly measured data are supposed to be stored as spectra at Question 80.
* For DRDT type measurements, the temperature data are supposed to be given in column 3 but they are not used when the subfile is evaluated since the directly measured data are supposed to be stored as spectra at Question 80. There are cases in which the core temperature was measured in several locations of the core in order to check whether these data show some dispersion. If so, the raw results of the temperature measurements themselves are given as "spectra" at Question 80 and the identifiers of the relevant spectra are given in column 8 at Question 79. (In that latter case, a comment at Question 24 can be rather informative and helpful for those who try to intepreta the subfile long after the experiment has been performed.)
* For MI02 type measurements, columns 8 and 9 contain the data relative to the macroflux correction (see Question 82) and the position data required by fitting functions NNo. 32 and 38 (cf. Section 1.2.10 of Part 2), respectively. As to the position data, the general rule applies: if no data are given in column 9 of the matrix "79", the program looks for them in column 4 of

the matrix "66". If no data are given there, neither, an error message is generated. The macroflux correction data, however, are exceptions: the program looks for them only in column 3 of the matrix "66" and ignores what is eventually given in column 8 of the matrix "79". The fact is that actually no subfile exists which would contain such data at Question 79.

*80. Spectra

There are complex measurements which cannot be evaluated in one step. Examples of this are the experiments in which thin cuts of the fuel pellets are activated in selected positions of the core. In the thin fuel pellets, the Np-239 and Ce-143 activities are determined in the following way. The gamma-spectra emitted by the irradiated fuel cuts are measured by a Ge-Li semiconductor detector. One of the Gaussian functions NNo. 16 to 19 can be fitted to these spectra. The areas under the respective Gaussian peaks are proportional to the activities of Np-239 and Ce-143 built up in the fuel cuts. If this measurement is compared with foil octivation measurements, it is seen that the fitted peak areas may be considered as the data which normally come into column 10 of the matrix "79". It follows from this that a data storage system of raw experimental data would have to accomodate a whole measured spectrum in each element of column 10 of the matrix "79". Now, this problem is solved with the help of Question 80: only the identifiers of the meausured spectra are given in column 10 but the spectra themselves are stored at Question 80. As a matter of fact, the possibilities of Question 80 were used mainly in case of DRDH and DRDT type subfiles for which the spectra contain the results of doubling time measurements. As a rule, some exponential fitting function (e.g. NNo. 1 or 3) is applied for the determination of the reactivity first which can then be considered as "data" in subsequent evaluations. As already mentioned at Question 79, the results of the temperature measurements are also stored as spectra in some DRDT type subfiles. Remarks concerning the evaluation of the spectra:

- No corrections other than the dead time correction may be applied. The latter is based on the information given at Questions 61 and 62. This means that only such spectra may belong to the same material for which both the dead time and the measuring time are the same. This circumstance can influence the definition of the materials at Question 9.
- The x variable can be only equidistant i.e. one must have

$$x_i' = x_1 + (i - 1)\Delta x \qquad (2.8)$$

for channel i of the spectrum where Δx is the channel width of the analyser used for measuring the spectrum. Parameters x_1 and Δx are given along with the spectrum and the data eventaully given at Questions 65 or 79 are ignored.

81. Point drop limits

The point drop limits are normally specified along with the input data given for data evaluation. However, it is sometimes expedient to store them in the subfile itself. This is the case e.g. when the data defining the point drop limits are complicated. Anyhow, the program takes these data into account only if no point drop data are given for the actual evaluation (see Chapter 6). The point drop options available here are similar to those described in Chapter 6 of the present part and Chapter 5 of Part 2 for fitting tasks EVAL and RFIT. As there are some *differences*, the options are exposed here fully in order to avoid annoying cross references with other chapters. Two variables are necessary to characterizing the point drop data: the number of point drop steps (NSTEP) and option variable MST. NSTEP can range from 1 to 20. As exposed in Section 2.1 of Part 4, the point drop limits are defined in program RFIT by intervals of subscript i (see Equs. (2.8) of Part 4). This is the basic option corresponding to MST = 0. In some cases, however, it is simpler to define the point drop limits on the basis of the x variable. The possible values MST and the corresponding options are listed as follows: only such points will be taken into account in an individual point drop step for which

- subscript i belongs to a given interval (MST = 0);
- the value of the x variable belongs to a given interval (MST = 1 or 2);
- the value of the x variable is different from given values (MST = 3 or 4);
- the value of the x variable belongs to a given interval and it is different from given values (MST = 5 or 6); this is a combination of the previous two options.

It no data are given for some material, the default values are taken when evaluating the subfile by task EVAL:

NSTEP = 1
MST = 0

and no points are dropped in the fitting. Here, options 1, 3, and 5 are the same as 2, 4, and 6, respectively. These options are different only for fitting (i.e. for RFIT and EVAL). When MST = 0, the subscript limits for i should correspond to the set of the x values resulting after the determination of the *decay correction factors* (for KORD > 3) and the background corrections (for KORA > 3): the data belonging to the monitors will be left out, and the points will be reordered in ascending order of the x values.(See Chapter 2 of Part 2, Questions 68 and 70, and Section 6.3 for further details.) When MST > 0, the subscript limits will be computed after the corrections.

82. Macroflux correction

The role of the macroflux correction is given in Equs. (2.1) of Part 2. One has two options in PDF for its specification

controlled by option variable KORM whose values can be:

0 - no macroflux correction;
1 - the correction factors are calculated by the formula

$$\mu_z = \cos[B_z(z - z_o)] \quad . \tag{2.9}$$

where the values of B_z and z_o are stored at Question 82 and those of z in column 3 of the matrix "66";
2 - the correction factors themselves are given in column 3 of the respective matrix "66".

It follows from the foregoing statements that no information stored at Question 79 is used for the determination of the macroflux correction. It may be necessary for MI02 type subfiles. The values of B_z and z_o to be used in Equ. (2.9) are usually estimated on the basis of macroflux measurements. Variable z appearing in Equ. (2.9) is typically the axial position at which the foil was irradiated. There are cases, however, in which one has to correct for differences in the radial positions, too. Option KORM = 2 has been introduced for the sake of such cases. For practical reasons, the macroflux correction factors will be treated formally as foil calibration factors when the subfile is evaluated i.e. they will become calibration factors if no true calibration factors have to be taken into account else the products $\mu_c \mu_z$ will appear as calibration factors.

83. Results of spectrum evaluations

The evaluation of spectra requires a lot of computing time. Therefore, a special question has been introduced for storing the results of such evaluations. The format of these data is the same as that of Question 66 while the meaning of the columns is different:

1 - the identification number of the evaluated spectrum;
2 - the number of the fitting function used (MODE);
3 - version number (if the same spectrum has been evaluated more than once);
4 - the fitted value of the parameter;
5 - the error of the fitted parameter.

In the present version of program RFIT, Question 83 serves only for documentation i.e. no direct evaluation of the data stored there is possible as yet. It should also be noted that, due to severe complications not discussed here, the results of a spectrum evaluation can not be included into the subfile automatically. Maybe these drawbacks will be eliminated in later versions of program RFIT (especially as far as the former one is concerned).

84 to 99. Undefined questions

Table 2.1. Subfile types

type	definition			calibration data set numbers
MA01	spatial distribution of reaction rates	rad.	foils inside the fuel rod	Cu: 1000-1039 In: 1300-1399 Dy: 1400-1499 Au: 1500-1599 Eu: 1600-1699 small foils: ..01-..50 ring foils: ..51-..99
MA02		ax.		
MA03		rad.	foils on the clad surface	
MA04		ax.		
MA05		ax.	wire activation	
MA10		rad.	fuel activity measured by a scintillator	2000-2999
MA11		ax.		
MA13		rad.	fuel activity by ion. chamber	
MI01	microdistribution in the fuel rod measured by foils			Dy: 3000-3099 Mn: 3100-3149 Eu: 3150-3199 In: 3200-3299 Cu: 3300-3399 U: 3500-3599 Pu: 3900-3999
MI02	spectral ratios measured by foils			
MI06	azimuthal distribution measured by foils			
R281	activity of fuel pellets measured by	Ge(Li) detector		4000-4999
R282		column chromatography		
R283		paper chromatography		
CA01	calibration of foils			
TR01	track detectors	microdistribution		
TR02		fast fission index δ^{28}		
DRDH	$\partial\varphi/\partial H$ measurement			
DRDT	$\partial\varphi/\partial T$ measurements			
MEAS	other types of measurement			

Table 2.2. Defined questions

No.	type	definition
1	text	purpose of the measurement
2	text	name of the person who prepared the measurement
3	text	name of the person who performed the measurement
4	text	name of the operator in charge
5	numbers	parameters characterizing the reactor state
6	text	sketch number
7	numbers	start and end of the irradiation in ZR-6
8	numbers	start and end of the irradiation in the thermal column
*9	numbers	detector identifications
10	text	comment on the foils or detectors used
11	text	comment on the irradiation conditions
12	text	comment on the sketch
13	text	comment on the boron concentration determination
19	text	general remark
20	text	pertinent references
21	text	description of the device used for measuring the activity
22	numbers	start of the activity measurement
23	text	comment on the conditions or the devices of the activity measurement

Table 2.2. (continued)

No.	type	definition
24	text	positions within the core
25	text	comment concerning foil calibration
39	text	CAMON data
61	numeric	dead time of the device used for measuring the activity
62	numeric	counting time T (if it is constant)
*63	numeric	cycles
64	numeric	equidistant running time
65	numeric	equidistant axial coordinate
66	numeric	data for the individual foils
67	numeric	hexagonal coordinates of the core center
68	numeric	background specification
69	numeric	error of background
70	numeric	decay correction
71	numeric	error of decay correction
72	numeric	correction for detector efficiency
73	numeric	error of correction for detector efficiency
74	numeric	error of counts
75	numeric	remanent activity
76	numeric	error of x data
*78	numeric	columns of the data matrices "79"
*79	numeric	data

Table 2.2. (continued)

No.	type	definition
*80	numeric	spectra
81	numeric	point drop limits
82	numeric	macroflux correction
83	numeric	results of spectrum evaluations

Table 2.3. Symbols of detector materials to be used at Question 9

symbol	material
DYAL3.1	Dy O in Al, 1 mg/cm of Dy O
DYAL5.0	5 % of Dy in alloy with Al
DYAL10.0	10 % of Dy in alloy with Al
INAL1.0	1 % of In in alloy with Al
INAL0.43	0.43 % of In in alloy with Al
EUAL10.0	10 % of Eu in alloy with Al
EUAL5.0	5 % of Eu in alloy with Al
MNNI24.0	24 % of Mn in alloy with Ni
MNNI10.0	10 % of Mn in alloy with Ni
MNAL10.0	10 % of Mn in alloy with Al
UAL17.0	17 % of U in alloy with Al (enrichment: 90 %)
PUAL17.0	17 % of Pu in alloy with Al
AU100	gold metal
AUAL3.85	3.85 % of Au in alloy with Al
CU100	copper metal
FUEL	fuel element used as detector
FUEL1.6	1.6 % enriched fuel pellet used as detector
FUEL3.6	3.6 % enriched fuel pellet used as detector
FUEL4.4	4.4 % enriched fuel pellet used as detector

Table 2.4. Options for specifying corrections

Notations:

N - identifier of the monitor foil
K - number of the material which is the monitor
x_o - the special value of the x variable which identifies the monitor
n - the number of data taken for the interpolation; a positive (negative) value of n means logarithmic (linear) interpolation

N.B. In the described version of program RFIT, n = 2 is taken independently of what is in the subfile.

KORA KORD KORE	definition	input data at Quest. 68, 70, 72	remark
0	no correction	nothing	
1	correction given at Question 79	nothing	only for background
2	$A*\exp(-\lambda t_i)$ (A and λ constant for a material)	$A, \lambda, (\Delta A)^2$	only λ is given for decay correction since A = 1; as to variable counting time, see remarks at Questions 68 and 69
3	$A_j*\exp(-\lambda_j t_i)$ (A_j and λ_j vary with cycle index j)	$[A_j, \lambda_j, (\Delta A_j)^2, \text{for } j=1,\ldots,M_K]$	
4	logarithmic (linear) interpolation of n foil monitor data (N)	N, n (-n)	since N=0 for the background, only n (-n) is given
5	logarithmic (linear) interpolation of n monitor data given as a special material (K)	K, n (-n)	
6	averaging of foil monitor data (N)	N	since N=0 for the background, nothing is given
7	averaging of monitor data given as a special material	K	
8	the same as 4 but the the monitor is given by a $x_O = x$	x_O, n (-n)	

Table 2.5. Options for specifying errors

Notations:

μ - the value of the variable concerned
C - its variance

N.B. Concerning option MW = 0, see remarks at Questions 69, 71, 73, 74, and 76.
MWX = 7 generates an error message.

MWA, MWD, MWE, MWY, MWX	definition	input data at Questions 69, 71, 73, 74, 76
0	see remarks at N.B.	nothing
1	Poissonian data	nothing
2	constant variance	C
3	variance given point-wise (not to be used)	nothing
4	constant relative variance	$(\mu/C)^2$
5	Poissonian data	nothing
6	the counting time is measured by a preset scaler	preset count C
7	the same as 6 but the preset count varies with the cycles	preset count C for each cycle

3. Task NEW : creating new subfiles

The files composing PDF are described in Section 5.1. New subfiles are created on the temporary disk file 77 via the SYSIN input. The rules of this are described in the present chapter. It is explained in Chapter 5 how to use these subfiles further and how to record them to the permanent disk files 80 and 81 and magnetic tapes 60 and 61.

3.1. General structure of the input data

The general structure of the input data is illustrated in the next page. As this scheme shows, the overall structure of the input data is:

- the data are introduced by task Keyword NEW ,
- the data belonging to a subfile are introduced by the subfile identifier,
- the answers to the text and numeric questions are separated from one another by a line containing only "++++",
- the data belonging to a subfile are also terminated by a line containing only "++++",
- the input data belonging to a Keyword NEW are terminated by another task Keyword (i.e. one of those given in Table 2.1 of Part I); if no other operation follows, an END should close the input data lines.

It is worthwile to note that, of course, the data belonging to any other operation may preceed Keyword NEW .

The identifiers of the subfiles created by the same task Keyword NEW are noted in the computer memory. The array used for this is dimensioned to 20 items which imposes the limitation that the maximum number of subfiles following one Keyword NEW is 20. The noted subfile identifiers remain available for the other task Keywords which are eventually given in the same line as NEW . Further details of this are given in Chapter 5. This list is available also for task EVAL even if it is not given in the same line as NEW but after the input data pertinent to it (see Chapter 6). This list remains valid until the next operation which involves explicite subfile selection (cf. Section 5.2.2).

```
NEW
identifier of the first subfile
        :
  answers to text questions
        :
++++
        :
  answers to numeric questions
        :
++++
identifier of the second subfile
        :
  answers to text questions
        :
++++
        :
  answers to numeric questions
        :
++++
```

and so on until the next task keyword.

If task keyword NEW is given alone, this is equivalent to the sentence

NEW LIST

i.e. the successfully created new subfiles will be listed automatically unless some other task keyword is given atfer NEW (in the same line). For the case when no other operation is applicable, the no operation keyword NLST (No LiST) has been introduced to this end. (Refer to Section 5.5 for more details.)

Program RFIT checks the input data in all possible ways in order to prevent creating uncorrectly formulated subfiles since no subfile can be further manipulated (i.e. listed, corrected, evaluated etc.) if it contains structural inconsequences. The chances of the program are rather limited for detecting input errors. For example, it is not possible to check (in the moment of creating the subfile) whether the value of a particular matrix element is given correctly or not. It is in a position to check only whether the input data having structural significance are correct (i.e. numbers of questions, materials, matrix columns, matrix rows, option variables etc.). It is not difficult to detect an input error if reference is made to e.g. column 15 of a matrix "79" or if option variable KORA is given to be 10. Both are inacceptable according to Section 2.2.3. In order to further facilitate detecting ev ntual input errors, some not too severe restrictions have deliberately been imposed. That is why the question numbers have to follow one another in ascending order although the way of storing the data in PDF would not require this.

The checks actually performed are summerized as follows:

- the total number of materials (as determined by Question 3) may not exceed 20;

- no reference to material numbers may exceed the total number of the materials;
- the values of the option variables should be one of those mentioned in Chapter 2 (especially in Tables 2.4 and 2.5);
- all structural variables should be positive (or at least non-negative in some cases);
- the total amount of data to be stored in a matrix is limited to 5000;
- the total amount of data to be stored in a spectrum is limited to 5000;
- the references made to the individual columns of the matrices should be within the ranges specified for this in Chapter 2 (see more about this at Questions 66, 79, and 83);
- for some variables, certain limitations hold in case of special options (see e.g. the case of M_K at Questions 64 and 65);
- the question numbers should follow each other in ascending order;
- within one question, the references to individual materials should be made in asecending order of their numbers (Questions 61 to 65, incl., are exceptions to this rule).

At the individual questions, further limitations are mentioned which are special or additional with respect to those cited above.

Should an error of this kind be detected, the program stops creating the subfile at hand and reads the input lines without interpreting them until the next task keyword or the next subfile identifier and the subfile is not created. (The possible error messages are explained in Chapter 5 of Part 1.)

3.2. Text questions

The format of the text questions is simple: each line is introduced by the number of the question followed by the corresponding information. There is no limitation concerning the number of lines belonging to a given question. Remarks on individual questions:

Question 5's recommended format is illustrated by the following example:

```
5    110/110   75.59   4.0   3.0
```

which corresponds to core configuration 110/110 with a critical moderator level of 75.59 cm, a boron concentration of 4.0 g/liter, and a reactor power of 3 watts. From all this, only the identification of the core configuration (i.e. 110/110) will be used for subfile selection (cf. Chapter 5).

Question 9 should be answered if numeric questions follow (as is usually the case). The total number of materials in the subfile are defined by the number of lines given with question number 9. The general format of such a line is illustrated by the following example:

```
9  1401  DYAL5.0
```

where 1401 is the identifier of the calibration data set and DYAL5.0 is the name of the detector material. (Tables 2.1 and 2.3 contain recommendations for the data set identifiers and the material names, respectively.) When the subfile is created, no check is made whether the specified calibration data set exists in CLIB. Both the calibration data set number and the material name may be omitted. Referring to the previous example, this means that each of the following lines are accepted:

```
9  1401
9  DYAL5.0
9
```

The consequences of these omissions are:

- if the data set number is omitted, this means that it is 0 (instead of 1401 in the actual case);
- if the material name is omitted, the program generates a name like ANONYM01, ANONYM02, etc. (where the last 2 digits show the serial number of the material at hand also determined by the program).

Both may be omitted i.e. a line containing only the question number 9 is also accepted by the program. When the data set number is omitted but the material name is given, the latter may directly follow the question number if the first character of the name is a letter. (In the opposite case, it is better to give a 0 for the data set number.)

Question 22's recommended format is illustrated by the following example:

```
22     0/09/31/00
```

which means that the activity measurement has been started at 9 o'clock 31 minutes of the same day as the irradiation. These data will be used when the running time t will be calculated from the values given in columns 5, 6, 7 of the matrices "79".

3.3. Numeric questions

The formats of the numeric questions are not as uniform as those of the text questions. They will be explained without reference to the corresponding sections of the previous chapter. We assume their knowledge in the present section.

Questions 61 and 62: the first line contains the question number and the number of the materials for which data follow. Then for each material for which data are given, the format is the following:

No. of mat., dead time (counting time).

If the dead time (counting time) is identical for all materials, it is not necessary to give it for all materials. In that case, it is sufficient to give a 0 for the material number and to give the common dead time (counting time) only once. This is made clear by the following example:

```
61 1
0  2.1E-6
62 3
1  20.0  3  30.0  2  30.0
```

This means that the dead time is 2.1 μsec for all materials and the maesuring time is

$$T_1 = 20 \text{ sec}, \quad T_2 = T_3 = 30 \text{ sec}.$$

The information given for Question 61 is equivalent to

```
61  3
1  2.1E-6  2  2.1E-6  3  2.1E-6
```

The sequence of mentioning the materials is arbitrary.

Question 63: the first line of the input data contains only question number 63 while the next line contains the pairs (N_K, M_K) for all materials. If one line is not sufficient, the data may be continued in the next ones.

Questions 64 and 65: the first line contains the question number (i.e. 64 or 65) followed by the number of those materials for which input data follow. The following data must be started in separate lines for each material mentioned:

option variable (MX or MT), No. of material, data

The data depend on the actual value of the option variable. If it is

1: t_1 and Δt (z_1 and Δz),
2: t_{j1} and Δt_j, for $j = 1, 2, \ldots, M_K$
(z_{j1} and Δz_j, for $j = 1, 2, \ldots, M_K$)

Here, t and z refer to Questions 64 or 65, respectively. The sequence of mentioning the materials is arbitrary. It is noted furthermore that the value of M_K may not exceed 20 if the option variable is 2. (It is checked only here and for a value of 2 of the option variable whether this restriction is observed.)

Questions 66 and 83: the first line contains the question number (i.e. 66 or 83) followed by the number of those materials for which input data follow. The latter may be

divided in two parts:

- first the structural data of the matrices are given in separate lines for every material mentioned here:

  No. of the material, number of the rows, total number of the columns, and the list of the numbers of the individual columns;

- next follow the elements of the matrices row by row.

Special rules and limitations to be observed here:

* the material numbers should follow one another in ascending order; the matrices themselves should also be given in the same order;
* the elements of a matrix should follow each other row by row but there is no restriction as to how they are broken into lines except that matrices for the individual materials should always start in separate lines;
* the number of columns is limited to 5;
* the numbers of the individual columns should be listed in the same order as the matrix elements are given within the rows; independently of this order, the matrix will be stored in PDF according to the natural order of the columns.

Question 67: the first line contains only question number 67. The next line contains

h10, h20, p, h11, h21.

(See explanations in the previous chapter.) At least the first two of these five numbers should be given.

Questions 68 to 74, and 76: the question number is followed (in the same line or in the next one) by the values of the option variable for all materials. According to Tables 2.4 and 2.5, additional input data may be necessary depending on the actual values of the option variable. These latter data are given in the next lines (in separate lines for different materials). Remarks:

- 0 is accepted for the option variable but its value may not be negative neither may it be out of the range defined in Tables 2.4 and 2.5;
- the additional data are given without the number of the material since their order determines to which material they belong;
- no empty lines are necessary for those materials for which additional data are not given.

Question 75: the question number is followed (in the same or in the next line) by an integer for all materials in the following way:

- the integer is 0 if no remanent activity belongs to the given material,
- the integer is equal to the number of the special material under the name of which the results of the corresponding remanent activity measurements are given.

Question 78: the first line contains question number 78 and an integer. For each material, the columns of matrix "79" are specified as follows:

total number of columns, list of the numbers of the individual columns.

If the integer number given together with question number 78 is positive, this should be given for every material. If it is negative, this means that the information concerning the columns is the same for all materials. Correspondingly, it is sufficient to give this information only once which will be valid for every material.

Remarks:

* the total number of the columns in a matrix may not exceed 10;
* the order of listing the column numbers should correspond to their actual permutation according to which they will be given at Question 79;
* the matrices will be stored in PDF according to the natural permutation of the columns (see Section 2.2.3);
* although it is possible to correct these column data later (see next chapter), this is rather complicated, therefore, it is advisable to proceed carefully here when the subfile is created.

Question 79: the first line contains only question number 79. This is followed by the elements of the matrices "79" given rowwise for all materials. The number of rows is $N_k M_k$ as given at Question 63 while the number of columns is given at Question 78. There is no limitation as to how the data belonging to the same matrix are broken into lines except that matrices for the individual materials should always start in separate lines.

Question 80: the first line contains question number 80 and the number of those materials for which data are to follow. Then for each of these materials, it is given:

No. of the material and the number of the spectra to be given for this material.

Then for each spectrum, the following data should be given:

an identifier, the number of channels, x_1 and Δx, followed by the spectrum itself in a new line

where the spectrum identifier is a (generally positive) integer, Δx is the channel width, x_1 is the value of the x variable for the first channel (cf. Equ. (2.8)). The number of data given for the spectrum should be equal to the number of channels. Remarks:

- There no limitation for the number of spectra to be stored.
- The number of channels is limited to 5000.
- The data belonging to each material and each spectrum should be started in a separate line.
- The material numbers should follow one another in ascending order.
- It is advisable to put the spectrum identifiers in column 10 of the corresponding matrix "79" in order to make clear to which reactor states the individual spectrums belong. However, the program does not check whether this is really done.
- As the spectra are included into the subfile one by one, the program does not check whether their identifiers are different. If two spectra get the same identifier, the program may ignore the second one of such spectra in some later operations. It is discussed in Section 4.4.4 how the first sepctrum can be renamed in order to make the second one unambiguously available.

Question 81: the first line contains question number 81 and the number of those materials for which data are to follow. Then for each of these materials, it is given:

- in the first line: the No. of the material and an integer followed by
- further data depending on the actual value of this integer.

The integer shows the value of option variable MST and the number of point drop steps NSTEP according to the following convention:

100*MST + NSTEP

where the meaning of these variables is explained in Section 2.2.3. For example, 309 means MST = 3 and NSTEP = 9 while 12 means MST = 0 and NSTEP = 12 etc. The further data mentioned above are to be given as functions of MST (for ℓ = 1, 2, ..., NSTEP):

When MST = 0, 2*NSTEP integer numbers are given which will be the subscript limits ($i_{1\ell}$, $i_{2\ell}$) appearing in Equs. (5.8) of Part 2. (See remark below.)
When MST = 1 or 2, 2*NSTEP real numbers are given which

will be the lower and upper limits (x^{ℓ}_{min} and x^{ℓ}_{max}) of the x variable pertaining to the individual point drop steps.

When MST = 3 or 4, the following data are given:

$$n_\ell\ ,\ x^\ell_1\ ,\ x^\ell_2\ ,\ \ldots\ ,\ x^\ell_{n_\ell}$$

where n_ℓ is the number of x values to be left out in step ℓ followed by the list of these particular x values. The value of n_ℓ may be 0. In this case, the list of the x values is omitted.

When MST = 5 or 6, the combination of the previous two cases is given i.e. it is given:

$$x^\ell_{min}\ ,\ x^\ell_{max}\ ,\ n_\ell\ ,\ x^\ell_1\ ,\ x^\ell_2\ ,\ \ldots\ ,\ x^\ell_{n_\ell}$$

where the meaning of the variables has been explained previously.

The data belonging to each material and each step should be started in a new line. When MST = 0, subscript limits $i_{1\ell}$ and $i_{2\ell}$ should correspond to the set of the x values resulting after the determination of the decay correction factors (for KORD > 3) and the background corrections (for KORA > 3): the data belonging to the monitors will be left out and the points will be reordered in ascending order of the x values. (For further details, see Questions 68 and 70, Section 6.3, and Chapter 2 of Part 2.) When MST > 0, the subscript limits will be computed after the corrections.

Question 82: question number 82 is followed (in the same or in the next line) by the values of option variable KORM for all materials in the following way:

- if KORM = 0 or 2 for a material, no further data are required,
- if KORM = 1, the values of B_2 and z_0 (see Equ. (2.9)) should be given in separate lines.

Any other values of KORM will generate an error message.

4. Task CORR: correcting and complementing existing subfiles

The files composing PDF are described in Section 5.1. Existing subfiles can be corrected or complemented via the SYSIN input. The rules of this are described in the present chapter. The subfile to be corrected is first copied to disk file 74. (For the discussion of where the program takes the original subfile from, detailed considerations are necessary which are made in Section 5.2.) The corrected version of the subfile is put to disk file 75 first which, upon successfully terminating the correction, is copied to disk file 77. It is explained in Chapter 5 how to use it further and how to record it to permanent disk files 80 and 81 and magnetic tapes 60 and 61.

The corrected version of the subfile will have the same identifier as the original one. The program does not delete the original version of the corrected subfile automatically. If it is no more needed, the user has to take care of its deletion (cf. Section 5.4.4). If, however, both the original and the corrected versions are needed, it is advisable (but not mandatory) to give a new identifier to the corrected version by using task RNM7 (cf. Section 5.3.3).

As a rule, the corrected subfiles are identified explicitely, i.e. by giving their identifiers directly. There is also an implicite way of selecting subfiles which, with some restrictions, can be used also in connection with task CORR. (The details of both modes of subfile selection are discussed in Section 5.2.) The present chapter, however, is formulated in terms of the explicite subfile selection. The use of the implicite subfile selection in connection with task CORR is illustrated by Example 3 of Section 5.6.

4.1. Generalities on task CORR

The general structure of the input data is the same as that shown in Section 3.1 for the case of creating new subfiles. The only difference is that the task keyword now is CORR:

```
CORR
identifier of the first corrected subfile
        :
  corrections of the text questions
        :
++++
        :
  corrections of the numeric questions
        :
++++
identifier of the second corrected subfile
        :
  corrections of the text questions
        :
++++
        :
  corrections of the numeric questions
        :
++++
```

and so on until the next task Keyword. As this scheme shows, the overall structure of the input data is:

- the data are introduced by task Keyword CORR,
- the data belonging to a subfile are introduced by the subfile identifier,
- the corrections of the text and numeric questions are separated from one another by a line containing only "++++",
- the data belonging to a subfile are also terminated by a line containing only "++++",
- the input data belonging to a Keyword CORR are terminated by another task Keyword (i.e. one of those given in Table 2.1 of Part 1); if no other operation follows, an END should close the input data lines.

It is worthwile to note that, of course, the data belonging to any other operation may preced Keyword CORR.

If task Keyword CORR is given alone, this is equivalent to the sentence

```
CORR CLST
```

i.e. the successfully corrected subfiles will be listed automatically unless some other task Keyword is given atfer CORR (in the same line). (Refer to Section 5.5 for further details.) For the case when no other operation is applicable, the no operation Keyword NLST (i.e. No LiST) has been introduced to this end.

The identifiers of the subfiles created by the same task Keyword CORR are noted in the computer memory. The array used for this is dimensioned to 20 items which imposes the limitation that the maximum number of subfiles following one Keyword CORR is 20. The noted subfile identifiers remain available for the other task Keywords which are eventually given in the same line as CORR.

Further details of this are given in Chapter 5. This list is available also for task EVAL even if it is not given in the same line as CORR but after the input data pertinent to it (see Chapter 6). This list remains valid until the next operation which involves explicite subfile selection (cf. Section 5.2.2).

Furthermore, all what is stated in Section 3.1 on the limitations to be observed and the checks performed by the program concerning the internal consistency of the input data having structural significance remains also valid according to the sense. Consequently, it is advisable to get acquainted with Chapter 3 before the study of the present one.

In order to simplify the expression in the following, we shall speak of "questions" by what we shall mean the "information corresponding to the respective questions". As the information corresponding to a question can be subdivided into data corresponding to different detector materials (to be defined at Question 3), we shall use the following simplification, too: "the information corresponding to a particular material within a given question" shall simply be referred to as the "material". We hope that this apparent carelessness will lead to less misunderstanding than the rather involved style of repeating these long expressions many times at full length.

The changes allowed by program RFIT can be classified as follows:

- complementing the subfile with new questions;
- deleting existing questions from the subfile;
- changing parts of an existing question.

The latter case can be divided into further cases in an analogous way:

* complementing the question with new materials;
* deleting existing materials from the question;
* changing parts of an existing material within a question.

Complementing the subfile with a question and deleting a question from it is a relatively simple matter just as is complementing a question with new materials and deleting materials from it. All this can be done in almost the same way as new subfiles are created (see Chapter 3). The only complicated (and unfortunately the most frequent) case is the last one i.e. when a material is changed within a particular question. Most of the present chapter is devoted to the explanation of how the corresponding input data are prepared. The rules of this might appear complicated at first sight but, in reality, they are not more complicated than the changes themselves: program RFIT can perform too great a variety of modifications to allow this to be simple.

It is noted finally that it is not less complicated to assure the internal consistency of the structural parameters of the subfile when it is changed than when it is created from scratch. In this context, changing Questions 3, 63, 78, and 79 can lead to

structural problems. If, for example, the value of the product $N_K M_K$ is changed at Question 63, the corresponding matrix "79" should also be changed accordingly. The user should bear such interdependences in mind. The explanations will make them clear in a detailed way. Anyhow, the program performs all the necessary checks of the input data in order not to allow to spoil the internal consistency of the existing subfiles while they are changed or amended.

In the following, the rules of input preparation are presented according to the previous classification. It is important to note, however, that different types of corrections may be mixed in the same input stream i.e. the subsequent questions may be corrected according to different rules.

If an error message is generated, the further input data are read without intepretation until the next subfile identifier or task keyword and no corrected version of the subfile appears. The possible error messages are explained in Chapter 5 of Part 1.

4.2. Complementing the subfile with a question

When the input data prepared according to the rules of task NEW are introduced by task keyword CORR (in place of NEW), the program accepts them and the following happens:

- if the question did not occur previously in the original subfile, the corresponding information is simply included in it;
- if the question occurred previously in the original subfile, the action of the program depends on the character of the question:
  * in case of text questions, the new information is put after the information already stored for this question;
  * in case of numeric questions, the old information is replaced by the new one.

In addition to this, the program allows to insert new information into the text questions in any place of the existing text. It is explained in Section 4.4 how to do this.

Such corrections can change the structural parameters of the subfile. The consequences of this are the following for the questions concerned:

Questions 3 and 5: As explained in Section 5.2, the information stored at these questions is used for subfile selection. If this is changed, the selection is done according to the new situation.

Question 9: If it is changed in this way, this is equivalent to an increase of the total number of materials with all its structural consequences for numeric questions:

- the program accepts reference also to the newly defined materials at later questions;
- all option variables and structural parameters (e.g. N_K and M_K at Question 63) are considered as 0 for the new materials unless they are specified ultimately.

Question 22: If it is complemented in this way, this results in a change of the number of lines at this question. As explained in Section 2.2.2, this leads to another way of interpreting these initial time data. Assume for example that originally there was only one line at Question 22. According to the general treatment, this line was valid for all materials existing in the subfile. If a new line is added, this results in the following change of treatment: the original line will hold only for the first material while the new line for the second material, and the program will consider that no initial time is defined for the other materials.

Question 63: If the value of M_K changes for some of the materials, this can have an influence on Questions 64, 65, and 68 to 74 (incl.) in case of special options (see Section 2.2.3). Similarly, if the value of the product $N_K M_K$ changes for some material, the corresponding matrix "79" should be corrected in accordance with this. An error message is generated if the corresponding corrections are not made at these later questions.

Question 78: If the columns of a matrix "79" changes, the corresponding changes should be performed at Question 79. An error message can be generated otherwise.

4.3. Deleting a question

Existing questions can be deleted from the subfile in a very simple way. It is sufficient to give

* N

in a separate line where N is the question number. As an effect of this, all the information stored in the subfile for the given question is deleted. Of course, if Question N does not exist in the subfile, this generates an error message. With respect to deleting entire questions, there is no difference between the text and the numeric parts of the subfile. Restrictions to be observed:

- The program does not allow to delete Question 9.
- If attempt is made to delete Questions 63 and 78, this is ignored without printing a message about it (of course, only if the subfile contains further data).

4.4. Correcting parts of a question

If only parts of a question are changed, several groups of questions have to be considered separately:

- text questions,
- matrices (Questions 66, 78, 79, and 83),
- spectra (Question 80),
- other numeric questions.

Within each of these special cases, parts of the input data formats can be identical with those corresponding to task NEW . In order to avoid cross references with Chapter 3, some paragraphs of the latter are reproduced here. The input formats are summarized in Tables 4.1 to 4.5 (incl.). Hopefully, after the user makes himself/herself acquainted with the detailed explanations given below, it will be sufficient to consult only these tables.

4.4.1. Text questions

The general form of changing parts of a text question is

* N, *n1*, *n2*

followed by the new text to be added to the question where

N - is the question number,
n1 - is the number of the first line to be corrected within Question N,
n2 - is the number of lines to be deleted from Question N (including and starting from line n1).

If other lines need to be corrected and/or added, several such corrections may follow one another. Parameter n2 may be 0. If so, the new lines are added in front of line n1.

This way of correcting text questions may be combined with corrections of the type described in Section 4.2. In order to separate the two formats from one another, a line containing only an asterisk (*) must be put after the corrections discussed in the present chapter. All this is made clear by the following example. Assume that Question 10 originally contained three lines:

```
original line 1
original line 2
original line 3
```

which is corrected by the following input data:

```
*10  1  0
10  new line 1
*10  2  1
10  new line 2
10  new line 3
*
10  new line 4
```

As a result of this, the corrected state of Question 10 will be:

```
  new line 1
  original line 1
  new line 2
  new line 3
  original line 3
  new line 4
```

As to this form of correcting text questions, the following remarks apply:

- References to lines within the question (i.e. the values of parameter n1) should correspond to the original numeration of the lines independently of how their numeration changes as results of the corrections actually performed.
- The successive corrections should go forward within the question i.e. the value of parameter n1 may not be less than the value the sum (n1 + n2) had in the previous correction.
- If corrections are made with asterisks (i.e. according to the format discussed in the present section), all new text lines should be related to the question just corrected. For example, corrections of the form

  ```
  *10  2  1
  10  new line 1
  11  new line 2
  ```

  are not allowed and will generate an error message. The correct form of the correction intended in the previous example is:

  ```
  *10  2  1
  10  new line 1
  *
  11  new line 2
  ```

  The general rule is: if the correction of a question has been started with the asterisk format discussed in the present section, this has to be closed by a line containing only an asterisk before the correction can go over to another question.
- If attempt is made to delete more lines than exist within the question i.e. the value of (n1 + n2) goes beyond the existing lines, an error message is generated.
- The previous restriction is stronger when the correction of this type would result in deleting all existing lines without adding new ones. This generates an error message, too. The logic behind this restriction is that this complicated way of deleting a question fully occurs probably by mistake since there is a much

simpler way of doing this (cf. Section 4.3).

As to some of the text questions, some special remarks are necessary:

Questions 3 and 5: As explained in Section 5.2, the information stored at these questions is used for subfile selection. If this is changed, the selection will be done according to the new situation.

Question 9: If it is changed in the way discussed, one has to be very careful since it can have a serious impact on the overall subfile structure. That is why several special cases have to be considered separately:

- If, in every correction step, the number of lines remains the same, the subfile structure does not change. This is a simple amendment of the material names and/or the calibration data set identifiers.
- If, in one or more correction steps, the total number of the lines is reduced (i.e. less new lines are added than deleted), this is equivalent to the deletion of one or more materials from the subfile. Those materials become deleted for which an amendment line is not given. For example, the following data result in the deletion of the 4th material:

  ```
  * 9  3  2
  9  1401  DYAL5.0
  *
  ```

  since an amendment line is given only for the 3rd material. This is allowed and the following happens. Upon terminating the correction of Question 3, the program goes through all Questions 61 to 99, and deletes from them all information pertaining to the deleted materials. All later corrections should be formulated in accordance with this new situation. Returning to the previous example, a later reference made to material number 4 will correspond to the material whose number used to be 5 in the original subfile. Question 22 is the only text question at which the deletion of materials might induce some changes. If so, Question 22 should be corrected accordingly by the user (see remarks later).
- If more lines were added than deleted, this would be equivalent to an insertion of new materials between the existing ones. This would require an unnecessary restructuring of the subfile. Consequently, the program does not allow to change Question 9 in this way and an attempt to do so generates an error message. New materials can be added to the subfile in the way described in Section 4.2 in which case the new materials continue the sequence of the existing ones. As the order of the materials is

neutral, there is no advantage in inserting the new materials between existing ones. That is why it was considered better to prohibit such a complicated correction. Let us return to the previous example and let us assume that the total number of materials is 6. The following correction is not allowed for:

```
* 9  1  1
9  1001  CU100
9  1002  CU100
* 9  3  2
9  1401  DYAL5.0
*
```

since it would be equivalent to the insertion of a new material between the fist and the second materials. The correct way of doing the intended correction is the following:

```
* 9  .  1
9  1001  CU100
* 9  3  2
9  1401  DYAL5.0
*
9  1002  CU100
```

which induces the following changes in the subfile:

(1) the first material is renamed;
(2) the fourth material is deleted; as a result of this, the originally fifth material becomes the fourth one; the originally 6th material becomes the fifth one;
(3) the newly added material (named CU100) becomes the sixth one according to the new subfile structure.

As a result of all this, the total number of the materials remains 6 (which is only incidental since, as stated above, it might have been changed).

Question 22: If it is corrected in the way discussed, this can result in a change of the number of lines at this question. As explained in Section 2.2.2, this can lead to another way of interpreting these initial time data. Assume for example that there were originally two lines at Question 22 from which one is deleted. Then the remaining line will become valid for all materials existing in the subfile (since there is only one line) while the interpretation of the original information was quite different: the first line was valid for the first material, the second line for the second material, and the program assumed that no initial time was defined for the other materials.

4.4.2. Numeric questions (except matrices and spectra)

In most cases, the simplest and safest way of correcting numeric questions is to give the relevant information as a new one in the form described in Section 4.2 (i.e. according to the format of task NEW). This means in practice that not the subfile itself but those original input data are corrected which have been used when the subfile was created. If the original input data are no more available or their volume is too large, it might appear more advantageous to introduce some minor corrections in the existing subfile. This is mainly the case when matrices and spectra are corrected or complemented. The possible ways of correcting the latter are discussed in Sections 4.4.3 and 4.4.4 while the correction of the other numeric questions is discussed in the present section. The correction formats discussed in the present section are summarized in Table 4.1.

The first line pertinent to each corrected question has the following general form:

* N, n1, n2

where N is the question number while the meaning and role of parameters n1 and n2 depend on question number N. For some questions, one or both of them may be omitted.

Questions 61, 62, 64, 65, 81: The value of n1 gives the total number of those materials which are deleted from the question while n2 gives the total number of those materials for which new information will be given. Any of them may be 0 but not both at the same time. If n2 is not given, the program assumes that it is 0. The subsequent lines are:

- list of the material numbers to be deleted (i.e. n1 integers, only if n1 > 0);
- the new information is given for n2 materials according to the rules of task NEW (see Chapter 3), only if n2 > 0.

Both groups of data should start in separate lines. If, within the first group, reference is made to a material for which no information exists at Question N, this generates an error message.

Question 63: No data may be deleted from this question, consequently, only n1 has some significance here: it is the number of those materials for which new information is given. (n2 is not needed.) Then, beginning from the next line, n1 times three integers are given:

$$(K, N_K, M_K)$$

where K is the material number. N_K and M_K are defined in Section 2.2.3. If, as a result of correcting

Question 63, the product $N_K M_K$ changes for some material with respect to its original value, the corresponding matrix "79" should also be corrected in accordance with this. Otherwise, an error message is generated at Question 79.

Question 67: neither n1 nor n2 is needed for this question. The relevant new information should be given in a separate line as in case of task NEW.

Questions 68 to 76, and 82: only n1 is needed and its meaning is the total number of the materials for which new information will be given. Then, starting in the next line, n1 pairs of integers (K, L_K) are given where K is the material number and L_K is the new value of the control variable for material K (i.e. L_K is KORA for N = 68, MWA for N = 69, etc.). If L_K = 0 is given for material K, this deletes the original information (if any) from Question N. In the next lines, the additional data are given depending on the actual value of L_K just as in case of task NEW. These data should start in separate lines for the individual materials.

4.4.3. Correction of matrices (Questions 66, 78, 79, and 83)

The matrices stored at Questions 66, 79, and 83 can be corrected in several ways. Program RFIT allows to perform the following correction types:

(1) adding, replacing, and deleting complete matrices;
(2) adding, replacing, and deleting complete matrix rows;
(3) replacing submatrices of the matrices; as a special case of this, the value of any given element of the matrix can be corrected;
(4) adding, replacing, and deleting complete matrix columns;
(5) permutation of the matrix comlumns.

There can be cases when the desirable changes cannot be realized by the application of only one of these correction types. Therefore, the program allows to perform corrections of different types one after the other in relation to the same matrix. It is important to note that input data pertinent to each such correction step have to be formulated in accordance with the results of all previous correction steps. (In the following, the application of one type of correction to a matrix will be called a correction step.) The general format of the first input line is

* N, n1

where N is the question number and n1 is the total number of correction steps in relation to all materials. The further approach is slightly different in case of matrices "68" and "83", for one hand, and in case of matrices "79", for the other hand. The reason of the difference is that the strucural parameters of matrices "79" are stored at Questions 63 and 78 while those of matrices "66" and

"83" are stored at the questions themselves. Therefore, these questions will be discussed below separately.

Questions 66 and 83: The input formats are summarized in Table 4.2. Here, reference is made to the correction types defined above. In all what follows, the material number will be denoted by K. The individual correction types are differentiated from one another by the format of the first line belonging to the correction step.

(1) The first line contains

K (or **K), total number of rows, total number of columns, list of the column numbers

followed by the matrix elements given row by row. As usually, the first matrix element should start in a separate line but there is no further restriction as to how the following data are broken into lines. Remarks to this type of correction:

- Such a correction is allowed only once in relation to a given material.
- If K is given without asterisks, the new matrix will be added to the subfile replacing the old one (if any).
- If K is given with two asterisks (**), there must be a matrix for material K in the original subfile. (An error message is generated in the opposite case.) The new matrix replaces the old one.
- If only **K is given without the structural data, the corresponding matrix is deleted from the subfile. An error message is generated if there is no matrix for material K.

(2) The first line contains

* K * n

where n is the total number of corrections in this step. Then, for each correction, the following data are given beginning in a new line:

i1, i2, i3,
and the elements of the new rows

where

i1 is the number of the first corrected row,
i2 is the total number of the rows to be deleted,
i3 is the total number of the new rows.

Remarks to this correction type:

- Independently of how the matrix changes as results of the correction step, the values of parameters i1,

i2, and i3 are supposed to refer to the original state of the matrix.
- In subsequent corrections, the values of parameter i1 may only increase. More precisely, it must be more than the value the sum (i1 + i2) had in the previous correction.
- No reference may be made to rows deleted previously.
- The rows to be deleted may not go beyond the existing ones.
- Any one of i2 and i3 may be 0. If i2 = 0, the new rows are inserted preceding row i1.
- When the elements of the new rows are given, the first one should start in a separate line but there is no further restriction as to how they are broken into lines in the following.

(3) The first line contains

* K ** n

where n is the total number of the corrections in this step. Then, for each correction, the following data are given starting in a new line:

i1, i2, i3, $(l_j, j= 1, 2, \ldots, i3)$,
and the new elements of the submatrix (i.e. i2*i3 values) given rowwise

where

i1 is the number of the first corrected row,
i2 is the total number of the rows to be corrected (may not be 0!),
i3 is the total number of the new columns to be corrected (may not be 0!),
l_j are the numbers of the columns to be corrected.

Remarks to this correction type:

- In subsequent corrections, the values of parameter i1 may only increase. Would this restriction exclude some of the desired corrections, the latter can be performed in another step of this type.
- Reference may be made only to existing matrix rows and columns.
- The order within the list of column numbers should correspond to the order according to which the columns of the submatrix are given. Independently of this order, the program will restore the natural permutation of the columns.
- The first element of the submatrix should start in a separate line but there is no further restriction as to how the following data are broken into lines.

(4) The first line contains

* K * n1 * n2

where

n1 is the total number of the columns to be deleted,
n2 is the total number of the new columns.

The next line contains the list of the numbers of the deleted and inserted columns. Then the elements of the submatrix formed from the new columns follow row by row.

Remarks to this correction type:

- As columns to be deleted, only existing ones may be referred to.
- The order within the list of new column numbers should correspond to the order according to which the columns of the submatrix are given. Independently of this order, the program will restore the natural permutation of the columns.
- The first element of the submatrix should start in a separate line but there is no further restriction as to how the following data are broken into lines.
- If any of the new columns exists in the original subfile, the latter is replaced by the new one.

(5) The first line contains

* K, (l_j, j = 1, 2, ..., n_c)

where

l_j is the number of that column where column j of the original matrix is moved to,
n_c is the total number of the columns (not given in input since it is known from the subfile).

Question 78: The input formats are summarized in Table 4.3. Here, reference is made to the correction types defined above. In all what follows, the material number will be denoted by K. Only corrections of types (1), (4), and (5) are related to matrix columns. Consequently, types (2) and (3) are not mentioned here. If the changes performed in the matrix columns induce some changes in the matrices themselves, the corresponding changes will be performed only if they are referred to at Question 79, too (see below). The individual correction types are differentiated from one another by the format of the first line belonging to the correction step.

(1) The first line contains

K (or **K), the total number of columns, the list of column numbers.

Remarks to this type of correction:

- In relation to a given material, it may be performed only once.
- In case of new materials, this is the only possible type (without asterisks).
- In case of old materials, this may be done both with and without asterisks.
- If the total number of columns changes as a result of such a correction, the corresponding matrix "79" should be corrected accordingly (see below). An error message is generated in the opposite case.

(4) The first line contains

* K * n1 * n2

where

n1 is the total number of the columns to be deleted,
n2 is the total number of the new columns.

The next line contains the list of the numbers of the deleted and inserted columns.

Remarks to this correction type:

- Only existing columns may be referred to as columns to be deleted.
- The order within the list of column numbers should correspond to the order according to which the columns of the submatrix will be given at Question 79. Independently of this order, the program will restore the natural permutation of the columns.

(5) The first line contains

* K, $(\ell_j, j = 1, 2, \ldots, n_c)$

where

ℓ_j is the number of that column where column j of the original matrix is moved to,
n_c is the total number of the columns (not given in input since it is known from the subfile).

Remark to this correction type:

- If both corrections types (4) and (5) have been specified at Question 78, the deletion of the columns will be performed prior to their permutation. This should be borne in mind while compiling the input data.

Question 79: The input formats are summarized in Table 4.4. The individual correction types are differentiated from one

another by the format of the first line belonging to the correction step.

(1) The first line contains

K (or **K)

followed by the matrix elements given row by row. The total number of rows and the total number of columns are taken from Questions 63 and 78, respectively. As usually, the first matrix element should start in a separate line but there is no further restriction as to how the following data are broken into lines. Remarks to this type of correction:

- Such a correction is allowed only once in relation to a given material.
- Such a correction step is not permitted for a material for which a correction of type (4) has been performed.
- If K is given without asterisks, the new matrix will be added to the subfile replacing the old one (if any).
- If K is given with two asterisks (**), there must be a matrix in the original subfile. (An error message is generated in the opposite case.) The new matrix replaces the old one.
- If only **K is given without further data in the line, the corresponding matrix is deleted from the subfile. An error message is generated if there is no matrix for material K.

(2) The first line contains

* K * n

where n is the total number of corrections in this step. Then, for each correction, the following data are given starting in a new line:

i1, i2, i3,
and the elements of the new rows

where

i1 is the number of the first corrected row,
i2 is the total number of the rows to be deleted,
i3 is the total number of the new rows.

Remarks to this correction type:

- Independently of how the matrix changes as results of this correction step, the values of parameters i1, i2, and i2 are supposed to refer to the original state of the matrix.
- In subsequent corrections, the values of parameter

i1 may only increase. More precisely, it must be more than the value the sum (i1 + i2) had in the previous correction.
- No reference may be made to rows deleted previously.
- The rows to be deleted may not go beyond the existing ones.
- Any one of i2 and i3 may be 0. If i2 = 0, the new rows are inserted preceding row i1.
- When the elements of the new rows are given, the first one should start in a separate line but there is no further restriction as to how the following elements are broken into lines.
- As a result of these changes, the total number of the rows should become equal to the product $N_K M_K$ given at Question 63 for the material at hand.

(3) The first line contains

* K ** n

where n is the total number of corrections in this step. Then, for each correction, the following data are given starting in a new line:

i1, i2, i3, (l_j, j= 1, 2, ..., i3),
and the new elements of the submatrix (i.e. i2*i3 values) given rowwise

where

i1 is the number of the first corrected row,
i2 is the total number of the rows to be corrected (may not be 0!),
i3 is the total number of the new columns to be corrected (may not be 0!),
l_j are the numbers of the columns to be corrected.

Remarks to this correction type:

- In subsequent corrections, the values of parameter i1 may only increase. More precisely, it must be more than the value the sum (i1 + i2) had in the previous correction. Would this restriction exclude some of the desired corrections, the latter can be performed in another step of this type.
- Reference may be made only to existing matrix rows and columns.
- The order within the list of column numbers should correspond to the order according to which the columns of the submatrix are given. Independently of this order, the program will restore the natural permutation of the columns.
- The first element of the submatrix should start in a separate line but there is no further restriction as to how the following data are broken into lines.

(4) The first line contains

* K * n1 * n2

where

n1 is the total number of the columns to be deleted (it is given but not used since it is known from the corrections performed at Question 78),
n2 is the total number of the new columns. (It should be identical with its value given when Question 78 was corrected. This requirement serves as a means of checking whether Questions 78 and 79 are corrected in accordance with one another.)

Then the elements of the submatrix formed from the new columns are given row by row (only if n2 > 0).

Remarks to this correction type:

- If columns to be deleted were specified at Question 78, their deletion from the matrix "79" is carried out now.
- If a permutation of the matrix columns (i.e. a correction of type (5)) was specified at Question 78, this also is performed now. It follows from this that this form of correcting matrix "79" should be initiated at Question 79 even if only corrections of type (5) have been performed at Question 78. In this latter case, the input data are n1 = n2 = 0. If corrections of both types (4) and (5) have been specified at Question 78, the deletion of the columns is performed prior to their permutation.
- At Question 79, corrections of type (4) may refer to materials to which corrections of types (2) and (3) are performed. On the other hand, corrections of types (1) and (4) may not be related to the same material.
- The order of giving the columns within the rows should correspond to the order given at the correction of Question 78. Independently of this order, the program will restore the natural permutation of the columns.
- The first element of the submatrix should start in a separate line but there is no further restriction as to how the following data are broken into lines.
- If any of the new columns exist in the original subfile, they are replaced by the new one.

4.4.4. Correction of spectra (Question 80)

As a rule, the largest amount of data are stored at Question 80 but the data structure is much simpler than in case of matrices. Program RFIT allows to perform the following correction types:

(1) adding, replacing, and deleting all data belonging to a material;
(2) correcting some of the spectra belonging to a material and adding new ones;
(2a) adding, replacing, and deleting complete spectra from a material;
(2b) correcting parts of spectra:
(2b1) renaming a spectrum (i.e. changing its identifier),
(2b2) changing the contents of individual channels,
(2b3) changing x_1 and Δx.

The input formats of these correction types are summarized in Table 4.5. The first line contains:

* N, n

where

N - is the question number (i.e. 80),
n - is the total number of those materials for which data follow.

The following data depend on the correction type which is applied to the individual materials. The material numbers and the spectrum identifiers will be denoted by K and j, respectively. The individual correction types are differentiated from one another by the format of the first line belonging to the correction step.

(1) The data are given as in case of task NEW with the exception that material number K may stand both with and without two asterisks (**). The following cases are possible:

- If K stands without asterisks, the new information is included into the subfile. If something existed in the original subfile for material K, this is now replaced by the new one.
- If K stands with two asterisks (**), the effect is the same as in the previous cases but an error message is generated if no information exists in the original subfile for material K.
- If only

  ** K

  is given without continuation, this deletes all information for material K.

(2) The first line contains:

* K, n1, n2

where

n1 - is the total number of the corrected spectra,
n2 - is the total number of the new spectra.

Then the input data are given for the individual spectra (each

starting in a new line). First come the data for the existing spectra (i.e. for n1 of them) followed by the data for n2 new spectra. The existing spectra can be corrected both by schemes (2a) and (2b) while, according to the sense, the new spectra can be added only by scheme (2a). Any of n1 and n2 may be 0 but not both at the same time. The corresponding data formats are the following:

(2a) The data are given as in case of task NEW with the exception that spectrum number j may stand both with and without two asterisks (**). The following cases are possible:

- If j stands without asterisks, the new information is included into the subfile for material K. If a spectrum with identifier j exists in the original subfile for material K, this is now replaced by the new one.
- If j stands with two asterisks (**), the effect is the same as in the previous cases but an error message is generated if no spectrum with identifier j exists in the original subfile for material K.
- If only

  ** j

  is given without continuation, this deletes the spectrum identified by j.
- When the new spectra are included into the subfile, the program does not check whether their identifiers are different from those of the existing spectra. In this way, different spectra can get the same identifier. In some operations, the program will practically ignore the second one (and eventually the third one, etc.) of such spectra unless the first one (and eventually the second one, etc.) is renamed (see below at (2b1)).

(2b) There are three particular cases of correcting parts of the spectra. Several ones of them may be applied to the same spectrum. Therefore, the first line for each of the corrected spectra contains:

* j, n

where n is the total number of corrections to be applied to spectrum j.

(2b1) The spectrum is renamed by putting a third integer in the first line mentioned above such as

* j, n, j'.

Then the spectrum identifier will be changed from j to j'. It is noted that this correction is not taken into account in n. Consequently, n should be given as 0 if nothing else than the spectrum identifier is corrected.

(2b2) If the contents of individual channels need to be changed, then the first line contains

n1, n2, n3

in each such correction where

n1 - is the number of the first corrected channel,
n2 - is the total number of the channels to be deleted,
n3 - is the total number of new channels (i.e. the number of the data to be inserted in place of the deleted ones).

In the next line, this is followed by the n3 new values (only if n3 > 0).

(2b3) Parameters x_1 and Δx can be corrected according to the scheme (2b2) by giving n1 = 0 followed by the new values of x_1 and Δx (the latter in a new line). The values of n2 and n3 are irrelevant in this case.

Remarks concerning the correction of spectra:

- In relation to a material, only one correction type is allowed (i.e. either according to scheme (1) or to scheme (2)).
- In relation to a spectrum, either scheme (2a) or scheme (2b) may be applied and each only once.
- Although any integer number is accepted by the program as a spectrum identifier, the individual spectra have to be corrected in the same order as they are stored in the subfile. The new spectra will be included after the existing ones.
- If reference is made to a spectrum wich does not exist in the subfile for the specified material, an error message is generated.
- Special remarks apply to correction schemes (2b1), (2b2), (2b3):

  * the total number of channels need not be corrected since its new value is computed by the program as a result of the performed corrections;
  * independently of how the channels are deleted and new ones are inserted by the successive corrections, parameters n1, n2, and n3 should be given according to the original state of the spectrum;
  * if n2 = 0, the n3 new channels are inserted preceding channel n1; if new channels have to be added after the last channel, there is no other way than to delete it and reinsert it followed by the new channels;
  * the successive corrections have to go forward within the spectrum: the value of parameter n1 should be at least equal to the value the sum (n1 + n2) had in the previous correction;
  * according to the previous remark, an eventual correction of type (2b3) should preced all corrections of type (2b2);
  * the channels to be deleted may not go beyond the existing ones.

Table 4.1. Input formats for corrections (except matrices and spectra)

Notations:

N - question number
K - material number

N.B. In the last column of this table, the expression "the number of the" (or "the numbers of the") has to be inserted. For example,
"n1: 1st line to be deleted"
should be read as
n1: the number of the 1st line to be deleted.

question number	1st line	2nd line	further data	parameters: number of...
1 to 60	*N, n1, n2	-	text data as for NEW	n1: 1st line to be deleted n2: lines deleted
61,62,64, 65,81	*N, n1, n2	list of mat. No. to be deleted (if n1>0)	data as for NEW	n1: materials to be deleted n1: new mats.
63	*N, n1	-	(K, N_K, M_K); n1 times	n1: new mats.
68 to 72, 82	*N, n1	(K, L_k) n1 times	as for NEW depending on L_K	n1: new mats.

Table 4.2. Input formats for correcting matrices
(Questions 66 and 83)

Notations:

n - total number of corrections in the step
K - material number
matrix head - total number of rows, total number of columns, list of column numbers

N.B. In the last column of this table, the expression "the number of the" (or "the numbers of the") has to be inserted. For example,
"i1: 1st row corrected"
should be read as
i1: the number of the 1st row corrected.

type of the step	1st line	2nd line	further data	parameters: number of ...
(1) whole matrices	K or **K and matrix head	-	the matrix row by row	
(2) rows	*K*n	-	i1,i2,i3, and the new rows	i1: 1st row corrected i2: deleted rows i3: new rows
(3) sub-matrices	*K**n	-	i1,i2,i3, (l_j, j=1, 2,...,i3) and the new sub-matrix	i1: 1st row corrected i2: corrected rows (>0) i3: corrected columns (>0) l_j: cols. to be corrected
(4) columns	*K*n1*n2	numbers of the deleted and the new columns	the sub-matrix of the new columns	n1: deleted columns n2: new columns
(5) permu-tation of columns	*K, (l_j, j=1, 2,...,n_c)	-	-	l_j: column where the original column j is moved n_c: columns (not given since known)

Table 4.3. Input formats for correcting Question 78

Notations:

K - material number
matrix head - total number of columns, list of column numbers

N.B. In the last column of this table, the expression "the number of the" (or "the numbers of the") has to be inserted. For example,
"n1: deleted columns"
should be read as
n1: the number of the deleted columns.

type of the step	1st line	2nd line	parameters: number of ...
(1) whole matrices	K or **K and matrix head	-	
(4) columns	*K*n1*n2	numbers of the deleted and the new columns	n1: deleted columns n2: new columns
(5) permutation of columns	*K, $(\ell_j, j=1, 2,\ldots,n_c)$	-	ℓ_j: column where the original column j is moved n_c: columns (not given since known)

Table 4.4. Input formats for correcting matrices (Question 79)

Notations:

n - number of corrections in the step
K - material number

N.B. In the last column of this table, the expression "the number of the" (or "the numbers of the") has to be inserted. For example,
"i1: 1st row corrected"
should be read as
i1: the number of the 1st row corrected.

type of the step	1st line	further data	parameters: number of ...
(1) whole matrices	K or **K	the matrix row by row	
(2) rows	*K*n	i1,i2,i3, and the new rows	i1: 1st row corrected i2: deleted rows i3: new rows
(3) sub-matrices	*K**n	i1,i2,i3, (ℓ_j, j=1, 2,...,i3) and the new sub-matrix	i1: 1st row corrected i2: corrected rows (>0) i3: corrected columns (>0) ℓ_j: cols. to be corrected
(4) columns	*K*n1*n2	the sub-matrix of the new columns	n1: deleted columns n2: new columns

Table 4.5. Input formats for correcting spectra (Question 80)

Notations:

K - material number
j - spectrum identifier

N.B. In the last column of this table, the expression "the number of the" (or "the numbers of the") has to be inserted. For example,
"n1: corrected spectra"
should be read as
n1: the number of the corrected spectra.

type of the step	1st line for the mat.	1st line for the sp.	further data	parameters: number of ...
(1) whole materials	K or **K, number of spectra	-	data for mat. K as for NEW	
(2) some of the spectra	*K, n1, n1			n1: corrected spectra n2: new spectra
(2a) whole spectra	-	j or **j, further data as for NEW	see at task NEW	
(2b) parts of a spectrum	-	*j, n (,j')	n1,n2,n3, contents of the new channels (n3 data) repeated n times	n : changes n1: 1st corr. channel n2: deleted channels n3: new channels

5. Subfile manipulations

In practical work with the stored data, several operations need to be performed in which the subfiles appear as undivided items. Program RFIT provides for a large variety of such tasks. Before their discussion, it is necessary to explain the role and overall structure of the files composing PDF. These tasks and files are summarized in Tables 5.1 and 5.2, respectively.

5.1. The files composing PDF

PDF consists of two files: the directory file and the data file. In the described version of program RFIT, the logical number of the former is 80 while that of the latter is 81. Both are direct access disk files. The record length is 9 words for file 80 and it is 20 words for file 81 (i.e. 36 and 80 bytes, respectively).

In the directory file, one record corresponds to each subfile. The significance of the 9 words is the following:

- the first 3 words contain the subfile identifier (cf. Section 2.1);
- the next 3 words contain the experimenter's name as given at Question 3 (cf. Section 2.2.2);
- the 7th and 8th words contain the numbers of the core certificate and core map, respectively, as given at Question 5 (cf. Section 2.2.2);
- the 9th word contains the number of the first record corresponding to the subfile within file 81.

This information is used for implicite subfile selection (see Section 5.2.2). The maximum number of records allowed for file 80 is 3000.

For each subfile, the significance of the individual records within the data file (i.e. file 81) is the following:

- the first record contains the subfile identifier, the total number of the questions for which the subfile contains data, and the total number of records corresponding to the subfile;

- the next 10 records contain the question numbers and the total numbers of the records corresponding to each of them;
- the 12th and the following records contain the data given at the individual questions.

It follows from this that the minimum number of records corresponding to a subfile is 12 while practically there is no upper limit to it since it is limited only by the size of file 81 which is 130000 records (for the described version of program RFIT).

Files 80 and 81 will be referred to as permanent disk files. Originally it was for safety reasons that two backup copies of these disk files were prepared on magnetic tapes and tasks were provided for restoring the disk files from one of these magnetic tapes. The logical numbers of the latter files are 60 and 61. In the EC-1040 version of program RFIT (cf. Refs. [2] and [3]), the copies to files 60 and 61 were mandatory since this was the only way of preventing the loss of PDF due to the rather frequent malfunctions of the disk drives available at that computer. Of course, once copies exist on magnetic tapes, it is not expedient to consider them only as backups since otherwise they allow to perform such operations which would be rather difficult or even impossible without the tapes. The described version of the program runs on a rather safe computer which allows to leave the preparation of the backup copies to the user's consideration. In accordance with this, the additional uses of files 60 and 61 become further emphasized.

Files 60 and 61 are identical and both are combinations of files 80 and 81: corresponding to each subfile, the first record is taken from directory file 80 which is followed by the records taken from data file 81. The last word of each directory record is not copied to the tapes since it has no significance for them. It follows from this that the record length is variable on the tapes: these files are mixtures of records of 8 and 20 words in length.

Most subfile operations are performed in temporary disk files which are scratched and released upon terminating the job step. The main reason of this is that they allow to protect the permanent disk files from eventual user errors. Both new and corrected subfiles are first put to these temporary files and they are recorded to the permanent (disk and tape) files only if the user deliberately tells the program to do so. This allows the user to check his/her input data first and to make the subfile permanent only if he/she is sure that it is correct. As pointed out in Sections 3.1 and 4.1, subfiles with structural errors cannot be created even on the temporary files. Consequently, an eventually faulty subfile can not make any harm to the other subfiles. The point of this cautious approach is rather that the deletion of subfiles from the permanent files is a lengthy and costly operation (especially if the faulty subfile has already been copied to the magnetic tapes).

There are 4 temporary files:

- File 77 may contain 50 subfiles. The maximum number of records on it is 15000. It has the same structure as file 81. The corresponding directory data are in the computer memory. It is involved in most subfile operations but is used also for manipulating CLIB and EDF (see Chapter 7).
- Files 74 and 75 may contain only one subfile each. They have the the same structure as file 81. They may contain 1000 records at maximum. They are needed only for task CORR (see Chapter 4). Furthermore, file 74 is used also if some options of the implicite subfile selection is applied to files 60 or 62 (cf. Section 5.2.2).
- File 76 is mainly used for fitting tasks RFIT and EVAL (see Chapter 5 of Part 2 and Chapter 6, respectively). When manipulating PDF, its use is rather limited (see Section 5.2.2). It is composed from records of 20 words (as most files used by program RFIT) whose total number may be 5000 at maximum.

5.2. Subfile selection

There are two ways of selecting those subfiles to which the individual subfile operations are applied: explicite and implicite. It can be seen from Table 5.1 which task keywords may or must go with subfile selection. There are tasks which allow both ways of subfile selection but their parallel application is prohibited in connection with a given task keyword.

5.2.1. Explicite subfile selection

The explicite subfile selection consists in giving the subfile identifier directly. With the exception of task NEW (which creates the subfile), all tasks manipulate (correct, list, copy, etc.) subfiles stored in the files. In case of tasks involving existing disk files, the general rule is the following (but there are exceptions to it, see the discussion of the individual tasks): an explicitely selected subfile is searched in file 77 first and the program turns to permanent files 80 and 81 only if there is no subfile with the given identifier in file 77. Both in file 77 and the permanent files, there may be subfiles with a common identifier. If so, the program selects the first one of them and ignores all of the other ones. There are two ways of selecting the latter explicitely: either by renaming the former or by repeating the common identifier as many times as there are such subfiles. This problem is related only to the explicite subfile selection mode. In case of implicite subfile selection, subfiles having the same identifier can be reached without difficulty. Further details of all this are presented in connection with the individual task keywords.

The explicite subfile selection may be used in connection with eveery task keyword if indeed they allow or require subfile selection. (The same is not true for the implicite subfile selection, cf. Table 5.1.) There are task keywords which, according to the sense, allow only explicite subfile selection. These are the

following:

- TasK NEW , since it creates the subfile from scratch, consequently, the information needed for implicite selection is not available yet.
- TasK RENM (i.e. rename) since it involves nothing else than the subfile identifier (cf. Section 5.3.3).
- TasKs RNM7 and DEL7 since they operate only on file 77 for which no implicite subfile selection is accepted at all (cf. Sections 5.3.3 and 5.4.4).

In case of explicite subfile selection, the identifiers of the selected subfiles are noted in the computer memory. The array available for this is dimensioned to 20 items at maximum. That is why the total number of explicitely selected subfiles is generally restricted to 20 in connection with one tasK Keyword. This set of subfile identifiers applies to all tasK Keywords which are given in the same input line while all tasK Keywords which are given in separate and later input lines delete the subfile identifiers noted for the previous tasK Keywords.

5.2.2. Implicite subfile selection

There are cases when it is not expedient or is even impossible to specify subfile identifiers i.e. to select subfiles explicitely. The simplest example of this is the case when one just wishes to Know which subfiles exist in PDF satisfying certain criteria. Program RF1T provides for the implicite subfile selection in view of such cases. It is based on the information stored in the subfiles.

The general form of the implicite subfile selection is either

ALL

or

ITEM A K1, K2

where

- the first letter of Keywords ITEM or ALL should be given in the leftmost position of the input line;
- ALL means the selection of all subfiles in PDF;
- A stands for a letter specifying the characteristics according to which the subfiles will be selected; the possible values of A are discussed below and are summarized in Table 5.3;
- K1 and K2 are parameters the role of which depends on the actual value of A; in some cases, further data are needed (see below).

Parameters A, K1, and K2 should be given in the same line as Keyword ITEM.

The possible values of A are the following:

M means subfile selection according to the type of measurement defined by the first 4 characters of the subfile identifier. For parameter K1, the required measurement type should be given which has to be one of the symbols listed in Table 2.1. Parameter K2 has no role in that case, consequently, it may be omitted.

N means subfile selection according to the experimenter's name as given at Question 3. Following the letter N, the required name should be given (separated by one or more spaces from N). Parameters K1 and K2 will contain the 8 first characters of it. Of course, the specified name may be shorter than 8 characters. The program will select all subfiles for which the specified group of characters appears in words 4, 5, and 6 of the directory file (cf. Section 5.1). This flexible way of searching the experimenter's name had to be chosen instead of a simple comparison of the full names mainly in view of the uncertain transcription of Russian names with English characters. For example, the same Russian name has been found in the following forms: EpanechnikOv, JepanecsnyiKov, YepanechniKow. It follows from this that the input line

ITEM N YEPANECHNIKOW

would not result in the required subfile selection since the subfiles with different transcriptions would not be selected. That is why the safest approach is to give only

ITEM N EPANE

which contains only the common part of the mentioned transcriptions.

D means subfile selection according to the date given in characters 5 to 10 (incl.) of the subfile identifier. The values of parameters K1 and K2 give the first and last dates of the required time interval, respectively. All subfiles will be selected the dates of which are within the specified closed time interval (i.e. including the limiting dates).

K means subfile selection according to the core configuration. The values of parameters K1 and K2 give the numbers of the core certificate and core map, respectively. Each of them may be omitted (but not both at the same time). If so, the subfile selection will correspond only to the specified one. If K2 is omitted, the format is

ITEM K K1

while K1 is omitted in the following way:

ITEM K /K2

X means subfile selection according to the position of the subfile within the directory file. The values of parameters K1 and K2 give the first and last positions of the required range, respectively. As described in Section 5.3.1, the directory data of the

subfiles can be printed, and the printed items are numbered. This way of subfile selection makes reference to that numeration. The subfile selection will correspond to the specified closed interval (i.e. it will include the limiting numbers K1 and K2).

R means subfile selection according to which questions are answered in the subfile. The value of parameter K1 gives the total number of questions concerned. Parameter K2 is irrelevant in this case and should be omitted. Following K1, the numbers of the questions concerned (i.e. K1 integers) are given (which may be continued in the next lines if necessary). Sometimes, the list of the question numbers would be rather long. This can be simplified if it contains all numbers belonging to some intervals. Then the program allows to give only the ends of the intervals separated by asterisks (*). For example,

```
ITEM  R  6  63  78  1*10  21*40
```

means Questions 63, 78, 1 to 10 (incl.), and 21 to 40 (incl.). Note that the value of K1 is the total number of integers actually given (i.e. 6) and not the total number of the questions defined by these input data (which would be 32 in the example at hand). Note that here, exceptionally, the question numbers need not be given in ascending order. All subfiles will be selected in which at least one of the specified questions are answered. The program accepts negative question numbers as well: they mean the selection of those subfiles in which the specified questions are not answered. For example, the following input data

```
ITEM  R  3  2  -5  22
```

result in the selection of all subfiles in which either Questions 2 or 22 are answered, for one hand, or Question 5 is not answered, for the other hand. Negative question numbers may not be given in fc .tervals.

C means subfile selection according to the information stored in the subfiles: those subfiles will be selected in which some specified information is stored at specified Questions. This way of subfile selection may be related only to Questions 1 to 60 (incl.). The character of the information searched for subfile selection can be either a string or an integer or a real number. It depends on the particular question which of them is applicable. At Questions 7, 8, and 22, only an integer can be searched but two cases are possible for the other questions as shown by Table 5.4 and explained consecutively.

Table 5.4. Cases for subfile selection mode ITEM C

Question	case 1	case 2
5	integer	real number
7, 8, 22	integer	-
9	integer	string
text	string	real number

For the individual questions, these cases are the following:

- for Question 5, the integer is compared with the numbers of the core certificate and core map while the real number is compared with the moderator level, the boric acid concentration, and the nominal reactor power;
- for Question 9, the integer is compared with the calibration data set identifier while the string is compared with the material name;
- for text questions, the string is searched throughout the text available at them; when a real number is specified, the program reads any numeric data which are within the text and compares them with the specified real number.

In case of ITEM C, the value of parameter K1 gives the total number of such comparisons. Their relation to one another is the logical OR relation i.e. the subfile is selected if at least one of these comparisons results in a match for at least one of the questions specified for the individual comparisons. Parameter K2 is irrelevant in this case and should be omitted. Following the line

ITEM C K1,

further data defining these comparisons are given in separate lines. This means that the following is repeated K1 times:

- the question numbers are given first (see below) followed by
- the case number (i.e. 1 or 2 according to the table), finally
- the information to be matched is given.

There is no limitation as to how all these data are broken into lines except that the string should be finished in the same line as started and that the data for every comparison (test) should start in different lines. The question numbers are specified in a way similar to the case of ITEM R: the total number of these questions is given first followed by their numbers. Sometimes, the list of the question numbers would be rather long. This can be simplified if it contains all numbers belonging to some intervals. Then the program allows to give only the ends of the intervals separated by asterisks (*). For example, the input data

```
ITEM C 2
1  9  2  CU100.
6  13  18  1*10  21*40  1  COPPER ACTIVITY
```

define the following subfile selection criterium:

- material name 'CU100.' occurs at Question 9 and
- the string 'COPPER ACTIVITY' occurs at at least one of Questions 13, 18, 1 to 10 (incl.), and 21 to 40 (incl.).

Note that, in the last input line, 6 is equal to the total number of integers actually given and not the total number of the questions defined by these input data (which would be 32 in the example at hand). Note furthermore that here, exceptionally, the question numbers need not be given in ascending order. If, for the specified case number, different kinds of information would correspond to the specified questions (see Table 5.4), an error message is generated.

Except for the last two options (i.e. R and C), the information involved in the subfile selection is stored in the directory file 80. It follows from this that such options of the implicite subfile selection do not require too much computing time while the use of options R and C (especially the latter) may be rather time comsuming since they require the search of the subfile itself. This should be borne in mind when the input data are formulated. In addition to this, the following remarks should be taken into account:

- Keyword ITEM may be applied 5 times at maximum in connection with one task Keyword.
- The relation of the successive subfile selection conditions to one another is the logical AND relation: the subfile is selected if and only if all the specified conditions are fulfilled. Note that the different conditions belonging to the same ITEM C are in the logical OR relation to one another. This allows to compose quite sophisticated subfile selection criteria.
- As the successive selection criteria are evaluated, the program goes over to the next criterium only if the previous ones did not result in the rejection of the subfile. It follows from this that it is not neutral in which order the criteria are given in input. Those should be given first for which the odds of rejecting the subfiles are the greatest. Anyhow, the last ones should possibly be ITEM R and/or ITEM C.

At the end of the present chapter, a number of examples are given which help to understand the use of the various options of implicite subfile selection.

5.3. Subfile inventory

The knowledge of the subfile inventory is an important prerequisite of manipulating subfiles. The tasks are the following:

- In relation to the permanent disk files, tasks SUBF and SUBT allow to print out the directory data of all or some selected subfiles. The same can be done in relation to the subfiles available on the magnetic tapes by using task SFMT.
- For some subfile operations, it is necessary that the contents of the permanent disk files and the magnetic tapes were identical. Task Keyword COMP allows to check whether this is really the case.
- Finally, tasks RENM and RNM7 serve for changing the subfile identifiers. The former operates on the permanent disk files while the latter on temporary file 77.

5.3.1. Tasks SUBF and SUBT

Tasks SUBF and SUBT allow to print out the directory information (see Section 5.1) concerning the subfiles stored in the permanent disk files. Remarks on the use of these tasks:

- Both explicite and implicite subfile selections are accepted. If the subfile selection is omitted, an error message is generated.
- With task Keywords SUBF and SUBT, the explicite subfile selection is practically equivalent to checking whether the selected subfiles exist and it locates them within PDF.
- The difference between tasks SUBF and SUBT consists in that the former ignores those subfiles which have been deleted by the use of task DDEL while the latter takes all subfiles into account. For the deleted subfiles, the number of the first record within the subfile is printed with a negative sign.
- The printouts prepared by these tasks define the numbers which locate the subfiles for implicite subfile selection mode ITEM X (cf. Section 5.2.2).

5.3.2. Task SFMT

Task SFMT allows to print out the directory information (cf. Section 5.1) concerning the subfiles stored in magnetic tape 60. Remarks on the use of this task:

- Both explicite and implicite subfile selections are accepted. If the subfile selection is omitted, the program considers that all subfiles should be selected.
- With task Keyword SFMT, the explicite subfile selection is practically equivalent to checking whether the selected subfiles exist and it locates them within the tape.
- If ITEM R and/or ITEM C are used for implicite subfile selection, all subfiles are first copied to file 74 and the selection criteria are applied to it only after this. (This circumstance makes the operation rather time comsuming if the number of the subfiles is large in PDF.)
- The printouts prepared by this option define the numbers which locate the subfiles for implicite subfile selection mode ITEM X (cf. Section 5.2.2).

5.3.3. Tasks RENM and RNM7

Tasks RENM and RNM7 are used for renaming the subfiles i.e. for changing their identifiers. According to the sense, only explicite subfile selection is permitted with these task Keywords. The general form of the input data is

```
RENM (or RNM7)
1. old subfile identifier
1. new subfile identifier
2. old subfile identifier
2. new subfile identifier
  ....
```

and so on until the next task Keyword. Within each pair of successive "old" and "new" subfile identifiers, the subfile having the "old identifier" is found and its identifier is changed for the corresponding "new" one. Remarks concerning the use of these tasks:

- If a subfile to be renamed is not found, an error message is generated but this does not influence the renaming of those subfiles which could be found.
- The program notes the new (renamed) subfile identifiers and their list is available for other tasks given in the same line as RENM.
- In connection with one task Keyword RENM (or RNM7), 10 subfiles can be renamed at maximum. Would this not be sufficient, the task Keyword should be repeated after the 10th subfile.
- An error message is generated and no subfile is renamed if the total number of the specified subfile identifiers is odd.
- Differences between RENM and RNM7:
  * In case of RENM, the subfiles are searched in the permanent disk files while, in case of RNM7, they are searched in temporary file 77.
  * Application of task RENM creates a list of subfiles which can be used later by task EVAL (see Chapter 6). This is not the case for task RNM7.
  * In case of RENM, the renamed subfiles are copied to temporary file 77 while, in case of RNM7, the renamed subfiles remain in file 77 in their original places. It follows from this that the application of task RENM does not change the state of the permanent disk files, unless the renamed subfiles are not recorded to them (cf. Section 5.4.1). The most effective application of task RNM7 is the renaming of the corrected subfile versions created by task CORR (see Chapter 4).
  * Would several subfiles have the same identifier, the first one of them is renamed. As to the other ones, the situation is different for these tasks. In case of RNM7, the others can be reached by repeating the common identifier among the "old" ones since the state of file 77 changes after the first subfile has been renamed. In case of RENM, however, the other subfiles cannot be reached since, as said above, the renaming of the first subfile does not change the state of permanent files 80 and 81 unless the renamed subfile is recorded to files 80 and 81 and then the original subfile is deleted

(e.g. by task DDEL, see Example 5 in Section 5.6). That is why one had better avoid the presence of subfiles with common indentifiers in the permanent files.

5.3.4. Task COMP

The use of some tasks involving the copying of subfiles to magnetic tapes requires that their contents were identical to one another and to the permanent disk files. Task COMP allows to check this. It is used alone i.e. no subfile selection may go with it. If differences are found between the permanent disk and one of the tapes, an error message is generated and the execution of the program is terminated.

The comparison of the subfiles does not go into the details of the information stored in them. It is restricted

- to the directory data of the subfiles (cf. Section 5.1) and
- to the total number of records stored in the subfiles.

5.4. Copying and deleting subfiles

Program RFIT allows to copy subfiles

- from temporary file 77 to the permanent disk and tape files (tasks RCRD and DREC);
- from the permanent disk files to the tapes (tasks DEL and COPY);
- from a tape to the permanent disk files and to another tape (tasks REST, DADD, and ADD).

Subfiles can be copied to the permanent files only after they have been initialized by tasks DMMY and DDMY. There are several ways of deleting subfiles:

- task DDEL deletes subfiles only from the permanent disk files; the result of this operation can be reversed by task UNDL;
- task DEL deletes subfiles both from the permanent disk files and tape 80;
- task DEL7 allows to delete subfiles only from file 77.

It can be seen from this introduction that there are pairs of tasks for some of the operations (RCRD/DREC, DEL /DDEL, DMMY/DDMY, and ADD /DADD). Those which are differentiated by an additional letter D in the task keyword affect only the permanent disk files while the other ones change the contents of tapes 80 and 81, too.

5.4.1. Tasks DMMY and DDMY

Most operations which copy subfiles to the permanent disk and tape files assume that some subfiles already exist in them. It

follows from this that PDF needs to be initialized before any data can be included into it. That is the role of tasks DMMY and DDMY which create a dummy subfile identified as

MM01010100

and containing only the minimum number of (i.e. 12) records. Tasks DMMY and DDMY differ from one another by that the latter puts this dummy record only to disk files 80 and 81 while the former puts it to tapes 60 and 61 as well. Remarks concerning their use:

- They are used only at the very beginning of the work with PDF. When, later on, some real subfiles will have been recorded to PDF, this dummy subfile can be deleted from it.
- Task keywords DMMY and DDMY are used alone i.e. no subfile selection may go with them.
- Tasks DMMY and DDMY scratch all information eventually stored in permanent disk files 80 and 81. Task DMMY scratches all previous information also from tapes 60 and 61. Consequently, one has to be very cautious with their use.

5.4.2. Copying subfiles from file 77 to the permanent files

(tasks RCRD and DREC)

Tasks NEW , CORR, and RENM create subfiles on temporary file 77. They can be copied to the permanent files by using tasks DREC and RCRD. The former affects only disk files 80 and 81 while the latter copies the same subfile also to tapes 60 and 61. Apart from this, their effects are the same as formulated below in terms of task RCRD:

- In the first step, each subfile is copied from file 77 to disk files 80, 81, and tape file 60. Copying of the next subfile starts only after that the previous one has been copied fully to tape 60 and a message about this fact has appeared in the line printer output.
- When all subfiles have been copied from file 77 to files 80, 81, and 60, they are copied to tape 61 from files 80 and 81.
- Finally, the directory of file 77 is scratched.

As said above, no copy is made to tapes 60 and 61 in case of task DREC.

The new subfiles are copied behind the existing ones but their relative sequence remains the same as in file 77. When a new subfile is included into the permanent files, the program does not check whether their identifiers are different from those of the existing ones. It is noted finally that no subfile selection may go with tasks DREC and RCRD.

5.4.3. Copying subfiles from the tapes to the permanent files

(tasks REST, ADD , and DADD)

There are three slightly different ways of copying

subfiles from magnetic tapes to the permanent disk files: tasks REST, DADD, and ADD .

In case of task REST, the permanent disk files are restored from magnetic tape 60. Remarks concerning its use:

- Both explicite and implicite subfile selection modes are accepted. If no subfile selection is specified, the program considers that all subfiles needs to be selected.
- When task REST is used for the first time within the job step, all information eventually existing in files 80 and 81 is overwritten. It follows from this that the use of task REST need not be preceded by the use of task DMMY (cf. Section 5.4.1) and the same precaution is advisable as in case of DMMY.
- When task REST is used several times within a job step, the subfiles selected by the second, third, etc. uses of task REST do not overwrite the existing information but simply continue the sequence of the previously restored subfiles.
- The program does not check whether the identifiers of the copied subfiles are different from those of the existing ones. Furthermore, the program does not check whether the subfile identifiers to be copied belong to those listed in Table 2.1.

In case of task DADD, the operations are almost the same as in case of REST. The differences are the following:

- Task DADD acts as the second and repeated applications of task REST: it copies subfiles from the magnetic tape behind the subfiles already existing on disk files 80 and 81.
- The logical number of the tape file file from which the copy is made is 62.
- The program checks whether the subfile identifiers to be copied belong to those which are listed in Table 2.1.

Tasks DADD and ADD differ from one another only by the simple fact that the latter copies the subfiles not only to disk files 80 and 81 but also to tapes 60 and 61 as a continuation of the subfiles already existing on the tapes. It follows from this that operation ADD can be successfull only if the subfile structures on disk files 80 and 81, for one hand, and on tapes 60 and 61, for the other hand, are identical. This is a typical case when the application task COMP is desirable before one can go over to the application of these tasks (cf. Section 5.3.4).

5.4.4. Deleting subfiles from the disk files

(tasks DEL , DDEL, DEL7, and UNDL)

Subfiles can be deleted directly only from the disk files. Task DEL7 operates on temporary file 77 while tasks DEL , DDEL, and UNDL operate on permanent disk files 80 and 81. The use of task DEL involves also magnetic tape 60. Correspondingly, its use is discussed also among the respective copying tasks (cf. Section 5.4.5).

Let us discuss the use of task DDEL first. It serves for deleting subfiles from the permanent disk files 80 and 81. The

records themselves which belong to the selected subfiles are not deleted from the disk physically but they are only marked: for the selected subfiles, the numbers stored in the 9th words of the directory file records are changed to negative. The effect of this is that such subfiles will not be selected afterwards neither explicitely nor implicitely. The only exceptions to this are tasks SUBT and UNDL. The former is discussed in Section 5.3.1. Task UNDL is the inverse operation of DDEL. Actually, the numbers stored in the 9th words of the directory file records are changed to positive for the selected subfiles thus making them available later on for all tasks. Remarks to the use of tasks DDEL and UNDL:

- Both explicite and implicite modes of the subfile selection are accepted. If no subfile selection is specified, this generates an error message.
- The selected subfiles are searched only in the permanent disk files 80 and 81 even in case of explicite subfile selection.
- If some of the subfiles selected explicitely are not found on the disk, this generates an error message but this does not make the DDEL or UNDL operations ineffective for those subfiles which could be found.

The effect of task DEL is much more drastic than that of task DDEL: the selected subfiles are scratched in such a way that the subfiles selected for task DEL are physically lost. The program preceeds in two steps: first, all subfiles which are not selected are copied from the permanent disk files 80 and 81 to tape 60 (overwriting the previous contents of this tape); after having finished this, the contents of tape 60 are copied to disk files 80 and 81 (overwriting their original contents). The second step is actually equivalent to restoring all subfiles (i.e. to the first use of task REST without subfile selection, cf. Section 5.4.3). Remarks to the use of task DEL :

- Both explicite and implicite modes of the subfile selection are accepted. If no subfile selection is specified, this generates an error message.
- The selected subfiles are searched only in the permanent disk files 80 and 81 even in case of explicite subfile selection.
- No copy is made automatically to tape 61. This is done only if task keyword COPY is given in input (cf. Section 5.4.5).
- If some of the subfiles selected explicitely are not found on the disk, this generates an error message but this does not make the deletion of those subfiles ineffective which could be found. As a matter of fact, the deletion of the subfiles marked by task DDEL can be made definitive by a DEL operation if no subfiles are selected (e.g. by specifying a subfile identifier which does not exist in the permanent disk files).
- The contents of the permanent disk files change only in the second step i.e. when copying to tape 60 has been finished. It is important to take this into account when the job is terminated due to the time limit specified for the job. The program's output is clear in this respect:
  * while the subfiles are copied to tape 60, the program prints the identifiers of those which are deleted; if some of them were marked previously by task DDEL, they are lost now;

during this operation, the permanent disk files are not changed;
* this is finished when the notice

THE FOLLOWING SUBFILES HAVE BEEN RESTORED FROM MT:

appears in the line printer output; the identifiers of the subfiles copied to the permanent disk files are given in the line printer output.

It can be seen from this that whenever the job is terminated, we always have a version of PDF from which the subfile manipulations can be continued (either from the original disk files or from tape 60).

- As task DEL involves a rather lengthy operation (especially when there are many subfiles in PDF), it is advisable to be economical with its use. Normally, only task ODEL is meant for everyday use in connection with PDF.

In view of the difficulties connected with the use of tasks DEL , ODEL, and UNDL, it is better to record only such subfiles to the permanent files which need not be deleted from them afterwards. That is why it is recommended to delete the useless subfiles already from temporary file 77 before they are recorded to the permanent files. That is the role of task DEL7. Remarks to its use:

- Only the explicite mode of of the subfile selection is accepted. If no subfile selection is specified, this generates an error message.
- The effect of task DEL7 is restricted to temporary file 77. The permanent files are not affected at all.
- If some of the subfiles selected explicitely are not found on the disk, this generates an error message but this does not make the deletion of those subfiles ineffective which could be found.

The use of task DEL7 is illustrated by Example 2 of Section 5.6.

5.4.3. Copying subfiles from the permanent disk files to the tapes

(tasks DEL and COPY)

As discussed in Section 5.4.4, the use of task DEL results in a copy of the permanent disk files to tape 60 which is copied back to the disks. When this backward copy is not needed, task COPY should be used. Its effect depends slightly on whether some subfile selection has been specified in the input data or not. The main difference consists in that the copy is made to tape 61 if no subfile selection is specified while the subfiles are copied to tape 62 in the opposite case.

Let us consider the case of copying to tape 61 first. This task is primarily meant for preparing backup copies of PDF. That is why its use is as simple as possible: the input data

```
COPY
END
```

are sufficient for obtaining a backup tape 61. Copying is finished when the notice

A FULL COPY HAS BEEN MADE

appears in the line printer output. (The identifiers of the copied subfiles are not printed.) Task COPY is most frequently used in connection either with task DEL or with task REST. In view of their importance, these cases are discussed separately. (In the following examples, task Keyword END could be replaced by any other task Keyword without changing the effect of task Keyword COPY if the job is continued with other operations.)

The input data

```
REST COPY
END
```

result in the following operations:

- all subfiles are copied first from tape 60 to disk files 80 and 81 (destroying their original contents),
- then a backup copy is made to tape 61 from the disk files (destroying the original contents of tape 61).

If some subfile selection is specified in this example, both operation REST and COPY will affect only the selected subfiles and the copy will be made to tape 62. This remains true even if all subfiles are selected: the input data

```
REST COPY
ALL
END
```

result in the same operations as in the previous example but the copy will be made to tape 62.

Things are slightly different if task COPY is used in connection with task DEL since some kind of subfile selection must be specified with DEL . Let us suppose that the input data are the following:

```
DEL  COPY
MA0100100
END
```

The operations will be:

- all subfiles but the specified one will be copied to tape 60;
- the content of tape 60 will be copied to disk files 80 and 81;
- a backup copy will be made to tape 61.

As to this last example, it should be noted that the second step is equivalent to a REST operation without subfile specification (cf. Section 5.4.5). That is why the backup is made on tape 61 (and not on tape 62).

The backup copies made in these examples are always identical with tape 60. If, on the contrary, some subfile selection is specified for task COPY, the result can be different from tape 60. That is why the output is tape 61 in the former case and it is tape 62 in the latter one. The following remarks apply to copies made to tape 62:

- Both explicite and implicite subfile selections modes are accepted.
- In contrast to the case of copies made on tape 61, the identifiers of the selected subfiles are printed in the line printer output.
- When task COPY is used for the first time within the job step, all information eventually existing on tape 62 is overwritten.
- When task COPY is used several times within a job step, the subfiles selected by the second, third, etc. uses of task COPY do not overwrite the existing information but simply continue the sequence of the previously copied subfiles. This is analogous to the case of task REST. It follows from this that tasks REST and COPY may be used together several times in sequence: the contents of both the permanent disk files and tape 62 will be identical.

5.5. Listing the contents of subfiles

(tasks LIST, PLST, and CLST)

The contents of some selected subfiles can be printed by using task LIST or its versions PLST and CLST. In case of task LIST, the whole subfile is printed while only the information stored at part of the questions is printed for PLST and CLST. The question numbers concerned are determined as follows:

PLST: they are given in input;

CLST: they are the numbers of the questions which have been corrected during the last use of task CORR.

When task keywords LIST, PLST, and CLST are given alone, the format of the input data is the following:

```
LIST (or CLST or PLST)
  :
  :
data for subfile selection
  :
  :
data for specifying the question numbers (only for PLST)
```

The specification of the question numbers is not necessary in case of tasks LIST and CLST. For task PLST, the question numbers are specified in the following way: first the total number of the questions is given which is followed by the question numbers. Sometimes, their list is rather long. It can be simplified if it contains all numbers belonging to some interval: then, the program

allows to give only the limits of the interval separated by an asterisk (*). For example, the input data

```
PLST
ITEM  D  010175  311275
ITEM  M  MA10
6  63  78  1*10  21*40
END
```

mean printing the information stored at Questions 63, 78, 1 to 10 (incl.), and 21 to 40 (incl.) in the selected sub iles. Note that 6 is here the total number of integers actually given and not the total number of the questions defined by these input data (which would be 32 in the actual example). Here, exceptionally, the question numbers need not be mentioned in ascending order. The input data belonging to task PLST should always start in a separate line but there is no further restriction as to how they are broken into lines.

When these listing tasks are used in connection with some other task (e.g. NEW , CORR or RENM) which go with some subfile selection, the listing task will be related to the same subfiles as the first task. In such cases, the use of tasks LIST and CLST does not require any additional input data. The structure of the input data for task PLST is illustrated by the following example related to this case:

```
NEW  PLST
  :
  :
input data for task NEW
  :
  :
++++
6  63  78  1*10  21*40
END
```

where only the input line preceding keyword END is related to task PLST. The program can recognize very easily where these data start: when the line containing ++++ is not followed by a subfile identifier, this is the beginning of the data for task PLST.

Remarks to the use of listing tasks LIST and PLST:

- Both the explicite and implicite modes of subfile selection are accepted for tasks LIST and PLST. If no subfile selection is specified, this generates an error message.
- The selected subfiles are searched only in the permanent disk files 80 and 81 in case of implicite subfile selection.
- In case of explicite subfile selection, the subfiles are searched first in temporary file 77 and the program turns to the permanent disk files only if the specified subfile identifier is not found in file 77. If there are several subfiles with the same identifier in any of these files, the program lists only the first of them.
- If some of the subfiles selected explicitely are not found on

the disks, an error message is generated but this does not prevent the listing of those subfiles which could be found.

According to the sense, the use of task CLST is somewhat restricted in relation to the other listing tasks. In order to make this clear, let us suppose that several subfiles have been corrected by using task CORR (see Chapter 4). The corrected versions of these subfiles are put to temporary file 77 and the numbers of the questions which have been corrected are noted individually for each corrected subfile. If task CLST is used together with this task CORR, the program will list the corrected information (i.e. the information stored in each corrected subfile at those questions which have been corrected in them). If, however, task CLST is used somewhat later, the following restrictions should be observed:

- Only the explicite subfile selection mode is accepted.
- The specified subfile identifiers should be identical to those of the corrected subfiles and should be given in the same sequence. The only allowance is that their total number may be less.

The action of the program will be the same as if CLST would have been given together with the corresponding task keyword CORR. Error messages will be generated if:

- no or implicite subfile selection is specified;
- one of the specified subfile names does not coincide with the sequentially corresponding subfile name in file 77;
- more subfiles are specified than corrected previously.

It is noted that such errors can occur not only by improper subfile selection data but there can be various other reasons, too: e.g. subfiles might have been deleted from file 77 between the uses of tasks CORR and CLST. Anyhow, the separate use of these tasks is exceptional and should be avoided in view these restrictions and sources of error.

It is noted finally that tasks NEW and CORR automatically generate a subsequent use of tasks LIST and CLST, respectively, if the former are the last keywords in the input line. Therefore, they should be accompanied by task keyword NLST (i.e. No LiST) if the subfile listings are not required:

NEW NLST

and

CORR NLST

This is not necessary if these task keywords are followed by some other ones such as for example

NEW EVAL

or

```
CORR DEL7
```

5.6. Examples for subfile manipulations

As a conclusion of this chapter, some illustrative examples of subfile manipulations are given. Most of them are trivial to those who are already acquainted with the use of program RFIT but they will hopefully provide some help for those who just want to understand the possibilities offered by the program.

Example 1. Creating and recording two subfiles

```
NEW  LIST RCRD
MA10050677
  :
  :
data for the first subfile
  :
  :
++++
MA10060677
  :
  :
data for the second subfile
  :
  :
++++
END
```

Explanation: The new subfiles are created first on temporary file 77. Then, their full contents are listed on the line printer output. Finally, both subfiles are recorded to permanent disk files 80 and 81, and tapes 60 and 61.

Example 2. Creating, correcting, recording, and listing two subfiles

Let us assume that the input data for the second subfile of Example 1 are available on a magnetic tape in card image format but this latter tape contains some errors to be corrected before the subfile may be recorded to the permanent files. Let the logical number of this input tape be 10. The structure of the corresponding input data is the following:

```
NEW  NLST
MA10050677
  :
  :
data for the first subfile (on SYSIN)
  :
  :
++++
PERI 10
MA10160677
  :
  :
data for the second subfile (on tape 10)
  :
  :
++++
PERI 5
CORR DEL7 RCRD
MA10160677
  :
  :
correction data for the second subfile (on SYSIN)
  :
  :
++++
LIST
MA10050677
MA10160677
END
```

Explanation: The effect of the use of task keyword NEW is analogous to the previous example with the exception that the automatic listing is now suppressed by the no operation keyword NLST. After the input data belonging to the first subfile, the reading of the input data is switched from SYSIN input to tape 10 from which the program creates the second subfile. After this, the reading of the input data returns to the SYSIN input. The data belonging to the second operation are introduced by task keyword CORR. When the corresponding task is accomplished, a third subfile appears in file 77 which has the same identifier as the second one. As the latter version of subfile MA10160677 is not correct, this has to be deleted from file 77 before the subfiles are recorded to the permanent files. In order to make it clear what

actually happens, let us think over the effects of the individual task keywords:

- The following subfiles are in file 77 after the correction:

  MA10050677
  MA10160677
  MA10160677

- The subfile selection for task DEL7 is determined by the list of the corrected subfile identifiers since keyword DEL7 is given in the same line as CORR. In the actual case, this list contains one item:

  MA10160677

- When task DEL7 is executed, one subfile identified by

  MA10160677

  is deleted from file 77. As usual in case of the explicite subfile selection, this will be the first subfile having this identifier i.e. the just the incorrect version of the subfile.
- Finally, the remaining two subfiles will be recorded to the permanent files.

The last operation is the listing of the two subfiles selected explicitely. As the previous operation RCRD scratches all information from file 77, the listing will be made from disk files 80 and 81. Of course, if the latter files contain subfiles having these identifiers, not these newly recorded subfiles but the older ones will be listed.

Example 3. Correcting some subfiles in the permanent files

Let us assume that the copper foils used for MI02 type measurements have been recalibrated on 10 September 1977 and the new calibration factors have been put to CLIB (see Chapter 7) as the data set identified by 3302. The identifier of the previous calibration data set is 3301. Assume furthermore that it is known from previous listings that copper is the third material in the subfiles created before the summer of 1978 while it became the second material later on.

```
CORR CLST DREC DDEL
ITEM  M  MI02
ITEM  D  100977  300678
ITEM  C  2
1  9  2  CU100.
1  9  1  3301
*9  3  1
9  3302  CU100.
++++
++++
CORR CLST DREC DDEL
ITEM  M  MI02
ITEM  D  010978  311299
ITEM  C  2
1  9  2  CU100.
1  9  1  3301
*9  2  1
9  3302  CU100.
++++
++++
END
```

Explanation: Both correction steps are related to MI02 type measurements (defined by the lines containing ITEM M MI02). The first step is related to measurements performed between 10 September 1977 and 30 June 1978. As usual in the practice of the ZR-6 measurements, the reactor has been stopped for the summer period. Consequently, the the second correction step is defined for the period after 1 September 1978. These correction steps differ from one another by that the calibration data set identifiers are corrected for the second and third materials in the first and second steps, respectively. Further remarks:

- In both correction steps, the correction data

```
*9  3  1
9  3302  CU100.
++++
++++
```

and

```
*9  2  1
9  3302  CU100.
++++
++++
```

are stored in file 76 and are applied for all of the selected subfiles.

- As an effect of task Keyword CLST, the answers to Question 9 are printed fo the corrected versions of the selected subfiles.
- The corrected subfile versions are recorded to and the original versions are deleted from permanent files 80 and 81 as results of giving task Keywords DREC and DDEL, respectively.
- Note that the order of using Keywords DREC and DDEL is not arbitrary: it is advisable to record the corrected subfiles before the original versions are deleted.

Example 4. Correcting and renaming subfiles in the permanent files

Suppose that the original version of one of the corrected subfiles need to be kept. Let it be the second one of three corrected subfiles. Then the input data look as follows:

```
CORR NLST
MI02151076
   input data for correcting this subfile
++++
MA05101075
   input data for correcting this subfile
++++
MI02171076
   input data for correcting this subfile
++++
RNM7
MA05101075
MA05101075-1
ODEL
MI02151076
MI02171076
OREC
END
```

Explanation: As results of the first step (i.e. CORR NLST), three subfiles are corrected without listing the corrected subfile versions, the corrected subfiles are created in temporary file 77. Further operations:

- The second one (i.e. subfile MA05101075) is renamed.
- The original versions of the first and third subfiles (i.e. subfiles MI02151076 and MI02171076) are deleted from permanent files 80 and 81.
- Finally, all corrected subfile versions are recorded to permanent files 80 and 81.

Example 5. Renaming subfiles having the same identifier

Suppose that we have 3 subfiles in our data library with the same identifier MA05101075. As explained in Section 5.2.1, only the first one of them can be reached in case of explicite subfile selection. This unpleasant situation can be changed in the following way:

```
RENM DREC
MA05101075
MA05101075-1
DDEL
MA05101075
RENM DREC
MA05101075
MA05101075-2
DDEL
MA05101075
END
```

Explanation: The first step (RENM DREC) puts the first subfile MA05101075 to file 77 under the name MA05101075-1 which is recorded (as an effect of task DREC) to permanent files 80 and 81. The subsequent use of task DDEL deletes the first subfile identified by MA05101075 from the permanent disk files. When this is repeated, the same is done with the second subfiles identifed by MA05101075 which gets recorded to the permanent disk files under the new identifier MA05101075-2 and gets deleted from the permanent disk files. It is noted finally that task keyword DDEL can not be put in the same line as RENM and DREC for the following reason. The renaming operation (i.e. RENM) produces a list of subfile identifiers which would be available for DDEL but it contains the renamed identifiers (i.e. MA05101075-1 or MA05101075-2) instead of the old ones (i.e. MA05101075). Consequently, not the original but just the renamed subfiles would be deleted.

Example 6. Compiling a reduced library of the stored data

Suppose that we have a large number of subfiles in our data library and we have obtained a magnetic tape from a partner research centre which contains some subfiles compiled there. We want to unite these two libraries and we are requested by a further institute to send them all subfiles related to core configurations 57 and 72. This can be done in two job steps. The corresponding input data are the following:

```
Job step 1:  COMP
-----------  ADD
             SUBF
             ALL
             END
```

Explanation: It is checked first (by task COMP) whether the contents of files 80 and 81, for one hand, and magnetic tapes 60 and 61, for the other hand, are identical. If this comparison turns out to be positive, the new subfiles are added to PDF (by task ADD). The added subfiles are read from tape 62. Finally, the subfile identifiers in the new PDF are listed (by using task SUBF).

```
Job step 2:  COPY
-----------  ITEM K 57
             COPY
             ITEM K 72
             END
```

Explanation: As subfile selection data are provided for task keyword COPY, the selected subfiles will be copied to tape 62 (cf. Section 5.4.3). That is is why these operations have to be performed in distinct job steps unless the original contents of the foreign magnetic tape may be destroyed. Note that the first use of task COPY opens the tape while the second use continues the copying at the point where the first copy will have been finished.

Table 5.1. Tasks for manipulating subfiles

N.B. See remarks given below the table identified by (1), (2), ... in the last column. The full list of the task keywords is given in Table 2.1 of Part 1. For keywords not mentioned in the present table, no subfile selection is taken into account.

task keyword	subfile selection: explicite	subfile selection: implicite	input file	output file	remark
ADD	yes	yes	62	80,81, 60,61	(6)
CLST	yes	no	80,81 77,75	line printer	(1), (2),(5)
COMP	no	no	60,61, 80,81	-	-
COPY	yes	yes	80,81	61,62	(4),(6)
CORR	yes	yes	SYSIN 77,80, 81	77	(1),(3)
DADD	yes	yes	62	80,81	(6)
DDEL	yes	yes	80,81	80,81	(1)
DEL	yes	yes	80,81	60,80, 81	(1)
DEL7	yes	no	77	77	(1)
DDMY	no	no	-	80,81	-
DMMY	no	no	-	80,81, 60,61	-
DREC	no	no	77	80,81	-
EVAL	yes	yes	77,80, 81	77	(3)
LIST	yes	yes	80,81, 77	line printer	(1), (2),(5)
NEW	yes	no	SYSIN	77	(1)
PLST	yes	yes	80,81, 77	line printer	(1), (2),(5)

Table 5.1. (continued)

task	subfile selection		input	output	
Keyword	explicite	implicite	file	file	remark
RCRD	no	no	77	80,81, 60,61	-
RENM	yes	no	80,81	-	(1)
REST	yes	yes	60	80,81	(6)
RNM7	yes	no	77	77	(1)
SFMT	yes	yes	60	74	(2),(6)
SUBF	yes	yes	80	-	(1)
SUBT	yes	yes	80	-	(1)
UNDL	yes	yes	80	80	(1)

Remarks:

(1) An error message is generated if no subfile selection is specified.
(2) All subfiles read from tape 60 will be copied to file 74 if ITEM R or ITEM C have been specified for implicite subfile selection.
(3) In case of explicite subfile selection, the input subfile is searched in temporary file 77 first. The search is continued in permanent files 80 and 81 if the subfile is not found in file 77. The search is restricted to the permanent files in case of implicite subfile selection.
(4) The output file is tape 61 if no subfile selection is specified while it is tape 62 in the opposite case.
(5) Warning: if the original versions of the corrected subfiles have not been renamed by task RNM7, not the corrected but the original versions will be printed.
(6) If no subfile selection is specified, all subfiles will be selected.

Table 5.2. The files composing PDF

number of the file	peripheral unit	record length (bytes)	max. number of records
74	temporary disk	80	1000
75	temporary disk	80	1000
76	temporary disk	80	5000
77	temporary disk	80	15000
80	permanent disk	36	3000
81	permanent disk	80	130000
60	magnetic tape	32 and 80	-
61	magnetic tape	32 and 80	-
62	magnetic tape	32 and 80	-

Table 5.3. Tasks for implicite subfile selection

Remark: The format of the implicite subfile selection mode is

ITEM A K1, K2

and eventually some additional data depending on the actual value of parameter A.

A	K1	K2	additional data
M	measurement type	-	none
N	the required name		none
D	starting date	final date	none
K	No. of core certificate	No. of core map	none
K	starting number	final number	none
R	number of questions	-	the question numbers
C	number of tests	-	data for the individual tests

6. Task EVAL: evaluating the data stored in PDF

There are two ways of specifying the input data for fitting: via the SYSIN input and by taking them from PDF. The former way is task RFIT which is discussed in Chapter 5 of Part 2 while the latter is the subject of the present chapter. The corresponding task keyword is EVAL. After the experimental data are read from PDF, the parameter estimation and the statistical analysis go on in the same way as in case of task RFIT.

If the physical contents of the subfiles are compared with those of the input data to be specified for task RFIT (see Chapter 1 of the present part and Chapter 5 of Part 2, respectively), it can be seen that the subfiles contain all the experimental facts but not those input data which depend on the user's consideration and which can vary from fitting to fitting even in relation tc the same set of raw experimental data. The following groups of the input data need to be given via SYSIN in case of task EVAL (references will be made to the input data groups defined in Chapter 5 of Part 2):

Group A: title of the problem (Group 1);
Group B: identification of the fitting function (Group 2) and the data defining the runs (Group 4);
Group C: output EDF file (Group 3) controlled by keywords BIBL or CLIB;
Group D: the data related to the iteration (controlled by keyword INIT) i.e. variables ITM, MALL, KEZD (Group 2) and the initial guess and fixed parameters (Group 12);
Group E: point drop limits (Group 18) controlled by keyword STEP.

One could say that the subfiles contain the objective part of the input data while those listed obave form their subjective part. In order to avoid confusion with the notations used in Chapter 5 of Part 2, these latter groups of the input data are denoted here by letters A to E (incl.).

All subfiles evaluated by task EVAL should be in temporary file 77. Either they may be created there (e.g. by using tasks NEW or CORR) or they may be copied to file 77. These operations will be called in the following "the preparation of the subfiles" the details of which are discussed in Section 5.1. On the basis of a

given set of prepared subfiles, an unlimited number of fitting problems may be formulated according to the rules formulated in Section 5.2.

6.1. Preparing the subfiles for evaluation

There are 4 task keywords which prepare subfiles for evaluation: NEW , CORR, RENM, and EVAL. The prepared subfiles are later identified by their preparation numbers. The total number of the subfiles which can be prepared by one task keyword is 20 as a rule. Task RENM is an exception since this number is 10 for it.

Preparation by NEW :

All those subfiles which are created in connection with one task keyword NEW are taken as prepared for a subsequent evaluation. Their total number can be 20 at maximum. The corresponding task keyword EVAL may be given either in the same line as NEW or separately. The preparation numbers are the serial numbers of creating the subfiles. The general structure of the input data is illustrated as follows (in case of preparing and evaluating 3 subfiles):

```
NEW
     :
   input data for creating subfile 1
     :
   input data for creating subfile 2
     :
   input data for creating subfile 3
     :
EVAL
     :
     :
   input data  for fitting
     :
     :
END
```

(In this and the following examples, any other task keyword - except EVAL - can stand instead of keyword END .) As said before, this is equivalent to the following input data:

```
NEW  EVAL
     :
   input data for creating subfile 1
     :
   input data for creating subfile 2
     :
   input data for creating subfile 3
     :
     :
     :
   input data for fitting
     :
     :
END
```

It is worthwhile to point out how the program is able to separate the input data given for creating subfile 3 from those belonging to the first fitting problem. As discussed in Section 3.1, the input data for creating a subfile start with the subfile's identifier and end with a line containing only ++++. Now, in the scheme presented above, the ++++ closing the data for creating subfile 3 are not followed by another subfile identifier indicating the beginning of the input data for fitting. It is noted finally that, according to the sense, only the explicite mode of the subfile selection is possible.

Preparation by CORR:

All those subfiles which are corrected in connection with one task Keyword CORR are taken as prepared for a subsequent evaluation. Their total number can be 20 at maximum. The corresponding task Keyword EVAL may be given either in the same line as CORR or separately. The preparation numbers are the serial numbers of correcting the subfiles. The general structure of the input data is illustrated as follows (in case of preparing and evaluating 3 subfiles):

```
CORR
     :
   input data for correcting subfile 1
     :
   input data for correcting subfile 2
     :
   input data for correcting subfile 3
     :
EVAL
     :
     :
   input data  for fitting
     :
     :
END
```

As said before, this is equivalent to the following input data:

```
CORR EVAL
    :
   input data for correcting subfile 1
    :
   input data for correcting subfile 2
    :
   input data for correcting subfile 3
    :
    :
   input data for fitting
    :
    :
END
```

It is worthwhile to point out how the program is able to separate the input data given for creating subfile 3 from those belonging to the first fitting problem. As discussed in Section 4.1, the input data for correcting a subfile start with the identifier of the corrected subfile and end with a line containing only ++++. Now, in the scheme presented above, the ++++ closing the data for correcting subfile 3 are not followed by another subfile identifier indicating the beginning of the input data for fitting.

Remarks:

- As a rule, the corrected subfiles are selected explicitely (as in case of NEW) but it is possible to use the implicite mode of the subfile selection, too. An example of this last possibility is reproduced here on the basis of Example 6 of Scetion 5.6.

```
CORR EVAL
ITEM  M  MI02
ITEM  D  100977  300678
ITEM  C  2
1  9  2  CU100.
1  9  1  3301
*9  3  1
9  3302  CU100.
++++
++++
    :
    :
   input data for fitting
    :
    :
END
```

- It is a rather frequent case that not all subfiles to be evaluated need to be corrected before evaluation but their original versions are taken. Keyword NONE has been introduced for handling such cases: it is given instead of the correction

data for those subfiles which need not be corrected. Let us assume that only subfile 2 needs to be corrected in the example given above. The overall structure of the input data is:

```
CORR EVAL.
identifier of subfile 1
NONE
identifier of subfile 2
     :
   input data for correcting text questions
++++
     :
   input data for correcting numeric questions
++++
identifier of subfile 3
NONE
     :
     :
   input data for fitting
     :
     :
END
```

The input data for correcting subfile 2 have been shown in a more detailed way than previously in order to show that the lines with ++++ are omitted for those subfiles for which keyword NONE is given instead of the data of correction.

- As usual with task CORR, the subfiles selected explicitely are searched first in temporary file 77 and the search is continued in permanent files 80 and 81 only if they are not found in file 77. Correspondingly, two cases are possible when keyword NONE is used: if the subfile is found in file 77, it is left in place while it is copied to file 77 if it has been found only in the permanent files.

Preparation by RENM:

As stated in Section 5.3.3, the renamed subfiles are copied to file 77 from permanent files 80 and 81. The renamed subfiles are taken as prepared for a subsequent evaluation. Their total number can be 10 at maximum. The corresponding task keyword EVAL may be given either in the same line as RENM or separately. The preparation numbers are the serial numbers of renaming the subfiles. The general structure of the input data is illustrated as follows (in *case of preparing and evaluating 3 subfiles*):

```
RENM
old identifier of subfile 1
new identifier of subfile 1
old identifier of subfile 2
new identifier of subfile 2
old identifier of subfile 3
new identifier of subfile 3
EVAL
     :
     :
   input data  for fitting
     :
     :
END
```

It is noted finally that, according to the sense, only the explicite mode of the subfile selection is allowed for task RENM.

Preparation by EVAL:

When task Keyword EVAL is given in a separate line, two cases are possible:

- It *no* data are given for subfile selection, this means that the input data for fitting will be formulated in terms of a previous subfile preparation defined by one of tasks NEW , CORR, and RENM used previously. Some of the examples shown above correspond to this case. If no subfiles are prepared, this generates an error message.
- If data are given for subfile selection in connection with Keyword EVAL, the eventual previous subfile preparation is cancelled and the selected subfiles will be taken as prepared for evaluation. This situation is somewhat different for the two subfile selection modes:

  * When it is explicite, the selected subfiles are searched in file 77 first. If they are found there, they are left in place. If they are not found, they are copied to file 77 from permanent files 80 and 81. The preparation numbers of the subfiles are their serial numbers in the subfile selection list of their identifiers.
  * When it is implicite, the selected subfiles are searched only in permanent files 80 and 81 and they are copied to file 77. The preparation numbers of the subfiles are the serial numbers of their copying to file 77. (The latter will correspond to the order of their storage in the permanent disk files.)

The total number of the prepared subfiles may not exceed 20.

Remarks on subfile preparation:

- If the preparation of some of the subfiles is not successful for some reason (e.g. they could not be found, they could not be created or corrected due to input errors), no fitting problem will be solved: all data specified for task EVAL will be read but they will remain uninterpreted.
- The prepared subfiles take up room in file 77. In connection with this, the limitations formulated in Section 5.1 for file 77 apply (i.e. 50 subfiles and a total of 15000 records at maximum).
- The subfiles which are copied to file 77 only for the sake of preparation will not be recorded to the permanent files by tasks RCRD or DREC (cf. Section 5.4.2). Similarly, they will be ignored for other tasks, too, operating on file 77 (e.g. CORR, RNM7, LIST).
- The preparation of a given set of subfiles remains valid either until the next subfile preparation or the use of anyone of tasks RCRD, DREC, and DEL7. It follows from this that the use of task keyword EVAL without subfile selection does not cancel the preceding subfile preparation.
- When task keyword EVAL is given in the same line as other task keywords, it should be related to the subfiles prepared by the tasks preceding it.

6.2. Input data for fitting

Following a task keyword EVAL, an unlimited number of fitting problems may be formulated. As in case of task RFIT, the corresponding input data are divided into groups which, to some extent, remind some of the groups belonging to task RFIT (see Chapter 5 of Part 2) but the amount of the input data is much less for task EVAL since the bulk of the data are taken from the subfiles. In order to avoid confusion with task RFIT, the input data groups belonging to task EVAL are identified by capital letters (and not by numbers as in case of task RFIT). Those groups which should be specified for every fitting problem (i.e. Groups A and B) will be marked by an asterisk (*). The input data belonging to the other groups (i.e. Groups C, D, and E) are optional. The latter are introduced by special keywords and the order of their specification is arbitrary. In relation to a fitting problem, each of Groups C, D, and E may be given only once. The default values of the variables to be specified in these latter groups are:

ITM = 2, MALL = 0, KEZD = 1,

i.e. no parameters are fixed and the initial guess of the fitted parameters will be estimated by the program; furthermore,

NSTEP = 1, MST = 0,

i.e. there will be only one point drop step in which all points will be considered in the fitting. In the explanations given below,

frequent references need to be made to Part 2 of this user's manual. Furthermore, the interpretation of part of the input data is the same as in case of task RFIT. In order to minimize cross references with Chapter 5 of Part 2, the text is the same here as there in relation to such variables. In relation to some other variables, the interpretation is slightly different which are not always pointed out (also in order to simplify the text).

* Group A. Title of the problem:

Any text of 80 characters in length (at maximum) which will be printed at the top of every output page. Warning: the first 4 characters of the text must be different from the task keywords (cf. Table 2.1 of Part 1) since the opposite case would switch the program to the inadvertently specified task. The safest approach is to leave the first position of the title line blank.

* Group B. Identification of the fitting function and data defining the runs:

The contents of this group of input data depend on the type of the subfiles involved and on the number of the fitting function. Typically, one run is composed of data belonging to one material of a subfile. There are exceptions to this general rule which will become clear from the following. The general format of Group B is:

MM, J, ard further data (see below)

where

MM is a variable identifying the fitting function. As discussed in Section 1.1.1 of Part 2, it is a combination of two or three variables:

$$|MM| = MODE + 100*g + 10000*KUKAC \qquad (6.1)$$

or

$$|MM| = MODE + 100*KUKAC \qquad (6.2)$$

depending on the actual value of MODE. The sign of MM bears additional information:

for MM < 0, the program computes the bias according to Section 4.4 of Part 2 while,
for MM > 0, the bias is not computed.

The value of MODE should be one of the numbers given in Table 1.1 of Part 2. Variable KUKAC may have three values (0, 1, and 2) the significance of which is explained in Section 1.2.11 of Part 2. It is noted that m i.e. the total number of the fitted

parameters (which is one of the most important variables of a fitting problem) is not an input data but is computed by the program according to Table 1.1 of Part 2.

J is the total number of the runs limited by

$$0 < J < 21.$$

The continuation of the data in this group depends on the actual value of MODE and on whether the fitted data (i.e. y) are to be taken from Question 79 or from Question 80. The possible cases will be differentiated from one another as Groups B1, B2, etc. from among which always one and only one applies in a fitting problem. Two remarks are made before going over to their discussion:

- the data belonging to this group may be broken into lines without any restriction;
- the value of J must be given also for such fitting functions which do not involve runs according to Table 1.1 of Part 2; of course, J = 1 is given in such cases (however, J > 1 does not generate an error message directly but might cause some run time error later);
- only MODE = 30 and MODE = 38 are allowed for MI02 type subfiles;
- MODE = 32 and MODE = 40 are not accepted for in connection with task EVAL. (The character of the corresponding input data is such that they do not fit well enough to be stored in the subfiles.)

* Group B1. Group B in the general case are:

MM, J, [K(j) , m(j) , for j = 1, 2, ..., J]

where

K(j) is the preparation number of the subfile from which data are taken for run j;

m(j) is the number of the material within subfile K(j) which is to compose run j.

* Group B2. Group B for MODE = 30:

As explained in Section 1.2.10 of Part 2, the runs are composed in the following way:

- runs 1 and 2, 3 and 4, 5 and 6 etc. correspond to subfiles 1, 2, 3, etc., respectively,
- runs 1, 3, 5, etc, for one hand, and runs 2, 4, 6, etc., for the other hand, belong to the same physical materials and positions.

It follows from this that the total number J of the runs should be even. An odd value of J is a data error. Taking all this into account, the format of Group B2 is:

MM, J, P1, P2, [K(j/2), m(j-1), P1TH(j-1),

m(j), P2TH(j), for j = 2, 4, ..., J]

where

P1 and P2 are the position numbers to which the spectrum index to be estimated is related;

K(j/2) is the preparation number of the subfile from which data are taken for runs (j-1) and j;

m(j) is the material number within subfile K(j/2) which will compose run j;

P1TH(j) is the position number of the thermal column data for run j corresponding to reactor position P1;

P2TH(j) is the position number of the thermal column data for run j corresponding to reactor position P2.

Remarks:

- Position numbers P1 and P2, respectively, should be the same in all subfiles (given either in column 4 at Question 66 or in column 9 at Question 79) while the material numbers assigned to the same physical material need not be the same in every subfile (e.g., it may happen that plutonium is material 3 in the first subfile while it is material 1 in the other ones).
- Position numbers P1TH and P2TH used for the thermal column may vary from run to run.
- Material numbers m(1), m(3), m(5), etc., for one hand, and m(2), m(4), m(6), etc., for the other hand, should designate the same physical materials.
- As a result of the fitting, a spectral index will be obtained if P1 = P2 but material numbers m(j-1) and m(j) correspond to different physical materials while the result of the parameter estimation will be a disadvantage factor if position numbers P1 and P2 are different but all material numbers m(j) designate the same physical material.

* Group B3. Group B for MODE = 38:

As explained in Section 1.2.1 of Part 2, every run consists of data corresponding to a material within a subfile. The spectral ratios will be estimated for positions (P(l), for l = 1, 2, ..., LP) where LP is the total number of the positions. The format of Group B3 is:

MM, J, LP, [P(ℓ), for ℓ = 1, 2, ..., LP],

[K(j), m(j), P1TH(j), P2TH(j), for j = 1, 2, ..., J]

where

LP is the total number of the physical positions;

P(ℓ) is the ℓth position number;

K(j) is the preparation number of the subfile from which data are taken for run j;

m(j) is the material number within subfile K(j) which will compose run j;

P1TH(j) is the position number of the thermal column data for run j corresponding to round foils;

P2TH(j) is the position number of the thermal column data for run j corresponding to hexagonal foils.

Remarks:

- As in Group B2, position numbers P1TH and P2TH of the thermal column data may vary from run to run but position numbers P(1), P(2), ... should be the same in all subfiles (given either in column 4 at Question 66 or in column 9 at Question 79).
- The value of LP may not exceed 20. In addition to this, the total number of the estimated parameters a may not exceed 20 neither. However, it can be checked only after all runs have been composed whether this restriction is observed.
- Position numbers P(1), P(2), ... need not occur in every subfile. Even the extreme case may happen that some of them do not occur in any of the subfiles without generating an error message. If, however, a position number occurs in several subfiles, it should designate always the same physical position.
- The position numbers given for the thermal column data should be in accordance with the general convention: P1TH and P2TH should be odd and even integers, respectively.

* Group B4. Evaluation of spectra stored at Question 80:

When not the data matrix "79" but the spectra stored at Question 80 are evaluated, all runs should correspond to the same subfile. The program differentiates this case from the previous one (i.e. from B1, B2, and B3) by that the preparation number K of this common subfile is given with a negative sign. Correspondingly, the input data for Group B4 are:

MM, J, -K, [m(j), S(j), for j = 1, 2, ..., J]

where

K is the preparation number of the common subfile from which data are taken for all runs;

m(j) is the number of the material within subfile K which is to compose run j;

S(j) is the number of the spectrum within subfile K and material m(j) which is to compose run j.

It is noted that it is an error if one of the spectra S(j) is not found in subfile K for material m(j).

Remarks:

- This option is allowed only for subfile types

  DRDT, DRDH, MEAS, R281, R282, R283

  (see Table 2.1);

- There are several restrictions as to which options may be specified at Questions 68 and 70. Section 6.3.9 is devoted to their discussion.

Group C. Output EDF file (Keywords BIBL or CLIB):

If one of the following two lines is given, a new data set will be included into EDF (see Chapter 7):

BIBL, data set identifier

or

CLIB, data set identifier

The logical number of the sequential file containing EDF is 64 or 62 depending on whether the keyword is BIBL or CLIB, respectively. (The job control DD statements should be formulated in accordance with this.) The recorded data set will be composed as follows (see also Group 3 for task RFIT in Chapter 5 of Part 2):

- it gets the specified identifier;
- its title will be that given at Group A;
- the recorded field is:

  * for fitting functions NNo. 20, 21, 34, and 43:

x_I, Ψ_I, $D\Psi_I$

for all positions I (cf. Sections 1.2.7, 1.2.8, and 1.2.12 of Part 2),

* for fitting functions NNo. 47, 48, and 49:

x_I, y_I, z_I

for all positions I (cf. Section 1.2.14 of Part 2),

* for fitting function No. 38:

foil identifier, A_{wl}, error

for all parameters having this physical meaning (cf. Section 1.2.10 of Part 2).

This group of the input data has no effect for the fitting functions not mentioned. In case of functions No. 38, the foil identifiers are determined in the following way:

- the first 2 characters will be identical with those of material names given at Question 9;
- the third character will be the serial number of the corresponding position number P(ℓ);
- the fourth character will be T or M for P(ℓ) odd or even, respectively.

It is noted that the program does not check whether the specified data set identifier already occurs in EDF or not. If Group C is not given, no new data set will be recorded to EDF. The total number of records in EDF may not exceed 15000. Would the new data set go beyond this limit, an error message is generated.

Group D. Data related to the iteration (Keyword INIT):

The data related to the iteration are introduced by Keyword INIT which is followed by the values of variables ITM, MALL, and KEZD where

ITM determines the iteration mode. According to Section 3.1 of Part 2, its value may range from 1 to 4, incl. (Its usual value is ITM = 2.)

MALL is a variable the absolute value of which gives the total number of fixed or dependent parameters (cf. Section 1.1.3 of Part 2). The sign of MALL bears additional information:

for MALL < 0, some parameters will depend on other ones while,
for MALL > 0, some parameters will be fixed.

Further data concerning the fixed or dependent parameters are given at Subgroups D2 to D5, incl. Warning: |MALL| should be less than the total

number of the fitted parameters (i.e. m). The latter is the result of a rather complicated computation in case of fitting functions No. 38. Therefore, it is advisable to let the program choose the fixed parameter(s) if necessary and give MALL = 0 here.

KEZD ıdicates how the initial guess of the fitted .arameters will be determined: for KEZD = 0, they will be given as input data while they will be computed by the program if KEZD > 0 (cf. Section 3.4 of Part 2). Further details are given at Subgroups D2 to D5, incl.

If Keyword INIT is not given, the default values are taken:

ITM = 2, MALL = 0, KEZD = 1.

If another value is desirable for at least one of these variables, Keyword INIT needs to be given followed by the actual values of these variables. The continuation of the input data depends on their actual values (as in case of task RFIT, see Chapter 5 of Part 2). Therefore, Group D is subdivided into subgroups:

Subgroup D1: ITM, MALL, KEZD

given in the line following Keyword INIT. This subgroup is the only one which should always be given following Keyword INIT.

The other subgroups need to be given only if

ITM > 2

and/or

MALL > 0 or MALL < 0

and/or

KEZD = 0.

The cases are the following (the subgroups defined below *must be given in separate lines*):

KEZD = 0

(i.e. the initial guess ot the estimated parameters are input data):

MALL = 0: Subgroup D2: the initial guess is given for all parameters a_K.
Subgroup D3: the values of s_K for all paramaters a_K if ITM = 3 or 4.

MALL > 0: Subgroup D2: subscripts K of the fixed parameters a_K;
Subgroup D3: the initial guess for all parameters a_K taking into account that these values will not change for the fixed parameters.
Subgroup D4: the values of s_K for all variable parameters a_K if ITM = 3 or 4.

MALL < 0: Subgroup D2: subscripts K for the dependent parameters a_K;
Subgroup D3: the initial guess for all parameters a_K taking into account that these values will be taken as γ_K^* for the dependent parameters (cf. Equ. (1.1a) of Part 2);
Subgroup D4: (K', β_K) for the dependent parameters (cf. Equ. (1.1a) of Part 2).
Subgroup D5: the values of s_K for all independent parameters a_K if ITM = 3 or 4.

KEZD > 0

(i.e. the initial guess of the estimated parameters are calculated by the program)

MALL = 0: Subgroup D2: the values of s_K for all parameters a_K if ITM = 3 or 4.

MALL > 0: Subgroup D2: (K, a_K) for the fixed parameters;
Subgroup D3: the values of s_K for all variable parameters a_K if ITM = 3 or 4.

MALL < 0: Subgroup D2: (K, γ_K^*) for the dependent parameters (cf. Equ. (1.1a) of Part 2);
Subgroup D3: (K', β_K) for the dependent parameters (cf. Equ. (1.1a) of Part 2).
Subgroup D4: the values of s_K for all independent parameters a_K if ITM = 3 or 4.

Group E. Point drop limits (Keyword STEP):

As exposed in Section 5.1.1 of Part 2, the point drop limits are defined in program RFIT by intervals of subscript i (see Equs. (2.8) of Part 4). This is the basic option corresponding to MST = 0. In some cases, however, it is more straightforward to define the point drop limits on the basis of the x variable. The possible values of MST and the corresponding options are the following. Only such points will be taken into account in an individual point drop step for which:

- subscript i belongs to a given interval (MST = 0);
- the value of the x variable belongs to a given interval (MST = 1 or 2);
- the value of the x variable is different from given values (MST = 3 or 4);

- the value of the x variable belongs to a given interval and it is different from given values (MST = 5 or 6); this option is a combination of the previous two options;
- the form of the above mentioned conditions depend on the runs (MST = 7);
- in case of task EVAL, the value of MST can be 81: this means that the point drop data are to be taken from Question 81 within the specified subfiles.

It follows from this that the odd values of MST (i.e. 1, 3, and 5) mean that the data defining the point drop limits are the same for all runs while the even values (i.e. 0, 2, 4, and 6) mean that they depend on the runs.

The data related to the point drop limits are introduced by keyword STEP followed by the value of variable NST (in the same line as keyword STEP) i.e. the first line of Group E is:

STEP, NST, NN, $[j(\ell)$, for $\ell = 1, 2, \ldots, NN]$

Here,

NST is a combined variable:

$$NST = NSTEP + 100*MST \qquad (6.3)$$

where NSTEP is the number point drop steps denoted by L in Section 5.1.1 of Part 2 and MST indicates in which way the point drop limits will be defined in the following data lines. Limitations:

$0 < NSTEP < 21$ and $NSTEP*J < 201$,

furthermore,

$-1 < MST < 8$ or $MST = 81$.

If the input value of NSTEP does not comply with these restrictions and the condition

$$NSTEP < \min(20, 200/J), \qquad (6.4)$$

an error message is generated.

NN is the total numbers of the runs for which point drop data will be given in the following lines; NN is read only for MST = 7; it may be omitted even when MST = 7: this means that point drop data will follow for every run;

$j(\ell)$ are the numbers of the runs for which point drop data will be given in the following lines; they are read only if NN > 0; it is an error if a repetition occurs among them.

Two cases need to be considered separately: firstly, when Group E is omitted (i.e. the default option is taken), and secondly, when Group E is given. The significance of the different values of variable MST is discussed in connection with the second case.

Default option: Group E is not given

When Group E is omitted, this is equivalent to MST = 81. This case should be further subdivided into cases:

(1) When no subfile contains data at Question 81, the default values are the following:

NSTEP = 1,

MST(j) = 0 for j = 1, 2, ..., J,

and the point drop limits are chosen so that no points were dropped in the fitting.

(2) When some (or all) of the subfiles contain data at Question 81, they will be used for the corresponding runs while the default values defined above will be taken for those runs for which no data are found at Question 81. Let us denote the value of NSTEP specified for run j by NSTEP(j). Would they be different for j = 1, 2, ..., J, the final value of NSTEP will be their maximum. If NSTEP(j) < NSTEP for run j, the point drop data specified for step NSTEP(j) will be used for steps NSTEP(j)+1, ..., NSTEP.

The case when Group E is given

When Group E is given, the following input data depend on the actual value of MST. The input data are the following for its different values:

When MST = 81, no further data are needed in Group E. This is almost the same as the default option discussed above. The difference consists in that the value of NSTEP is now given and it will overwrite those data which are stored for it in the subfiles. The treatment is the following:

- if more steps are specified in the subfiles, their number is truncated to this input value of NSTEP;
- if less steps are specified in the subfiles, the steps are complemented up to NSTEP by taking for the missing steps the point drop data given for the last specified step.

When MST=0, 2*NSTEP integer numbers

$$(i_{1\ell},\ i_{2\ell},\ \ell = 1, 2, \ldots, \text{NSTEP})$$

are given which will be the subscript limits appearing in Equs. (2.8) of Part 4. These numbers are given for every run but the subscript limits are related to the subscript values within each run separately.

When MST = 1, 2*NSTEP real numbers

$$(x^{\ell}_{min},\ x^{\ell}_{max},\ \ell = 1, 2, \ldots, \text{NSTEP})$$

are given which will be the lower and upper limits of the x variable pertaining to the individual point drop steps. As these intervals are valid for every run, they are given only for j = 1.

When MST = 2, 2*NSTEP real numbers

$$(x^{\ell}_{min},\ x^{\ell}_{max},\ \ell = 1, 2, \ldots, \text{NSTEP})$$

are given which will be the lower and upper limits of the x variable pertaining to the individual point drop steps. As these intervals vary with the runs, they are given for every run.

When MST = 3, the following data are given:

$$(n_{\ell},\ x^{\ell}_{1},\ x^{\ell}_{2},\ \ldots,\ x^{\ell}_{n_{\ell}},\ \ell = 1, 2, \ldots, \text{NSTEP})$$

where n_{ℓ} is the number of the x values to be left out in step ℓ followed by the list of these particular x values. The value of n_{ℓ} may be 0. In this case, the list of the x values is omitted. As these data are valid for every run, they are given only for j = 1.

When MST = 4, the following data are given:

$$(n_{\ell},\ x^{\ell}_{1},\ x^{\ell}_{2},\ \ldots,\ x^{\ell}_{n_{\ell}},\ \ell = 1, 2, \ldots, \text{NSTEP})$$

where n_{ℓ} is the number of the x values to be left out in step ℓ followed by the list of these particular x values. The value of n_{ℓ} may be 0. In this case, the list of the x values is omitted. As these data vary with the runs, they are given for every run.

When MST = 5, the combination of the data pertaining to options 1 and 3 the data are given:

$$(x^{\ell}_{min}, x^{\ell}_{max}, n_{\ell}, x^{\ell}_{1}, x^{\ell}_{2}, \ldots, x^{\ell}_{n_{\ell}},$$
$$\ell = 1, 2, \ldots, NSTEP)$$

where the meaning of the variables has been explained previously. As these data are valid for every run, they are given only for j = 1.

When MST = 6, the combination of the data pertaining to options 1 and 3 the data are given:

$$(x^{\ell}_{min}, x^{\ell}_{max}, n_{\ell}, x^{\ell}_{1}, x^{\ell}_{2}, \ldots, x^{\ell}_{n_{\ell}},$$
$$\ell = 1, 2, \ldots, NSTEP)$$

where the meaning of the variables has been explained previously. As these data vary with the runs, they are given for every run.

When MST = 7, the option variable MST(j) used for specifying the point drop limits varies with the runs. Therefore, the data belonging to each run start with MST(j) and continue with the data to be given as functions of the value of MST(j) according to the rules formulated above for MST(j) = 0 to 6. When MST(j) = 81 is specified, the point drop data will be taken from the subfile corresponding to run j. If the value of NSTEP is different from that stored in the subfile, the treatment is the same as explained above for MST = 81.

Remarks:

- When MST > 0, the total number of the point drop data is limited to 2000.
- When MST > 0, the point drop data should be formulated in terms of the x values determined on the basis of the subfiles (cf. Section 6.3.1) since the final $i_{1\ell}$ and $i_{2\ell}$ subscript limits are computed before the x values are eventually transformed (as e.g. in case of MODE = 20, 21, 30, 34, etc.).
- If MST = 81, it will be ultimately changed to MST = 7. Furthermore,
  * if MST(j) = 81 for some run j, this value will ultimately be changed to the value read from the corresponding subfile;
  * MST(j) becomes equal to 0 if no data are stored at Question 81 in the corresponding subfile.
- When MST = 0, the input values of subscript limits $i_{1\ell}$ and $i_{2\ell}$ should correspond to the set of the x data resulting after the determination of the decay

correction factors (for KORD > 3) and the background corrections (for KORA > 3): the data belonging to the monitors will be left out, and the points will be eventually reordered in ascending order of the x values. (For more details, see Chapter 2 of Part 2 and Questions 68 and 70 in Section 6.3.) When MST > 0, the subscript limits will be computed after the determination of the corrections.

- When the initial guess of the fitted parameters is estimated by the program (i.e. KEZD > 0 in Group D), the subscript limits $i_{1\ell}$ and $i_{2\ell}$ given for the first and last point drop steps can play some role in case of some fitting functions (see Section 3.4 of Part 2).
- The program verifies whether the subscript limits are specified correctly only when it proceeds to the point drop analysis (see Chapter 2 of Part 4).

6.3. Interpretation of the subfiles

The previous two sections deal with the rules of specifying the input data for task EVAL. As a conclusion of the present chapter, it will be discussed how the subfiles are interpreted i.e. how their contents are converted into the data matrix defined as Group 17 in case of task RFIT (see Chapter 5 of Part 2). The data handling is different for data groups B1 to B4 (incl.). Due to the large number of simplifications in relation to Groups B1 to B3 (incl.), the case of Group B4 will be discussed separately.

The discussions presented below apply to each run individually. As defined at Groups B, the data matrix belonging to run j is composed of the data stored for material m(j) in the subfile with preparation number K(j). The total number of the matrix elements may not exceed 5000 for a run. According to the practice of the ZR-6 measurements, this is not a severe restriction since the total number of the data belonging to a run is usually far less than 5000 (or, at least, the data may be split into several runs so that this restriction could be observed).

Once the data matrices are built up for all runs and the option variables determining their structure obtain some value, there is no more difference between tasks RFIT and EVAL: the computations go on along the same lines. In this respect, some general remarks might be useful concerning the additive and decay corrections. In case of task RFIT, both corrections (and their variances) can be determined in two ways: firstly, they can explicitely given as input data; secondly, they can be computed by the program using the input value of some option variables and parameters. In case of task EVAL, the corrections (and their variances) can be determined in three ways: firstly, they may be directly given in the subfiles; secondly, they may be computed while the subfile is interpreted; thirdly, the contents of the subfiles may be converted into some option variables and parameters with the help of which the corrections are computed as in case of task RFIT. In the present section (see e.g. Sections 6.3.5 and

6.3.7), the following expression is frequently used: "the correction is directly computed". It is clear that the second of the above mentiond ways of interpreting the subfiles is meant by this.

6.3.1. The x variable

It is the x variable which requires the most sophisticated discussions. From the point of view of its determination, four basic cases need to be distinguished from one another on the basis of the subfile type. A further distinction applies to type MEAS subfiles according to the the number of the fitting function. These cases are summarized in Table 6.1.

Table 6.1. Cases for the determination of the x variable

type of the measurement	type of the subfile	No. of the fitting function (MODE)
calibration	CA01	usually: MODE = 20
axial and azimuthal	MA02, MA04, MA05, MA11, MI06, MEAS	for MEAS: MODE = any number which is not radial
radial	MA01, MA03, MA10, MA13, MEAS	for MEAS: MODE = 9, 10, 14, 20, 21, 24, 25, 29, 34, 37, 39, 43
spectral and microflux	MI01, MI02	only MODE = 30 or 38 is accepted for MI02

The way of determining the x variable will be different for these four cases. Independently of which one of them applies, the core map number is always taken from the subfile if it is given as the second number at Question 5. It will be used for the determination of the lattice pitch in case of the radial measurements. However, this number should be identical for all such runs for which it is given in case of all types of measurements. (In case of radial measurements, the program requires more, cf. Section 6.3.1.3.)

We use here the notations of Chapter 5 of Part 2: LX(j) is the option variable for the determination of the x variable for run j (see at Group 10). MX is the option variable specified at Question 65.

6.3.1.1. Foil identifiers

The foil identifiers are normally taken from column 2 of the matrix "79" given for material m(j). If column 2 is missing, the program takes the foil identifiers from column 2 of the corresponding matrix "66". In this case, the foil identifiers will be determined with the help of the conditional identifiers to be given in columns 1 of both matrices "79" and "66". When column 2 is missing from the matrix "79", typical errors are the folloing:

- there is no matrix "66" for material m(j);
- column 2 is missing also from matrix "66" for material m(j);
- a conditional identifier mentioned in the matrix "79" is not found in the corresponding matrix "66";
- column 1 is missing in any of the matrices "66" and "79" unless the following special case occurs: the number of rows in the matrix "66" is equal to N_K given at Question 63 for material m(j); in this case, the values given in column 2 of matrix "66" are copied M_K times into column 2 of the corresponding matrix "79". (M_K is also given at Question 63 for material m(j).)

The foil identifiers will become the values of the x variable only for CA01 type subfiles i.e. for foil calibration measurements. In this case, the fitting function is normally MODE = 20 although any other fitting function is accepted as well. In addition to this case, the foil identifiers are needed also for the determination of:

- the foil calibration factors (if they are not given in column 12 of the matrix "79" for material m(j));
- the remanent activity (see Question 75);
- the decay and/or background corrections (when KORD = 4, KORA = 4 or 6, see Questions 70 and 68, respectively);
- the macroflux correction (see Question 82).

When discussing these cases in other sections, reference is made to the present section since the determination of the foil identifiers is the same in all these cases.

6.3.1.2. Axial and azimuthal measurements

A case is treated as an axial distribution measurement if

- either the subfile type is one of MA02, MA04, MA05, MA11 (cf. Table 2.1),
- or the subfile type is MEAS and the number of the fitting function is not radial (see Table 6.1).

MI06 type subfiles are treated as azimuthal distribution measurements (cf. Table 2.1). From the point of view of the data handling, there is no formal difference between the treatment of the axial and the azimuthal distributions.

The x variable can be specified in three different ways:

- If column 8 exists in the matrix "79" for material m(j), the values of the x variable are taken from there. Control variable LX(j) becomes equal to 2.
- If there is no column 8 in the matrix "79" for material m(j), the program turns to column 3 of the matrix "66" given for material m(j). If this column is given, the values of the x variable will be taken from there based on the correspondence between
  * the foil identifiers stored in the columns 2 if column 2 is given in both matrices "66" and "79" for material m(j); if a foil identifier mentioned in the matrix "79" is not found in the respective matrix "66", this generates an error message unless the missing identifier belongs to the background or the decay monitor (cf. Sections 6.3.5 and 6.3.7, respectively);
  * the conditional identifiers if column 2 is missing from one of the matrices "66" and "79" but column 1 is given in both matrices for material m(j); if a foil identifier mentioned in the matrix "79" is not found in the corresponding matrix "66", it is a data error;
  * when column 1 is also missing in one of the matrices "66" and "79", the values given in column 3 are copied M_K times into the data matrix if the numer of the rows of the matrix "66" is equal to N_K for material m(j); both N_K and M_K are given at Question 63.

  Control variable LX(j) becomes equal to 2.
- If the x variable cannot be determined from the matrix "66" for any reason (there is no matrix "66" for material m(j), column 3 is missing, etc.), the program turns to Question 65. If there are data for material m(j) at Question 65,
  * LX(j) = 1 if M_K = 1 for material m(j), and the values of x_i and δx_i are taken from the subfile; note that this remains so only if the foil identifiers are not needed (cf. Section 6.3.1.1); the values of the x variables are computed according to Equ. (2.4), and LX(j) is changed to 2 in the opposite case;
  * LX(j) = 2 if M_K > 1 for material m(j) and the values of the x variable are computed according to the data given in the subfile (cf. Equs. (2.3) and (2.4)).

  It is a data error if there are no data for material m(j) at Question 65.

6.3.1.3. Radial measurements

A case is treated as a radial distribution measurement if

- either the subfile type is one of MA01, MA03, MA10, MA13 (cf. Table 2.1),
- or the subfile type is MEAS and the number of the fitting function (MODE) is one of 9, 10, 14, 20, 21, 24, 25, 29, 34, 37, 39, and 43 (see Table 6.1).

Control variable LX(j) becomes equal to 2 in all radial cases since the values of the x variable will always be determined explicitely. There is a further distinction according to whether hexagonal

coordinates (h1,h2) need to be known or it is sufficient to know only the radial coordinate r. The former case will be called hexagonal which is identified by one of the following values of MODE: 20, 21, 34, 39, 43, 47, 48, and 49. All other radial cases will be called non-hexagonal. This distinction between hexagonal and non-hexagonal cases applies to all subfile types.

For non-hexagonal cases, the x variable is the radial r coordinate which is computed in the following way. First, the hexagonal coordinates (h1,h2) are transformed to the core centre:

$$h1' = h1 - h10, \tag{6.7a}$$

$$h2' = h2 - h20. \tag{6.7b}$$

(h10,h20) are the hexagonal coordinates of the core centre. Then the radial coordinate is computed as follows:

$$r = \begin{cases} ph1' & \text{if} \quad h1' = h2' \\ p\sqrt{h1'^2 + h2'^2 - h1'*h2'} & \text{if} \quad h1' < h2' \\ -p\sqrt{h1'^2 + h2'^2 - h1'*h2'} & \text{if} \quad h1' > h2' \end{cases} \tag{6.8}$$

where p is the lattice pitch. Geometrically, r is the distance between lattice points with hexagonal coordinates (h1,h2) and (h10,h20) (cf. Equs. (1.20), (1.21), and (A.12) of Part 2).

For hexagonal cases, the x coordinate is a simple combination of the hexagonal coordinates:

$$x = h1 + h2/100. \tag{6.9}$$

This representation allows an unambiguous decomposition of x into h1 and h2 since the latter are always positive and less than 70. As a matter of fact, coordintes less than 5 and greater than 65 did not occur in the practice of the ZR-6 experiments. (This circumstance will be taken into account below.) As discussed in Chapter 1 of Part 2, the values of this x variable will be further transformed (with the exception of MODE = 20). These transformations make the knowledge of the lattice pitch and the core centre coordinates necessary. For MODE = 43, the coordinates (h11,h21) of another lattice point needs also to be known.

It follows from this that the knowledge of parameters p, (h10,h20) is necessary in both hexagonal and non-hexagonal cases. There is, however, an essential difference: these parameters are directly used for the computation of the x variable in the non-hexagonal cases (cf. Equs. (6.7) and (6.8)) while these parameters do not directly intervene in Equ. (6.9) applied in the hexagonal cases since they will be used only during the fitting process as described in Part 2. As a consequence of this, the program has to make sure that the same parameter values are used for all runs in the non-hexagonal cases while some parameter values

need to be specified for at least one of the runs in the hexagonal cases. In the following, the possible ways of determining these parameters and the eventual restrictions are discussed separately for hexagonal and non-hexagonal cases.

Before proceeding to this, some remarks are necessary in order to clarify the relation of the parameter values specified here to those needed for task PLOT (see Part 4). Some plotting options also require (or allow) the specification of parameters p and (h10,h20). It follows from this that, when task EVAL is used, some values might be available for these parameters as results of a previous application of task PLOT. The existence of the latter data, however, does not absolve the user from their specification for task EVAL. This obligation does not hold in the opposite direction: the values specified along with task EVAL overwrite those which eventually might have been given for task PLOT. Of course, the latter will be restored after the use of task EVAL will have been terminated.

Lattice pitch

The normal way of specifying the lattice pitch (i.e. p) is to give it directly at Question 67. The program considers p as given if the third number stored at Question 67 is different from 0. When p is not given directly, the program determines it on the basis of the core map number to be given as the second number at Question 5. This is done according to the Table 6.2.

Table 6.2. Lattice pitch for various core map numbers

p [cm]	core map No.
1.1	39, 40, 41, 188, 189, 190, 197, 198, 207
1.5	220, 221, 231, 232, 316, 318
1.905	325, 330
1.27	otherwise

The following remarks should be borne in mind concerning the lattice pitch:

- The program accepts that the core map number is not given for some or all of the runs but it should be the same for all such runs for which it is given. This remark applies independently of whether the core map number is used or not. (The reason of this is that the physical meanings of the fitting functions defined in Chapter 1 of Part 2 involve the assumption that all runs are related to the same reactor core. If it is desirable to get rid

of this restriction, Question 5 should be omitted or the core map number should be temporarily corrected to the same value for all runs.)
- When, for some run j, p is given at Question 67, this value is used independently of the core map number.
- When, for some run j, p is not given neither at Question 67 nor through the core map number, the program sets p = 1.27 for j = 1 while p is set equal to its previous value for j > 1.
- When, for some run j, p is given either at Question 67 or through the core map number, this should be the same for all such runs. This holds for all radial cases. For hexagonal cases, it is sufficient to give p in at least one of the runs. For non-hexagonal cases, an additional restriction holds: the value of p should be the same for all runs. For example, it is a data error if the default value is used for run 1 but something else is given for run 2. It follows from this that it is advisable to define the runs in such a way that p gets some value in the first run since this value can then be used in all subsequent runs even if p is not given for them.

Coordinates of the core centre

The coordinates of the core centre (i.e. h10 and h20) are specified at Question 67. The program considers them as directly given if Question 67 is answered (i.e the program accepts any value for them). In the opposite case, the program takes the default values in the first run (j = 1):

h10 = h20 = 35.

while it takes the values used in the previous runs for j > 1.

When, for some run j, (h10,h20) are given at Question 67, they should be the same for all such runs. This holds for all radial cases. For hexagonal cases, it is sufficient to give them in at least one of the runs. For non-hexagonal cases, an additional restriction holds: (h10,h20) should be same for all runs. For example, an error message is generated if the default values are used for run 1 but something else is given for run 2. It follows from this that it is advisable to define the runs in such a way that the core centre coordinates get some values in the first run since these values can then be used in all subsequent runs even if they are not given for them.

Coordinates of the point defining the core axis for MODE = 43

The core axis is defined for fitting function No. 43 (cf. Section 1.2.12 of Part 2) through the specification of another lattice point (h11,h21) of the core axis (which should be different from the core centre). These coordinates can be given as the 4th and 5th numbers stored at Question 67. The program considers them as given if Question 67 is answered and both the 3rd and 4th numbers (i.e. p and h11) are different from 0.

There are no default values for (h11,h21). It follows from this that these coordinates should be given at least for one of the runs. If they are given for several ones, they should be identical. An error message is generated if these coordinates are not given for MODE = 43.

Hexagonal coordinates

The hexagonal coordinates (h1,h2) are given either in columns 8 and 9 of the matrix "79", or in columns 3 and 4 of the matrix "66", respectively. The following cases are possible:

- If both columns are given in the matrix "79" for material m(j), the program calculates the values of the x variable as shown in Equs. (6.7), (6.8), and (6.9). In this case, the program does not check the coordinates whether they are realistic i.e. any numbers are accepted as h1 and h2.
- If only one of columns 8 and 9 is given in the matrix "79" for material m(j) (it is much the same which one), the program verifies whether the given data can be considered as being of hexagonal or radial form i.e. whether realistic values of h1 and h2 can be found with the help of which the given data can be calculated according to Equ. (6.9), or Equs. (6.7) and (6.8), respectively. (It will be explained in detail below how this is done.) The main lines of the procedure are the following:
  * if all of the given data are of hexagonal form, they give the values of the x variable for hexagonal cases while they are decomposed into h1 and h2 in non-hexagonal cases; then the x variable is computed according to Equs. (6.7) and (6.8);
  * if at least one of the data is not of hexagonal form, this is a data error in hexagonal cases; in non-hexagonal cases, they give the values of the x variable if all of them are of the radial form; the opposite case is a data error.
- If neither column 8 nor 9 is given at Question 79 for material m(j), the program looks for the hexagonal coordinates in columns 3 and 4 of the matrix "66". As regards the latter, the same verification procedure is applied as at Question 79. If its outcome is positive, the determination of the x coordinate is based on columns 2 or columns 1 if at least one of them exist in both matrices "66" and "79":
  * if columns 2 exist in both matrices "66" and "79", the determination of the hexagonal coordinates is based on them; if a foil identifier mentioned in the matrix "79" is not found in the respective matrix "66", this is a data error unless the missing identifier belongs to the background or the decay monitor (cf. Sections 6.3.5 and 6.3.7, respectively);
  * if column 2 is missing in one of the matrices but columns 1 exist in both matrices "66" and "79", the determination of the hexagonal coordinates is based on them; if a conditional identifier mentioned in the matrix "79" is not found in the corresponding matrix "66", this is a data error.
- In addition to the cases mentioned above, an error message is generated if:
  * there is no matrix "66" for material m(j);

* there is a matrix "66" for material m(j) but neither columns 1 nor columns 2 exist in both matrices "66" and "79";
* neither column 3 nor column 4 is given in the matrix "66" for material m(j).

The verification procedure mentioned above is applied when only one of columns 8 and 9, or 3 and 4 are given in the matrices "79" and "66", respectively. Before proceeding to its description, it is noted that such cases should be rather exceptional than normal. That is why the program does not look at Question 66 when the procedure leads to a negative conclusion at Question 79 ignoring that both columns 3 and 4 can exist in the matrix "66".

For discussing the verification procedure, we shall need two functions which are used also in other sections of this user's manual (e.g. Appendix 1 of Part 2). Let z be any non-negative, real number. Function [z] or entier(z) is the largest integer which is not greater than z:

$$[z] = \max_{K} \{K: K \leq z\}. \tag{6.10}$$

Function nint(z) (= nearest integer) is the rounded off value of z:

$$\mathrm{nint}(z) = [z + 0.499999...]. \tag{6.11}$$

We consider the verification of the hexagonal form first. Let z be one of the data which is supposed to be of the form given by Equ. (6.9). It is easy to see that

$$h1 = [z] \tag{6.12a}$$

and

$$h2 = 100(z - h1). \tag{6.12b}$$

Now, z is considered as being of hexagonal form if all of the following conditions are met:

$$4 < h1 < 66, \tag{6.13}$$

$$4 < h2 < 66, \tag{6.14}$$

and

$$|h2 - \mathrm{nint}(h2)| < 0.05. \tag{6.15}$$

Condition (6.15) expresses that h2 should be an integer number. It cannot be made more severe (i.e. the right hand side cannot be less than 0.05) since the storage precision of the computer is not better than 8 decimal places. Would z not fulfill any one of these conditions, the program concludes that z is not hexagonal.

If z is of radial form, it follows from Equs. (6.7) and (6.8) that

$$z' = (z/p)^2 \qquad (6.16)$$

is an integer number. This is verified by the condition

$$| z' - \mathrm{nint}(z') | < 0.0005. \qquad (6.17)$$

This is formulated according to the experience that experimenters who calculate radial coordinate r use rarely more than 4 decimal digits. Let us assume that condition (6.17) holds and let us introduce the notation

$$K = \mathrm{nint}(z'). \qquad (6.18)$$

z is of radial form if there are integers (h1,h2) satisfying conditions (6.13) and (6.14) for which

$$K = h1^2 + h2^2 - h1*h2. \qquad (6.19)$$

As a matter of fact, these are the (h1',h2') coordinates which are obtained through the transformation (6.7). For the sake of simplicity of the notations, the primes (') are omitted in the following derivation. It is sufficient to consider only the 30-degree symmetry sector for which the additional condition

$$0 < h1 < [h2/2] \qquad (6.20)$$

holds (cf. Section A.1.1 of Part 2). If h2 is fixed and h1 varies in the range defined by condition (6.20), the minimum and maximum values of K are (as it can be simply derived from Equ. (6.19)):

$$\min(K) = 3h2^2/4 \qquad \text{(occurring for } h1 = h2/2),$$

and

$$\max(K) = h2^2 \qquad \text{(occurring for } h1 = 0).$$

In our case, K is given and we are looking for some similar range within which h2 can vary. This can be obtained by inverting the latter formulae:

$$K1 = \min(h2) = [\sqrt{K}] \qquad (6.21a)$$

and

$$K2 = \max(h2) = [\sqrt{4K/3} + 1]. \qquad (6.21b)$$

It follows from this that we have to check for

$$h2 = K1, K1+1, K1+2, \ldots, K2,$$

whether there exists an integer value of h1 satisfying Equ. (6.19). Given K and h2, we solve Equ. (6.19) for h1 as the unknown. Only one of the two solutions satisfies condition (6.20):

$$h1 = \frac{h2 - \sqrt{h2^2 - 4(h2^2-K)}}{2}. \qquad (6.22)$$

The program computes this h1 and verifies whether one of [h1], [h1]-1, and [h1]+1 stisfies Equ. (6.19). If so, the conclusion is that z is of radial form.

This algorithm tells not only whether z is of radial form or not but it yields also coordinates (h1',h2') which, by using Equs. (6.7), could be transformed into (h1,h2). This is still not done since these are not necessarily the real hexagonal coordinates but only their transforms into the 30-degree symmetry sector (cf. Section A.1.1 of Part 2). That is why it is required that z should be of hexagonal form in the hexagonal cases and no attempt is made to restore the hexagonal coordinates from radial coordinate r.

6.3.1.4. Spectral measurements

A case is treated as a spectral index or (spectral ratio) measurement if the subfile type is MI02. MI01 type subfiles are treated as microflux distribution measurements. The treatment of these cases is different whence they are considered separately. There is a common feature: the program sets LX(j) = 2.

Let us consider the simpler case i.e. that of MI01 type subfiles first. As they are distribution measurements, their treatment is similar to that of axial or azimuthal distribution measurements (cf. Section 6.3.1.2). There are only two differences:

- the values of the x variable are taken either from column 9 of the matrix "79" or from column 4 of the matrix 66";
- the program does not take into account the information which might be available at Question 65.

The case of MI02 type subfiles is much more complicated. First of all, only fitting functions NNo. 30 or 38 are accepted. It is a data error if variable MODE has an other value. This is true also vice versa: the data stored in the subfiles will be correctly prepared for fitting functions NNo. 30 and 38 only if the subfile type is MI02 although no error message will be generated if it is attempted to use fitting functions NNo. 30 and 38 in connection with other subfile types.

The determination of the x variable is based on the position numbers. For MODE = 38, the values of the x variable are given by Equs. (1.38) and (1.39) of Part 2 which are applied as

$$x = 20*P(\ell) + m(j) + K(j)/100 \qquad (6.23)$$

for foils irradiated in the investigated reactor lattice while

$$x = \begin{cases} 20 & \text{for circular foils,} \\ 40 & \text{for hexagonal foils} \end{cases} \qquad (6.24)$$

if the respective foils have been irradiated in the thermal column. Position numbers P(ℓ) have to be given in Group B3. The preparation number of the subfile (i.e. K(j)) will be treated as the irradiation number j in terms of the formulation of Section 1.2.10 of Part 2. The role of ℓ in Equ. (1.38) of Part 2 is played here by P(ℓ). For MODE = 30, the values of the x variable are P1 for j odd and P2 for j even (cf. Group B2). In both cases, monitor identifiers might also appear among the values of the x variable depending on the actual background and decay correction options (cf. Sections 6.3.5 and 6.3.7, respectively).

The position numbers are supposed to be given either in column 9 of the matrix 79" or in column 4 of the matrix "66". The program turns to the latter only if there is no column 9 in the matrix "79" for material m(j). If column 4 is given in the matrix "66" for material m(j), the position numbers will be taken from there based on the correspondence between either the foil identifiers or the conditional identifiers. As the treatment of the eventual data errors is different for these columns, they are formulated separately:

- The program considers the foil identifiers stored in the columns 2 first if a column 2 is given in both matrices "66" and "79" for material m(j). If a foil identifier mentioned in the matrix "79" is not found in the corresponding matrix "66", this is a data error unless the missing identifier belongs to the background or the decay monitor (cf. Sections 6.3.5 and 6.3.7, respectively).
- If column 2 is missing from at least one of the matrices "66" and "79" but column 1 is given in both for material m(j), the program turns to the conditional foil identifiers to be given in columns 1 of these matrices. If a conditional foil identifier mentioned in the matrix "79" is not found in the corresponding matrix "66", this is a data error.
- When column 1 is also missing in one of the matrices "66" and "79", the values given in column 3 are copied M_K times into the data matrix if the numer of the rows of the matrix "66" is equal to N_K for material m(j); both N_K and M_K are given at Question B3.

In case of spectral measurements, the program builds up the data matrix first including all position numbers occurring in the subfile for material m(j). As a second step, all those data are left out which are irrelevant for run j (i.e. for which the position numoer is equal to neither P1 nor P2 for MODE = 30, and to none of the P(ℓ), ℓ = 1, 2, ..., LP for MODE = 38). This remark should be born in mind taking into account the restriction that the total number of the elements in the data matrix may not exceed 5000. This restriction applies to all positions occuring in the subfile for material m(j) and not only to the positions actually considered in the fitting problem at hand.

Typical input data errors are:

- the positions given for the thermal column i.e. P1TH(j) and P2TH(j) coincide with one of positions P(ℓ);

- the rounded value of one of the P(ℓ) is 0;
- the parity of P1TH(j) and P2TH(j) is the same;
- the x0 for either the additive or decay corrections coincides with one of the P(ℓ).

6.3.2. The t variable

The t variable can have different significance:

- It can be the second independent variable of the fitting function. In the described version of program RFIT, this is restricted to function No. 13 (cf. Table 1.1 of Part 2). This function is not meant to be used in connection with task EVAL. Therefore, this significance of variable t is left out of consideration in the present section.
- It is used for the computation of the decay and additive corrections. In case of task EVAL, this is the only significance of t whence its values are determined only if at least one of the mentioned corrections is applied.
- The t variable becomes an auxiliary variable in case of some fitting functions such as e.g. NNo. 30, 34, 39, 43, etc. (see Chapter 1 of Part 2). As the auxiliary values of the t variable are computed only after the determination of the mentioned corrections, task EVAL does not regard this use of t either.

It follows from this that the t variable needs to be specified for material m(j) in the subfile if control variables KORA and/or KORD are greater than 1 for material m(j) (cf. Table 2.4 of Part 2). As discussed in Section 1.2.10 of Part 2, the decay correction is special for fitting function No. 30. Therefore, the t variable should always be specified for MODE = 30.

We use here the notations of Chapter 5 of Part 2: LT(j) is the option variable for the determination of the t variable for run j (see at Group 10). MT is the option variable specified at Question 64.

There are three ways of specifying t:

(1) If column 3 is given in the matrix "73" for material m(j), the values of t are taken from there. Control variable LT(j) becomes equal to 2. The initial time data eventually given at Question 22 are left our of consideration.

(2) If column 3 is missing but at least one of columns 5, 6, and 7 is given in the matrix "73" for material m(j), the values of t are determined as follows.

* Let us denote the elements of the matrix "73" in columns 5, 6, 7 by t5, t6, and t7, respectively. All time data will be expressed in the smallest unit of time used in the matrix i.e. the time data are computed according to the following formula for point of measurement:

$$t' = \begin{cases} t7 + 60*t6 + 1440*t5 & \text{(minutes: if col. 7 is given)} \\ t6 + 24*t5 & \text{(hours: if col. 6 is given but col. 7 is not)} \\ t5 & \text{(days: if only col. 5 is given)} \end{cases} \qquad (6.25)$$

In this formula, zeros are substituted for the elements of the eventually missing columns.

* The final values of the t variable will be related to an initial time i.e. they will be computed according to the following formula:

$$t = t' - t0 \qquad (6.26)$$

where t0 is the initial time to be specified at Question 22. Zero is substituted for it if Question 22 is not answered at all or it is answered but no initial time data are given for material m(j). To put it more precisely, the following convention is applied (cf. Section 2.2.2). Let us denote the number of lines given at Question 22 by n22. The following cases are possible:

if n22 = 1, the initial time data hold for all materials;

if n22 > 1, the initial time is taken as 0 for

m(j) > n22

while the initial time data are taken from line m(j) otherwise.

Let us denote the data taken from Question 22 by t1, t2, t3, and t4. If t1 < 0, the program sets t0 = 0 for the initial moment. In the opposite case i.e. when t1 > 0, the initial times for columns 5, 6, and 7 of the matrix "79" are obtained as

$$t05 = t1, \qquad (6.27a)$$

$$t06 = t2, \qquad (6.27b)$$

$$t07 = t3 + t4/60. \qquad (6.27c)$$

Now the initial time is computed according to the time unit defined at Equ. (6.25):

$$t0 = \begin{cases} t07 + 60*t06 + 1440*t05 & \text{(minutes)} \\ t07/60 + t06 + 24*t05 & \text{(hours)} \\ t07/1440 + t06/24 + t05 & \text{(days)} \end{cases} \qquad (6.28)$$

Control variable LT(j) becomes equal to 2.

(3) If none of columns 3, and 5 to 7 (incl.) are given in the matrix "79" for material m(j), the time data are taken from Question 64. If data are available at Question 64 for material m(j), two cases are possible:

* control variable LT(j) becomes equal to 1 and the values of parameters t_i and Δt_i (cf. Equ. (5.8) of Part 2) are taken from the subfile if option variable MT is equal to 1 (cf. Section 2.2.3);
* control variable LT(j) becomes equal to 2 and the values of the t variable are computed according to Equ. (2.2) if option variable MT is equal to 2 (cf. Section 2.2.3).

It is a data error if either Question 64 is not answered or, although it is answered, no data are given for material m(j).

6.3.3. Fitted data (y)

The values of the fitted data y are taken from column 10 of the matrix "79" for material m(j). It is a data error if column 10 is missing from the matrix. There is no other way of specifying y (unless spectra are evaluated, cf. Section 6.3.9).

6.3.4. Variances of the fitted data (y) and of the x variable

As the variances of the fitted data y and the x variable are stored in the same column of the data matrix, they are considered here in the same section. We shall denote by MWY the value of the option variable found in the subfile at Question 74 for material m(j) and by LWY(j) the value of the control variables determining the weighting option of y for run j. The significance of option variables MWX and LWX(j) is analogous in relation to Question 76 and the x variable, respectively. As a comparison of Table 2.2 of Part 2 and Table 2.5 of the present part shows, most values of option variables MWY and MWX have the same significance for tasks RFIT and EVAL. The values 0 and 7 are exceptions to this statement since the action of the program is different in case of x and y:

- if MWX = 0, this means that the variance of the x variable is zero; if MWY = 0, this corresponds to a Poissonian variable i.e. the program sets LWY(j) = 1;
- if MWX = 7, this generates an error message; if MWY = 7, the program computes the variances of y for all i, and the program sets LWY(j) = 3.

When the background correction is given as a separate material (whose number is denoted by mb(j) in Section 6.3.5), some complications can arise if the weighting options are different for materials m(j) and mb(j). In case of a mismatch, the variances are computed for both materials and option variable LWY(j) is set equal to 3. This occurs in the following cases:

- the weighting option for material mb(j) is different from 1 and 5;
- the weighting option is 5 for material mb(j) but it is not Poissonian for material m(j) (i.e. the option variable is neither 1 nor 5); if it is Poissonian, the program sets LWY(j) = 5;
- the weighting option is 1 for material mb(j) but material m(j) is not Poissonian.

Note that the cases when MWY = 1 and MWY = 5 (or MWX =1 and MWX = 5) are equivalent to one another when the program computes the variances.

As the subfile structure does not allow a pointwise specification of the variances, the value 3 is not allowed neither for MWX nor MWY. Would this occur, an error message is generated. This restriction excludes the case when LWX(j) = LWY(j) = 3 which is considered as an input error for task RFIT. Note finally that it is a data error if a negative or zero value occurs for y when LWY(j) = 1 or 5.

6.3.5. Additive correction

In the most general case, the additive correction is given by Equ. (2.13) of Part 2 as

$$b = (1) + (2) - (3)$$

where the terms are:

(1) = laboratory background during the measurement of y,

(2) = remanent activity,

(3) = laboratory background during the measurement of the remanent activity.

The remanent activity is specified through the data belonging to a special material whose number mra(j) has to be given at Question 75. It will be discussed after the discussion of the laboratory backgrounds i.e. components (1) and (3) of the additive correction. The laboratory background and its variance are specified at Questions 68 and 69, respectively. Let KORA and MWA be the option variables given there for the background and for its variance, respectively. (Both are related to material m(j).) Depending on their values, the program can either compute the additive correction and/or its variance already in the process of interpreting the subfile or perform only a transformation of the option variables.

The additive correction will be computed by the program (i.e. the data matrix will contain column 5) in the following three cases:

- KORA = 3 is specified at Question 68 for material m(j),

- the background is given in column 11 of the matrix "79" i.e. KORA = 1 at Question 68 for material m(j),
- mra(j) > 0 (i.e. there is a remanent activity for material m(j)).

Note that it is a data error if the two last cases apply for a spectrum evaluation (cf. Section 6.3.9). The variance of the additive correction will be explicitely given in the following two cases:

- KORA = 3 is specified at Question 68 for material m(j),
- there is a remanent activity for material m(j).

In all other cases, only the option variables are transformed (if at all). The possible values of variables KORA and MWA are discussed in Section 2.2.3 (see also Tables 2.4 and 2.5). The significance of the analogous option variables discussed in Sections 2.4 and 2.5 of Part 2 in relation to task RFIT are somewhat different. In order to emphasize this difference in the present section, we shall denote the values of these option variables by KRA(j) and LWA(j), respectively, resulting for run j from the interpretation of the subfile. In addition to this, there are further details which need to be taken into account in connection with the background:

- For KORA > 1, the value of option vaiable LWA(j) is determined by KORA alone and the program does not take into acount what eventually is given at Question 69 for material m(j).
- Being part of the additive correction, the background should be related to the same counting time T as y. In this respect, the treatment is different for constant and variable T (cf. Section 6.3.8).

All this is made clear in Table 6.3 and in the following remarks which show the correspondence between the possible values of the mentioned option variables.

Table 6.3. Conversion of the background option numbers

subfile option KORA	RFIT option KRA(j)	error option LWA(j)	the background is taken from:
0	0	0	no correction
1	1	MWA given at Question 69	column 11 of the matrix "79"
2	3	4	A, λ, $(\Delta A)^2$ from Quest. 68
3	1	3	computed from data given at Question 68
4	4	1	data for which the foil identifier is 0
5	4	1	data given for a special material
6	5	1	data for which the foil identifier is 0
7	5	1	data given for a special material
8	4	1	data for which the x variable is equal to x0

Meaning of and remarks to the individual options:

KORA = 0: The background correction is not applied.

KORA = 1: The background data are given in column 11 of the matrix "79". The variances of these data are determined on the basis of the data given at Question 69 for material m(j). (This is the only case when Question 69 is taken into account.) If Question 69 is not answered or MWA is equal to 0 for material m(j), the program sets LWA(j) = 1. It follows from this that the proper way of neglecting the variance is not to omit Question 69 but to give MWA = 2 or 4 with C = 0. Both the background data and parameter C should be related to the same counting time T as y. (This can be a problem for MWA = 2 if T is variable.) It is a

data error if:

* column 11 is missing in the matrix "79" for material m(j);
* MWA = 3 or 7 is specified at Question 68 for material m(j).

KORA = 2: The values of parameters A, λ, and $(\Delta A)^2$ are taken from Question 68. They will give the values of these parameters for run j corresponding to the run dependent case KRA(j) = 3. When counting time T is variable, the background data will be multiplied by its values. Therefore, parameter A and its variance should be related to unit counting time. When counting time T is constant, the background data will not be multiplied by it. Therefore, parameter A and its variance should be related to counting time T.

KORA = 3: As there is no equivalent of cycle dependent A and λ parameters for task RFIT, the background and its variance will be computed for all values of subscript i using the data given at Question 68 for material m(j). Correspondingly, the value of option variable KRA(j) is set equal to 1. When counting time T is variable, the background data will be multiplied by its values. Therefore, parameter A and its variance should be related to unit counting time. When counting time T is constant, the background data will not be multiplied by it. Therefore, parameter A and its variance should be related to counting time T.

KORA = 4 and 6: Parallelly to the x variable, the foil identifiers are also determined. As stated in Section 6.3.1.2 for this case, the x variable is determined according to option LX(j) = 2. The value of x is changed to x0 = -8888 for such rows of the data matrix for which the foil identifier is 0. The statistical behaviour of the background data treated according to options 4 and 6 is usually Poissonian. That is why the program sets LWA(j) = 1. (Would this be incorrect in the actual case, the background should be given as a separate material since then other error options are available, too, see remarks at KORA = 5 and 7.) The value of the interpolation variable n (to be given in the subfile for KORA = 4) is not taken into account: the program works always with n = 2.

KORA = 5 and 7: When the background is specified for run j through the data belonging to a special material, the data matrix is built up in several steps. Let mb(j) be the number of this special material. A data matrix is built up first by taking into account all data (except the background) given in the subfile for material m(j). Then, as a second step, another matrix is built up having the same columns as the previous ones. Finally, the two matrices are merged. In order to make them match, the following is necessary:

* If LX(j) = 1 for the first matrix, the values of the x variable are computed and the option variable is changed to LX(j) = 2. The value of the x variable for the background data is -8888.
* If LT(j) < 2 for the first matrix, the values of the t variable are computed and the option variable is changed to LT(j) = 2. The values of the t variable for the background will be those given in the subfile for material mb(j). It is a data error if no time data are specified for the latter.
* The values of y corresponding to the background measurements are taken from column 10 of the matrix "79" for material mb(j).
* The effect of this backgrond option on the variance of the fitted data is discussed in Section 6.3.4.
* If the first matrix contains columns for the calibration and decay corrections (cf. Sections 6.3.6 and 6.3.7), the corresponding columns of the background matrix will contain 1 and 0 for the correction factors and their variances, respectively.
* The correct treatment of the counting time and the dead time requires a match of options given in the subfile for materials m(j) and mb(j). In the following, we differentiate the corresponding option variables and parameters by subscripts y and b. As the values of option variables MWHy(j) and MWHb(j) can be 0, 2, and 4 (cf. Section 6.3.8), the resulting value of the option variable MWH(j) is determined according to Table 6.4. It follows from this table that the counting time becomes variable if it is constant separately for these matrices but these constant values are not equal. As to the dead time, the value given for material m(j) is taken also for material mb(j). This is justified since the background is usually measured at such low count rates that the dead correction is negligible.

Table 6.4. Cases when the counting time is variable

MWHy(j)	MWHb(j) = 0	MWHb(j) = 2	MWHb(j) = 4
0	0	data error	data error
2	data error	2 if $T_y = T_b$ 4 if $T_y \neq T_b$	4
4	data error	4	4

It is without any consequence that the value of option variable LWA(j) is set equal to 1 since the variance of the background is determined by the data given at

Question 74 for material mb(j). The value of the interpolation variable n (to be given in the subfile for KORA = 5) is not taken into account: the program works always with n = 2.

KORA = 8: This option is almost the same as option KORA = 4. The difference consists in that the foil identifiers are not needed but the value of the x variable is given in the subfile for the background monitor. The statistical behaviour of the background data treated according to options 4 and 6 is usually Poissonian. That is why the program sets LWA(j) = 1. (Would this be incorrect in the actual case, the background should be given as a separate material since then other error options are available, too, see at KORA = 5 and 7.) The value of interpolation variable n is not taken into account: the program works always with n = 2. Possible data errors are:

- x0 is the same as for the decay monitor.
- When foil identifiers are given in the subfile for material m(j), the program checks whether the same foil identifier belongs to all points for which x0 = x. An error message is generated in the opposite case.

When remanent activities need to be taken into account for material m(j), both the additive correction and its variance will be determined explicitely. Let mra(j) be the material number given at Question 75 for material m(j). When averaging is mentioned in the following, this means

* a weighted average in the general case where the weights are the reciprocals of the variances;
* an unweighted average if the variances are neglected;
* maximum likelihood estimates for options MWY = 5 (or MWA = 5):

$$\text{average} = \frac{\text{sum of counts}}{\text{sum of counting times}},$$

$$\text{variance} = \frac{\text{average}}{\text{sum of counting times}}.$$

The procedure consists in the following steps.

(1) The laboratory background and its variance are computed according to the option variables shown in the table above. The deviations from the standard are the following:

- the dead time correction is neglected;
- option 5 is taken instead of option KRA(j) = 4.

(2) The remanent activity data belonging to the same foil identifiers are averaged. (Option MWY = 7 is not allowed in this case.)

(3) If the laboratory background is given pointwise for the remanent activity measurements (i.e. KORA = 1 for material mra(j)), the data belonging to the same foil identifiers are also averaged. (As before, option MWA = 7 is not allowed.) Restrictions and special cases:

- Poissonian statistics is assumed if Question 69 is not answered or the option variable is 0 for material mra(j);
- Poissonian statistics is assumed if the option variable is 6 at Question 69 but the corresponding parameter C is 0;
- the variance of the background is taken as 0 if the option variable at Question 69 is 2 or 4 and the coresponding parameter C is equal to 0.

(4) All averages and their variances are related to unit counting time if it is given material mra(j). As to when the counting time is considered as given, cf. Section 6.3.8.

(5) At meterial mra(j), some remanant activity measurement should correspond to all foil identifiers which are mentioned in the subfile for material m(j). This gives the correspondence of the y data and the additive correction data. When Equ. (2.13) of Part 2 is applied, all components of additive correction b are related to the counting time specified for material m(j). In connection with this, it is a typical subfile error that the counting time is given only for one of meterials m(j) and mra(j), cf. Section 6.3.8.

Restrictions and typical data errors are:

- Only the following options are available for the specification of the laboratory backgorund of the remanent activity measurements: KORA = 0, 1, 2, and 6. Their significance is the same as for material m(j) (see Table 6.3).
- As to the counting time, the same conventions apply for KORA = 1 and 2 as for material m(j) (see Table 6.3).
- It is a data error if the foil identifiers are not given for one (or both) of materials m(j) and mra(j).
- It is a data error if, although the foil identifiers are given for both materials m(j) and mra(j), but a foil identifier mer ioned for material m(j) is not found at material mra(j).
- There are no monitor points in case of options which would involve them.

6.3.6. Foil calibration and macroflux corrections

The foil calibration correction may be related to both micro- and macroflux measurements while the macroflux correction is normally related to spectral index (or spectral ratio) measurements. Anyhow, the product of these two corrections factors appears in column 7 of the data matrix as the foil calibration correction. Neither the calibration nor the macroflux corrections are taken into account for CA01 type subfiles since the purpose of their evaluation is just the determination of the calibration factors (cf. Section 7.3.10). Let LWC(j) be the value of the option

variable which determines the variance of μ_c according to Table 2.2 of Part 2. The auxiliary parameter needed for some values of LWC(j) will be denoted by C.

The subfiles contain several places where information can be found concerning the foil calibration factors. If μ_c can be found at one of them, the other ones are left out of consideration. The program searches them in the following order:

(1) Column 12 of the matrix "79" for material m(j). If it is given, the program sets LWC(j) = 2 and C = 0. The foil identifiers are not needed unless the macroflux correction is applied (see below).

(2) Column 5 of the matrix "66" for material m(j). If it is given, the program sets LWC(j) = 2 and C = 0. The foil identifiers should be given in the usual way for the matrix "79" (cf. Section 6.1.1.1) and in column 2 of the matrix "66". Except the foil identifiers eventually belonging to the background and decay monitors, all foil identifiers mentioned in the matrix "79" should be mentioned in the corresponding matrix "66", too.

(3) If the identification number of the calibration data set is not 0 at Question 9 for material m(j), the program finds the identified data set in the calibration library (i.e. CLIB) and takes the calibration factors from it. Except the foil identifiers eventually belonging to the background and decay monitors, the calibration data set should contain a correction factor for all foil identifiers mentioned in the matrix "79". Option variable LWC(j) and parameter C are taken from the calibration data set. It is important to note that the logical number of CLIB is 62 to be specified in the respective job control DD statement.

No foil calibration is applied (or rather: μ_c = 1) if:

- column 12 is missing in the matrix "79" and
- column 5 is missing in the matrix "66" and
- the identification number is 0 at Question 9

for material m(j). Typical data errors are:

* the identified calibration data set is not found in CLIB;
* the foil identifiers are not given (except case (1));
* a foil identifier mentioned in the matrix "79" is not found in the matrix "66" or in the identified calibration data set.

The macroflux correction μ_z has to be specified at Questions 82 and 66. No macroflux is applied if Question 82 is not answered at all or it is answered but the value of option variable KORM is 0 for material m(j) (cf. Section 2.2.3). When KORM > 0, the data determining the values of μ_z are supposed to be given in column 3 of the matrix "66" for material m(j). Column 3 contains

for KORM = 1: the 2 values which give μ_z according to Equ. (2.9);

for KORM = 2: the values of μ_k explicitely.

In both cases, the foil identifiers should be given in the usual way for the matrix "79" (cf. Section 6.1.1.1) and in column 2 of the matrix "66". Except the foil identifiers eventually belonging to the background and decay monitors, all foil identifiers mentioned in the matrix "79" should be mentioned in the corresponding matrix "66", too.

Typical data errors are:

* the foil identifiers are not given;
* a foil identifier mentioned in the matrix "79" is not found in the matrix "66";
* KORM > 0 at Question 82 for material m(j) but Question 66 is not answered at all or there is no matrix "66" for material m(j).

Note that the conditional identifiers to be given in column 1 of the matrix "66" cannot be used neither for the calibration nor for the macroflux corrections. Furthermore, only the calibration factors taken from CLIB can have variances different from 0. The latter will be given pointwise if LWC(j) = 3. No variance is associated with the macroflux correction factors.

6.3.7. Decay correction

The decay correction is specified at Question 70. Let KORD be the option variable given there for material m(j). Although Question 71 may contain some information on the variance of the decay correction, this is not taken into account since the variance is determined by KORD (see below). Depending on its value, the program can either compute the decay correction already in the process of interpreting the subfile or perform only a transformation of option variables. From the point of view of the decay correction, the case when MODE = 30 requires a special treatment. Let us consider the general case first, and return to this special case at the end of this section.

There is no decay correction if Question 70 is not answered at all or KORD < 2 for material m(j). The decay correction factors are directly computed for KORD = 3. In all other cases, only the option variables are transformed (if at all). The possible values of variable KORD are discussed in Section 2.2.3 (see also Table 2.4). The significance of the analogous option variable discussed in Section 2.1 of Part 2 in relation to task RFIT is somewhat different. In order to emphasize this difference in the present section, we shall denote the value of this option variable by KRD(j) resulting for run j from the interpretation of the subfile. The option variable for the variance of the decay correction will be denoted by LWD(j). In addition to this, there are further details which need to be taken into account in connection with the decay correction:

- In the process of interpreting the subfile, the variances of the decay correction factors are never computed pointwise.

- For spectrum evaluations, KORD may not be greater than 3 (cf. Section 6.3.3).

All this is made clear in Table 6.5 and in the following remarks which show the correspondence between the possible values of the mentioned option variables.

Table 6.5. Conversion of option numbers for decay correction

subfile option KORD	RFIT option KRD(j)	error option LWD(j)	the background is taken from:
0 and 1	0	0	no correction
2	3	2 C = 0	λ taken from Question 70
3	1	2 C = 0	computed from data given at Question 70
4 (general case)	4	1	data for which the foil identifier is N
4 (MODE = 38)	6	1	see the remarks made below
5, 6, 7	data error		
8 (general case)	4	1	data for which the x variable is equal to x0
8 (MODE = 38)	6	1	see the remarks made below

Meaning of and remarks to the individual options:

KORD = 0 and 1: The decay correction is not applied. Note that there is no such option for the decay correction in column 11 of the matrix "79" for the background (cf. Section 6.3.5).

KORD = 2: The value of parameter λ is taken from Question 70. It will give the value of this parameter for run j corresponding to the run dependent case KRD(j) = 3. As stated in Section 2.1, no error can be associated with this decay correction. That is why the program sets C = 0.

KORD = 3: As there is no equivalent of cycle dependent λ para-

meters for task RFIT, the decay correction factors will be computed for all values of subscript i using the data given at Question 70 for material m(j). Correspondingly, the value of option variable KRD(j) is set equal to 1. As stated in Section 2.1, no error can be associated with this decay correction. That is why the program sets C = 0.

KORD = 4: When MODE = 38, this option is the same as KORD = 8. Therefore, only the cases of other fitting functions will be discussed here. Parallelly to the x variable, the foil identifiers are also determined. As stated in Section 6.3.1.2 for this case, the x variable is determined according to option LX(j) = 2. The value of x is changed to x0 = -9999 for such rows of the data matrix for which the foil identifier is equal to N where N is the foil identifier of the decay monitor to be given at Question 70. The statistical behaviour of the decay data treated according to option 4 is usually Poissonian. That is why the program sets LWD(j) = 1. The value of the interpolation variable n (to be given in the subfile for KORD = 4) is not taken into account: the program works always with n = 2

KORD = 5 to 7: These values of option variable KORD are treated as data errors.

KORD = 8: This option is almost the same as option KORD = 4. The difference consists in that the foil identifiers are not needed but the value of the x0 parameter is given in the subfile for the decay monitor. The statistical behaviour of the decay data treated according to option 4 is usually Poissonian. That is why the program sets LWD(j) = 1. The value of interpolation variable n is not taken into account: the program works always with n = 2. Possible data errors are:

- x0 is the same as for the background monitor.
- When foil identifiers are given in the subfile for material m(j), the program checks whether the same foil identifier belongs to all points for which x0 = x. An error message is generated in the opposite case.

The case when MODE = 38 has to be considered separately. As stated in Section 2.1 of Part 2, the option LWD(j) = 6 has been elaborated especially for this fitting function. The data for the decay monitor will be obtained from those belonging to thermal column positions P1TH(j) and P2TH(j) to be given within input data Group B3 (cf. Section 6.2). The round shaped foils will be identified by x0 = -9999. The same for the foils with hexagonal shape will be x0 = 9999.

In case of fitting function No. 30, only one option is possible for the decay correction (cf. Section 1.2.10 of Part 2). Therefore, the program does not take into account what eventually

is given at Question 70 but sets the following values for the option variables:

KRD(j) = 4, LWD(j) = 1,

and

$$x0 = \begin{cases} P1TH(j) \text{ for } j \text{ odd} \\ P2TH(j) \text{ for } j \text{ even} \end{cases}$$

where P1TH(j) and P2TH(j) are to be given within input data Group B2 (cf. Section 6.2).

It is a data error if no monitor points are found but the option applied for decay correction would involve them.

6.3.8. Counting time and dead time

The dead time (τ) is supposed to be given at Question 61. Its value is used for the determination of the dead time correction. It follows from Equ. (2.9) of Part 2 that the application of this correction necessitates also the knowledge of the counting time T. If the dead time is given but the counting time is not, this is a data error. The contrary case, however, is accepted: when the counting time is given but the dead time is not, the dead time correction factor will be equal to 1 for all values of subscript i. The dead time correction is not applied only if neither the dead time nor the counting time is given.

The program considers the dead time as given if its value is positive at Question 61 for material m(j). If Question 61 is not answered at all or it is answered but the value given for material m(j) is equal to 0, the program considers that the dead is not given for material m(j). It is a data error if a negative value is found at Question 61.

The computation of the dead time correction factor is only one of the uses of the counting time. Its other use consists in that it is a multiplicative correction if it is variable i.e. when T depends on subscript i (cf. Equ. (2.1b) of Part 2). Therefore, we need option variable LWH(j) in order to differentiate between the following three cases:

LWH(j) = 0: no dead time correction.

LWH(j) = 2: the dead time correction is applied and T is constant.

LWH(j) = 4: the dead time correction is applied, and T is variable.

These are transforms of option variable MWH defined in Section 2.3 of Part 2 and to be specified in Groups 8 and 13 of Chapter 5 of Part 2.

There are several ways for specifying the counting time.

They are tried by the program in the following order:

(1) When column 4 is given in the matrix "79" for material m(j), the values of T are taken from there and the program sets LWH(j) = 4 i.e. T is considered as variable. The program does not check whether the data actually given in column 4 are really different among themselves. Only positive values are accepted for T. The opositie case is a data error.

(2) When the previous option is not applicable, the program takes the data given at Question 62 for material m(j). The possible cases are:

T > 0: the program sets LWH(j) = 2 and this constant value is used as the counting time.

T < 0: this is a data error.

T = 0: see at (3) below.

(3) When Question 62 is not answered at all or it is answered but T = 0 is given for material m(j), the program concludes that the counting time is not given (if, of course, case (1) is not applicable either). This can have two consequences:

- the program sets LWH(j) = 0 if the dead time is not given;
- an error message is generated if the dead time is given.

Remarks:

- Option LWH(j) = 4 is not accepted for a spectrum evaluation (cf. Section 6.3.9).
- When the counting time turns out to be constant for material m(j), this does not mean, however, that it remains such as results of the final data compilation. As mentioned in Section 6.3.5, T becomes variable when the background is given for material m(j) via the data belonging to a special material mb(j), and either T is variable for it or, although T is constant also for material mb(j), these constant values are different for materials m(j) and mb(j).

6.3.9. Evaluation of spectra (Question 80)

It is pointed out at the beginning of Section 6.3 that the evaluation of the spectra stored at Question 80 is discussed separately since, in many respects, their treatment is simpler than that of the data stored at Question 73. Furthermore, several restrictions should be observed. Some of them mentioned individually in Sections 6.3.1 to 6.3.8 (incl.). They are summarized in the present section. Spectrum evaluation is allowed only for subfile types DRDH, DRDT, MEAS, R281, R282, and R283 (cf. Table 2.1). As stated in Section 6.2, it is in Group B4 that the contents of the runs j = 1, 2, ..., J are defined.

The main difference between the general case and the case of spectrum evaluations consists in that the fitted data i.e. the values of the y variable are taken for run j from spectrum S(j) stored at Question 80 for material m(j) (to be specified in Group B4) and the data eventually stored at Questions 63, 66, 78, and 79 are ignored. It follows from this that only such options are allowed for at the other questions which do not involve pointwise data i.e. which do not require the interpretation of the columns of the matrices "79" and/or "66". Sections 6.3.1 to 6.3.8 (incl.) discuss how the individual columns of the data matrix are built up for run j in the general case. When, on the contrary, spectra are evaluated, the same thing is done as follows:

- Option variable LX(j) is always set equal to 1 and the corresponding parameters x_{j1} and Δx_j are taken from spectrum S(j) specified in Group B4 (cf. also Section 2.2.3). This means that the x variable will be computed for point i according to Equ. (5.6) of Part 2 i.e.

$$x_{ji} = x_{j1} + (i-1)\Delta x_j$$

- The t variable is needed if some decay or background corrections are applied (see below which ones are accepted). It is determined as described in Section 6.3.2 but this can result in only option LT(j) = 1 since it is a data error if LT(j) = 2 results.
- The values of the y variable are taken from spectrum S(j) specified in Group B4. Note that the total number of the channels in the spectrum will give the total number of points belonging to run j.
- The variance of x and/or y is determined as in the general case (cf. Section 6.3.4).
- Options 1 and 3 are not accepted at Question 68 for the background correction. In addition to this, no remanent activity may be taken into account.
- The data eventually given in the subfile for the foil calibration and the macroflux correction are ignored.
- Only options 2 and 3 are accepted at Question 70 for the decay decay correction (cf. Section 6.3.7).
- Only options LWH(j) = 0 and 2 are accepted for the dead time and counting time. If the interpretation of the subfile results in LWH(j) = 4, this is a data error.

7. Task CALB: manipulating files EDF and CLIB

The present chapter is devoted to two files: the library of calibration factors (CLIB) and the evaluated data file (EDF). They are manipulated (listed, corrected, deleted etc.) by task CALB. In addition to this, these files are used by program RFIT also for other purposes whence they are referred to also in other chapters of this user's manual:

(1) The calibration library is a complement to the file of raw experimental data (PDF) and is referred to at Question 9 (cf. Sections 2.2.2 and 6.3.6).

(2) The evaluated data file contains fields (distributions) which are estimated by using certain fitting functions such as NNo. 20, 34, 43, etc. (see Chapter 1 of Part 2). It is an output file for tasks RFIT, SPEC, and EVAL (see Chapters 5 and 7 of Part 2 and Section 6.2 of the present part, respectively). It can be an input file for tasks RFIT and SPEC (see Part 2).

When one of EDF and CLIB is manipulated by task CALB or is used by tasks RFIT, EVAL, and SPEC independently of the other file, its peripheral number can be either 62 or 64. When they are used simultaneously, the peripheral number of CLIB is normally 62 while that of EDF is 64. However, the user may deviate from this. There is only one restriction: the peripheral number of the calibration library is always 62 (cf. Section 6.3.6). All other uses of these files go with keywords CLIB and BIBL which determine the peripheral number as 62 or 64, respectively.

If the user deems it expedient, it is not necessary to separate EDF from CLIB physically: both may be in the same sequential file. Their physical separation is useful only from practical points of view since CLIB is a more or less firm file while EDF is frequently changed. Thus, their physical separation helps to protect CLIB against eventual user errors. Anyhow, the peripheral numbers used in the respective job control DD statements and the data given in the SYSIN input should be carefully in accordance with one another.

7.1. Structure of the EDF or CLIB files

The structure of files EDF and CLIB is identical. Both are sequential files in which the record length is 80 bytes (20 words) at maximum. The end of the useful information stored in them is marked with a record containing only 'LAST' following the last data set. Both files are divided into independent data sets the internal structure of which is rather simple. They consist of the following records (or groups of records):

Record 1: ID

where ID is the identification number of the data set. It is normally an integer which may not exceed the 31th power of 2. (In order to avoid overflow safely, it is advisable to use only 9 decimal digits at maximum.) Although the program accepts any integer, it is recommended to use only positive numbers. This is the number of reference which is used at Question 9 of PDF and in connection with tasks RFIT, EVAL, and SPEC. As stated above, the program stops reading the file if it finds ID = 'LAST'. When ID > 1000 and its first digit is 7, a special convention applies defined at Record 4.

Record 2: N, MW, C

where

N is the total numbers of positions in the data set,

MW is an option variable which determines the way of computing the variances of the field values stored in the data set; refer to Record 4 concerning the options defined by the possible values of MW,

C is a parameter which is needed (in case of certain values of MW) for the computation of the variances.

Record 3: Title

where "Title" is a string of 80 characters in length (at maximum). It is practical to give it in order to explain the siginificance of the data stored in the data set, especially in order to differentiate the data set from others with similar physical contents. For example, if a set of activation foils is calibrated several times, this title record is the best means of telling precisely under which conditions the results of the individual calibrations can be applied. In case of EDF, the title record is the best means for distinguishing between various evaluations of the same subfiles. When the data set is an output file for tasks RFIT, EVAL, and SPEC, its "Title" will be the title of the fitting problem (see Chapters 5 or 7 of Part 2 or Chapter 6 of the present part).

Records 4: data matrix

The number of columns of the matrix is determined by the last decimal digit of MW i.e. by

$$MW' = MW - [MW/10]. \qquad (7.1)$$

(Here, like in some other chapters of this user's manual, [z] is the integer part of real number z, see e.g. Equ. (6.10).) If MW' = 3, the matrix has 3 columns:

position, field, variance (or error)

while it has only 2 columns otherwise:

position, field.

According to the sense, the matrix has N rows. The physical meanings of these quantities depend on the case at hand:

(1) "field" is the value of the evaluated distribution in case of EDF (i.e. Ψ in case of fitting functions NNo. 20, 21, 34, etc.) while it is the calibration factor in case of CLIB.
(2) The physical significance of "position" depends on the particular fitting function used for producing the data set at hand (see Chapters 1 and 5 of Part 2): in case of CLIB, "position" is the foil identifier while it is some core position for EDF. When the data set has been produced as results of using fitting functions NNo. 38 and 40, "position" is alphanumeric. ID is special in such cases: its first digit should be 7 and ID > 1000.
(3) The variance of "field" is given pointwise in the data set if MW' = 3 while it is computed according to Table 2.5 using the value of parameter C for other values of MW'. The entry to this table is now MW'. It follows from this that, according to Table 2.5, parameter C has no direct siginificance if MW' = 3. Therefore, C may be used in this case for distinguishing between the following two possibilities:

    if C = 0, column 3 of the matrix contains variances,

    if C = 0, column 3 of the matrix contains errors or standard deviations (i.e. the square roots of the variances).

Records 5: correlation matrix of the field

The correlation matrix of the "field" values (i.e. of the data stored in column 2) is stored in the data set if MW > 10. Its (i,j) element gives the correlation coefficient of "field" elements i and j (i.e. their covariance divided by their standard deviations). As this matrix is symmetric and all elements of its main diagonal are equal to 1, it is sufficient to store only its upper half without the main

diagonal. Thus, the total number of the significant elements is N(N-1)/2. Although the program accepts a correlation matrix for both EDF and CLIB, it is ignored in case of the calibration factors (cf. Sections 6.3.6).

7.2. Overall structure of the input data for task CALB

All manipulations with files EDF and CLIB are controlled by task keyword CALB ("calibration"). The latter may not go alone but only together with one or more other task keywords discussed in Chapter 5 and summarized in Table 7.1. They allow to perform most of the manipulations which can be applied to the subfiles composing PDF. The peripheral number to be given in the DD statement related to the file on which the manipulations are performed is normally 62. If it is desirable to use peripheral number 64, one has to put keyword

```
BIBL
```

in the line immediately following the line containing CALB. Taking the correction of some data sets as an example, the general structure of the input data can be one of the following three:

```
CALB CORR
 input data for CORR
```

In this case, the corrections will be related to the file whose peripheral number is 62. In other words, the default value of the peripheral number is 62. When keyword BIBL is given i.e. when the input data have the form

```
CALB CORR
BIBL
 input data for CORR
```

the corrections will be related to the file whose peripheral number is 64. If BIBL is replaced by keyword CLIB, this sets the peripheral number equal to 62. It follows from this that the input data

```
CALB CORR
CLIB
 input data for CORR
```

have the same effect as those shown in the first example. Consequently, it is superfluous to give keyword CLIB.

7.3. Input data for manipulating EDF or CLIB data sets

The task keywords which may go along with task keyword CALB are summarized in Table 7.1. Sections 7.3.1. to 7.3.10 (incl.) discuss them in detail. In this combination, the functions of some

of them are different from those which they have in connection with PDF (see Chapters 2 to 5, incl.). One has to be careful when several task keywords are given in the same line as CALB. The problem is that the program can not tell where the input lines end for the second keyword or where they start for the third one. Therefore, only such keyword combinations are recommended in which the third and further keywords do not require input data of their own. In practice, there are only two combinations which can be continued with further task keywords in the same line:

CALB NEW

and

CALB CORR

We can see examples of their combination in Sections 7.3.1 to 7.3.10 (incl.).

Some of the tasks discussed below involve the use of temporary disk file 77. As mentioned in other chapters of the present part, file 77 is used for subfile operations, too. One has to take into account the following remarks in connection with this:

- the subfiles eventually stored on file 77 are left unchanged;
- the space available on file 77 depends on the subfile operations eventually preceding the actual use of task CALB.

It is added to this last remark that it is a data error if the space available in file 77 is not sufficient. Would this occur in case of a large library, it should be split in two separate libraries.

7.3.1. Creating new data sets from SYSIN (task NEW)

New data sets can be included into the EDF or CLIB files by the following combination of task keywords:

CALB NEW

Each new data set is first created in the core memory and it is recorded to the file only if the data set could be created without input errors. The newly included data sets will be recorded behind the exisiting ones. After every successfully created data set, the file is closed i.e. 'LAST' is recorded at the end of the file so that eventual input errors could not do any harm to the file. The input data for the consecutive new data sets may follow one another until their sequence is cut by an task keyword (see Table 2.1 of Part 1). The maximum number of the data sets created in this way is 20. The identification numbers of the new data sets are noted in the core memory for being eventually used later by other tasks specified in the same line as for example

CALB NEW LIST

The input data to be given for each new data set have the same structure as the data set itself. In order to still make a difference between this section and Section 7.1, we use the expression "group" here instead of "record". Each of the following groups should be started in separate input lines. Note that Groups N1 to N4 (incl.) are mandatory but Group N5 is optional.

Group N1: ID
---

where ID is the identification number of the data set. The program does not check whether it is identical with the identification numbers of the existing data sets.

Group N2: Title
---

where "Title" is a string of 80 charatecters in length (at maximum) describing the contents of the data set.

Group N3: N, MW, C
---

where

N is the total numbers of positions in the data set; it may not exceed 500; if MW > 10, it is limited to 100 (see at Group N5),

MW is an option variable which determines the way of computing the variances of the field values,

C is a parameter which is needed (in case of certain values of MW, see Table 2.5) for the computation of the variances.

Group N4: data matrix

The number of columns of the matrix depends on the value of control variable MW. If MW' = 3 as computed according to Equ. (7.1), the matrix has 3 columns:

position, field, variance (or error)

while it has only 2 columns otherwise:

position, field.

The matrix has N rows. It is given row by row. Depending on MW' (see above), the total number of the matrix elements is 3N or 2N. There is no restriction as to how the elements are broken into input lines. If MW' = 3, the third column of the matrix contains the variances if C = 0 while it contains the errors or standard deviations (i.e. the square roots of the variances) otherwise. (See the explanations made at Record 4.)

Group N5: correlation matrix of the field values

The correlation matrix should be specified only if MW > 10. As the correlation matrix is symmetric and all elements of the main diagonal are equal to 1, only the elements of the upper half are given row by row leaving out the elements of the main diagonal. Their total number is N(N-1)/2. There is no restriction as to how they are broken into input lines. The maximum number of elements which can be handled by the program is 5000 whence N may not exceed 100 if MW > 10.

7.3.2. Correcting existing data sets (task CORR)

Data sets already existing in the EDF or CLIB files can be corrected by the following combination of task keywords:

CALB CORR

The corrections are performed in the following way. The existing data sets are read one by one into the core memory. The data sets to be corrected are corrected in the core memory. After successfully terminating the corrections, the corrected versions of the data sets are put to temporary file 77 (cf. Table 5.2). If a data set is not to be corrected, it is put to file 77 unchanged. When all data sets are copied to file 77 (in corrected or original forms), the contents of file 77 are copied back to files EDF or CLIB overwriting their original contents. Would one of the corrections be unsuccessful, the original file EDF or CLIP is left unchanged. It follows from this that the correction of the data sets of EDF and CLIB is different in several respects from that of the subfiles (see Chapter 4). The differences consist in the following:

- The input data for correcting the individual data sets should be given in the same order as the data sets are stored in the file since the program cannot find the data sets which are mentioned in wrong order.
- The original file is left unchanged in case of input errors. Even those data sets will remain unchanged the correction of which was successful.
- If a data set is corrected, its original version is lost.

The input data for the consecutive corrections may follow one another until their sequence is cut by an task keyword (see Table 2.1 of Part 1). The maximum number of the data sets which may be corrected in this way is 20. The identification numbers of the corrected data sets are noted in the core memory for being eventually used later by other tasks specified in the same line as for example

CALB CORR LIST

In order to have some analogy with task NEW , the structure of the input data to be given for each corrected data set are formulated

in terms of data groups C1, C2, etc. which roughly correspond to groups N1, N2, etc. Each of the groups should be started in separate input lines. The program distinguishes between them according to the number of asterisKs (*) introducing the line:

Group C2	I	no asterisK
Group C3	I	**
Group C4	I	*

From among these data groups, Group C1 is mandatory and is always the first one while the other ones are optional and may be given in arbitrary order. However, it is necessary to taKe into account that each one of them is related to the results of the application of the previous ones. With these introductory remarKs, the input data for correcting a data set should be formulated in the following way.

Group C1: ID, NC

where

ID is the identification number of the data set to be corrected. For the consecutive corrections, their order should correspond to the order according to which the data sets are stored in the file.

NC gives the total number of the correction steps. (Every application of Groups C2, C3, and C4 is considered as a separate correction step.)

Group C2: Title

where "Title" is a string of 80 charatecters in length (at maximum) describing the contents of the data set. One has to be careful with this data group: all input lines which do not start with an asterisK are considered as Group C2. This remarK is especially useful when one looKs for the reason of some input error: all input data lines which cannot be interpreted as Groups C3 or C4, will be treated as Group C2 by the program.

Group C3a: ** MW, C

or

Group C3b: ** MW * C

where

MW is the option variable which determines the way of computing the variances of the field values,

C is a parameter which is needed (in case of certain values of MW, see Table 2.5) for the computation of the variances.

This data group corresponds to Group N3 with the difference that parameter N i.e. the total number of the positions is not given since it is known (either from the original data set or from the results of the previous corrections). Depending on the original and new values of parameters MW and C, the specification of further subgroups can also be required:

Subgroup C34: column 3 of the matrix

Under conditions to be specified below, the variances or the standard deviations need to be given for all field values: these are N data which will form column 3 of the matrix. There is no restriction as to how they are broken into input lines. Warning: N is not necessarily the total number of the field values in the original data set since it might have been changed as results of the previous correction steps.

Subgroup C35: correlation matrix

Under conditions to be specified below, the correlation matrix of the field values need to be given: these are N(N-1)/2 values which are the elements of the upper half of the matrix without the main diadonal (like at Group N5). There is no restriction as to how they are broken into input lines. It follows from this that the correlation matrix can be corrected only as a whole. Warning: N is not necessarily the total number of the field values in the original data set since it might have been changed as results of the previous correction steps.

Let us first consider the cases which have to be distinguished from one another with respect to the variances. The possible cases are summarized in Table 7.2. As before, MW' is given by Equ. (7.1). Note that the values stored in column 3 are squared if parameter C changes from non-zero to zero while the square roots are taken vice versa.

Table 7.2. Cases of correctiong the variances

original / corrected	MW' ≠ 3	MW' = 3, C ≠ 0	MW' = 3, C = 0
MW' ≠ 3	the matrix is not changed	column 3 is deleted	column 3 is deleted
MW' = 3, C ≠ 0	Subgroup C34 is read into column 3	the matrix is not changed	variances converted into errors
MW' = 3, C = 0	Subgroup C34 is read into column 3	errors converted into variances	the matrix is not changed

With respect to the correlation matrix, other cases have to be distinguished from one another. First of all, if Group C3b is given i.e. an asterisk stands between MW and C, a new correlation matrix should always be given as Subgroup C35, and the value of parameter MW is set equal to MW' + 10 where MW' is computed by Equ. (7.1) with MW as given in Group C3b. Let us consider now the case when Group C3a is given i.e. there is no asterisk between MW and C. The cases which are then possible are summarized in Table 7.3.

Table 7.3. Cases of correcting the correlation matrix

original / corrected	MW < 10	MW > 10
MW < 10	correlation matrix unchanged	the correlation matrix deleted
MW > 10	Subgroup C35 is read into the correlation matr.	no change in the correlation matrix

The following important remark should be taken into account with respect to the correlation matrix. When the field values are changed, some of them are deleted and new ones are inserted (see at Group C4) but the correlation matrix is not changed by the program. It follows from this that the correlation matrix should be changed correspondingly by the user (if necessary). When number N of the field values

changes and the corrected version of the data set contains a correlation matrix, the program checks whether a new correlation matrix has been given for the final value of N. If not, an error message is generated.

Group C4: corrections of the data matrix

The starting line of Group C4 is introduced with one asterisk and it has the form:

* n1, n2, n3

where

n1 is the number of first row to be corrected,

n2 is the total number of the rows to be deleted (it may be 0),

n3 is the total number of the rows to be inserted (it may be 0).

Every line of this type is followed by the elements of the newly inserted rows i.e. 3 or 2 times n3 numbers depending on whether MW' = 3 or not. (MW' is computed according to Equ. (7.1) for the actual value of parameter MW.) There is no restriction as to how the elements are broken into input lines. If MW' = 3, the third column of the matrix contains the variances if C = 0 while it contains the standard deviations (i.e. the square roots of the variances) otherwise. According to the sense, no data are expected if n3 = 0.

Remarks:

- In the consecutive applications of Group C4, the values of n1 should increase monotonously. In addition to this, no reference may be made to a previously deleted row: the value of n1 should be more than the value the sum (n1+n2-1) had in the previous application of Group C4.
- Parameters n1 and n2 should be related to the original version of the data set. Furthermore, the value of (n1+n2) may not go beyond the original number of the rows.
- The new lines will be inserted preceding line n1 if n2 = 0. Adding new rows behind the existing ones is not so simple: the last row should be deleted and this row together with the new rows should be inserted into the matrix.
- The program accepts that n3 is not given. If so, it is set equal to 0.
- The total number of the rows (i.e. N) will be computed by the program automatically.

7.3.3. Selection of data sets

Most of the tasks discussed in the following sections involve input data for selecting the data sets concerned by the operations. As their structure is practically the same for the these tasks, it is formulated separately in the present section. (Some slight deviations are possible in connection with certain tasks, see below.) Like in case of subfiles, there are two ways of selecting data sets: explicite and implicite. The format of the corresponding input data is the following:

N, [ID(j), j = 1, 2, ..., N]

where

N is the total number of the data set identifiers; it is restricted by the condition

0 < N < 21;

ID(j) identifies a data set to be selected.

This way of selecting the data sets corresponds to the explicite subfile selection mode discussed in Section 5.2.1. The equivalent of the implicite subfile selection mode (cf. Section 5.2.2) is the use of asterisks (*) in place of the last digits of some (or all) of the ID(j). Instead of giving sophisticated general definitions, this is explained by the following example (other data of this form can be interpreted in an analogous way): ID(j) = 15** means all those four digits identification numbers in which the first two digits are 15. In other words, ID(j) = 15** results in the selection of all such data sets whose identification numbers range from 1500 to 1599. The explicite and implicite data set selection modes may be combined with each other i.e. input data of the following form are accepted:

```
 6  1401  1402  16**  16***  7013  8174
```

which mean the selection of data sets NNo. 1401, 1402, 7013, 8174, for one hand, and those data sets whose identifiers fall in the closed intervals [1600,1699] and [16000,16999], for the other hand.

For some tasks, the program accepts that no input data are given for data set selection. This is the case when an task keyword (see Table 2.1 of Part 1) or a subfile identifier (see Table 2.1 of the present part) is given in the input line in which the program looks for the data set selection data. When this occurs, the program selects all the data sets found in the source file. This means that the omission of the data set selection data has the same effect as the data

```
1  0*********
```

Remarks:

- Since integers exceeding the 31th power of 2 cause overflow, it is advisable to restrict the identifiers to 9 digits. That is why the program takes only 9 asterisks into account at maximum. (If more asterisks are given, the latter are not taken into account.)
- All operations are ineffective if one or more explicitely selected data sets are not found in the library. There is no such restriction for the implicitely selected data sets.
- Would several data sets have the same identification number, only that one is selected explicitely from among them which occurs first in the library. The other ones can be selected explicitely by repeating the common identifier the necessary number of times. According to the sense, there is no such difficulty in case of the implicite selection mode.
- It might be not superfluous to emphasize that the input value of N is the total number of the ID(j) values given (see the example above). When only the explicite selection mode is applied, this is equal to the total number of the actually selected data sets while no such equality holds for the implicitely selected data sets.

7.3.4. Opening the file (task DMMY)

All manipulations with the EDF and CLIB files discussed in the present chapter assume that some data sets already exist in them. It follows from this that the files should be opened somehow. That is the role of the following keyword combination:

```
CALB DMMY
```

As a result of this, the closing record containing only 'LAST' is recorded to the file. This is practically the same what task DMMY does as discussed in Section 5.4.1 in relation to PDF. Warning: one has to be careful with the use of DMMY since it deletes all information which might be in the library.

7.3.5. Copying data sets (tasks COPY and ADD)

The program allows to make copies in two directions:

a) task COPY: copying the existing library (as a whole or parts of it) to a new sequential file;

b) task ADD : copying a library (as a whole or parts of it) behind the data sets already stored in the existing library.

The peripheral number of the destination file is 61 in both cases. It is a scratch file for task COPY while it should have been opened previously as a library (at least by task DMMY, cf. Section 7.3.3).

The peripheral number of the source library can be either 62 or 64 depending on whether CLIB or BIBL has been specified in the input line following task keyword CALB (cf. Section 7.2). The default option is 62. The format of the input data for selecting the copied data sets is discussed in Section 7.3.3. The selected data sets are first copied to temporary disk file 77. When all data sets have been read from the source file, the selected ones are copied to the destination file only if all data sets selected explicitely have been found in the source file. The destination file is left unchanged in the opposite case.

Examples of the input data:

```
CALB COPY (or ADD )
BIBL (or CLIB)
 6  1401  1402  16** 16***  7013  8174
```

or

```
CALB COPY (or ADD )
 6  1401  1402  16** 16***  7013  8174
```

7.3.6. Deleting data sets (task DEL)

Some of the data sets can be deleted from the library by using task DEL . The input data for selecting the deleted data sets are discussed in Section 7.3.3. The program copies all data sets which are not selected to temporary disk file 77. When this is finished, the data sets are copied from the disk file back to the library (destroying its original contents). This latter operation is not performed if

* some of the explicitely selected data sets are not found in the library;
* the operation would delete all data sets from the library.

It follows from this last condition that it is an input error if no data are given for data set selection (cf. Section 7.3.3).

Examples of the input data:

```
CALB DEL
BIBL (or CLIB)
 6  1401  1402  16** 16***  7013  8174
```

or

```
CALB DEL
 6  1401  1402  16** 16***  7013  8174
```

7.3.7. Data set inventory (tasks SUBF and SUBT)

Sometimes it can be interesting to know which data sets are stored in the library. Tasks SUBF and SUBT allow to print the identification numbers of the selected data sets together with the total numbers of the field values stored in each of them. The input data for selecting the data sets are discussed in Section 7.3.3. The difference between tasks SUBF and SUBT consists in that no input data may be given in case of task SUBT since this means printing all of the data sets.

Examples of the input data:

```
CALB SUBF
BIBL (or CLIB)
 6  1401  1402  16** 16***  7013  8174
```

or

```
CALB SUBF
 6  1401  1402  16** 16***  7013  8174
```

7.3.8. Listing data sets (task LIST)

The contents of some selected data sets can be printed by using task LIST. The input data for selecting the data sets are discussed in Section 7.3.3.

Examples of the input data:

```
CALB LIST
BIBL (or CLIB)
 6  1401  1402  16** 16***  7013  8174
```

or

```
CALB LIST
 6  1401  1402  16** 16***  7013  8174
```

When the "position" data are alphanumeric according to the data set identifier (see at Record 4 in Section 7.1), the listing is printed in the corresponding format.

7.3.9. Renaming data sets (task RENM)

The identification numbers of some data sets can be changed by using task RENM. The input data have the following format:

$$N, [ID1(j), ID2(j), j = 1, 2, \ldots, N]$$

where

N is the total number of the data sets to be renamed; it is restricted by the condition

$$0 < N < 21;$$

ID1(j) is the old identification number of a data set to be renamed;

ID2(j) is the new identification number of the respective data set.

The action of the program will be the following. All data sets are copied first to temporary disk file 77. Those for which the identification number is equal to one of the ID1(j) get the new one (i.e. ID2(j)) before being copied to the disk file. When this is finished, the data sets are copied from the disk file back to the library (destroying its original contents). This latter operation is not performed if some of the data sets identified by the ID1(j) are not found in the library.

Examples of the input data:

```
CALB RENM
BIBL (or CLIB)
 6  1401  1402  1611 11611  7013  8013
```

or

```
CALB RENM
 6  1401  1402  1611 11611  7013  8013
```

Would several data sets have the same identification number, only that one is renamed from among them which occurs first in the library. The other ones can be renamed by repeating (as ID1(j)) the common identification number the necessary number of times.

7.3.10. Evaluating calibration measurements (task EVAL)

Task EVAL has been elaborated for evaluating CA01 type subfiles and for including the results of the evaluation into the calibration library (i.e. CLIB). The overall structure of the input data is the following:

```
CALB EVAL
  :
data for subfile selection
  :
BIBL (or CLIB)
  :
input data for evaluation
  :
END
```

or

```
CALB EVAL
  :
data for subfile selection
  :
  :
input data for evaluation
  :
END
```

The subfile selection may be either explicite or implicite in the sense of Section 5.2. It is an input error if no data are given for subfile selection following the line containing CALB EVAL. The effects of giving or omitting Keywords BIBL or CLIB are discussed in Section 7.2. The "input data for evaluation" have to be given as explained in Section 6.2.

It follows from this that the effect of Keyword combination CALB EVAL seems to be the same as the effect of Keyword EVAL used alone. This is really the case when input data Group C (of task EVAL) is given among the "input data for evaluation" (according to Section 6.2). CALB EVAL and EVAL have different effects only for such fitting problems for which Group C is not given. If this occurs, the program's action is the following:

- it takes the calibration data set numbers from Question 9 of the subfiles determined for each run; it is a data error if they are not equal or they are not positive;
- this number will be the identification number of the new data set to be included into the library (otherwise, this number is given as part of Group C);
- when the fitting is successfully terminated, the estimated field is put to the library (only for MODE = 20, 21, 34, 43, 47, 48, and 49).

It follows from this that this way of using CALB EVAL has some reasonable meaning only for CA01 type measurements since these are the only ones for which the calibration data sets specified at Question 9 are not taken into account as calibration factors (cf. Section 8.3.6). In spite of this, the program does not check the subfile type for which task CALB EVAL is used.

Table 7.1. Task Keywords which may be used with Keyword CALB

task Keyword	section where discussed	input file	output file
ADD	7.3.5	61	77, 62 or 64
COPY	7.3.5	61	77, 62 or 64
CORR	7.3.2	62 or 64	77, 62 or 64
DEL	7.3.6	62 or 64	77, 62 or 64
DMMY	7.3.4	-	62 or 64
EVAL	7.3.10	80, 82, 77, 62 or 64	77, 62 or 64
LIST	7.3.8	62 or 64	-
NEW	7.3.1	-	66 or 64
RENM	7.3.9	62 or 64	77, 62 or 64
SUBF	7.3.7	62 or 64	-
SUBT	7.3.7	62 or 64	-

References

[1] Z. Szatmáry et al., Experimental investigations of the Physical Properties of VVER-type uranium-water lattices, Akadémiai Kiadó, Budapest, 1985.
[2] E. Kondor, Z. Szatmáry, L. Túri, System for storing raw experimental data, report KFKI-ZR-6/402/1977 (in Russian).
[3] Z. Szatmáry, Program RFIT. Storing and evaluating experimental data, report KFKI-ZR-6/414/1979 (in Russian).
[4] I. Vidovszky, Sketches for PDF, to be published.
[5] F. Adorján, Description of the CAMON system, private communication.

The issues of the KFKI preprint/report series are classified as follows:

A. Particle and Nuclear Physics
B. General Relativity and Gravitation
C. Cosmic Rays and Space Research
D. Fusion and Plasma Physics
E. Solid State Physics
F. Semiconductor and Bubble Memory Physics and Technology
G. Nuclear Reactor Physics and Technology
H. Laboratory, Biomedical and Nuclear Reactor Electronics
I. Mechanical, Precision Mechanical and Nuclear Engineering
J. Analytical and Physical Chemistry
K. Health Physics
L. Vibration Analysis, CAD, CAM
M. Hardware and Software Development, Computer Applications, Programming
N. Computer Design, CAMAC, Computer Controlled Measurements

The complete series or issues discussing one or more of the subjects can be ordered; institutions are kindly requested to contact the KFKI Library, individuals the authors.

Title and classification of the issues published this year:

KFKI-1991-01/M A. Ambrozy: Experience during applying the ORACLE 4GL system (in Hungarian)

KFKI-1991-02/M S. Dibuz Wagner: A knowledge base for protocols

KFKI-1991-03/G Gy. Ézsöl et al.: Adaptation of RELAP5/MOD2 code: Analysis of IAEA SPE-3 (in Hungarian)

KFKI-1991-04/G Gy. Ézsöl et al.: IAEA SPE-3. Valuation of measurement results (in Hungarian)

KFKI-1991-05/G A. Takács et al.: SMABRE - as the dynamical model of the primary circuit for the code KARATE (in Hungarian)

KFKI-1991-06/A J. Balog et al.: Locality problem for the Liouville field

KFKI-1991-07/E N. Éber et al.: Linear electromechanical effect in a polymeric ferroelectric liquid crystal

KFKI-1991-08/C B. Lukács: Prices and sunspots

KFKI-1991-09/E M.V. Kozlovsky et al.: A chiral side-chain polymer with layered structure

KFKI-1991-10/G P. Vértes: Calculation of transmission and other functionals from evaluated data in ENDF format by means of personal computers

KFKI-1991-11/E F. Tóth: Non destructive testing methods (in Hungarian)

KFKI-1991-12/H Vera Jánossy et al.: Multichannel recordings and data analysis from spinal cord explants and cell cultures

KFKI-1991-13/G Z. Szatmáry: User's manual of program RFIT. Part 1. General description

KFKI-1991-14/G Z. Szatmáry: User's manual of program RFIT. Part 2. Parameter estimation

KFKI-1991-15/G Z. Szatmáry: User's manual of program RFIT. Part 3. The data files

Kiadja a Központi Fizikai Kutató Intézet
Felelős kiadó: Gadó János
Szakmai lektor: Gadó János
Nyelvi lektor: Bod László
Példányszám: 229 Törzsszám: 91-144
Készült a KFKI sokszorosító üzemében
Felelős vezető: Gonda Péter
Budapest, 1991. június hó