GANDALF - A Generally Applicable Numerical Data Acquisition Laboratory Facility

Lynov, Jens-Peter; Michelsen, Poul

Publication date:
1983

Document Version
Publisher's PDF, also known as Version of record

RISØ-M-2393

GANDALF - A GENERALLY APPLICABLE NUMERICAL DATA ACQUISITION LABORATORY FACILITY

J.P. Lynov and P. Michelsen
Association Euratom - Risø National Laboratory

Abstract. This report contains a description of an interactive computer program, GANDALF, for general purpose data handling. GANDALF is designed for different types of data input, it has flexible data processing possibilities, and it can produce various kinds of data output. The program is written for an HP-9825 desk-top computer in HPL-language. However, the fundamental program structure can also be applied to other computer systems with other computer languages.

INIS descriptors: DATA ACQUISITION; DATA PROCESSING; G CODES; NUMERICAL DATA

UDC 681.3.06

January 1983
Risø National Laboratory, DK 4000 Roskilde, Denmark

ISBN 87-550-0953-0
ISSN 0418-4635

Risø Repro 1983

CONTENTS

PREFACE

This report describes a general data acquisition and processing program for use in laboratories where the specific need for computerized measurements can vary considerably from time to time. Although still high, the speed of the program has been somewhat sacrificed in order to make a flexible program which, hopefully, is easy to use.

The program was developed during the years 1978 - 83 and is originally implemented on a Hewlett-Packard desk-top computer in HPL-language. However, it is our belief that the fundamental program structure described on the following pages can be transferred with success to other computer systems with other computer languages.

"Do not meddle in the affairs of wizards for they are subtle and quick to anger". Gildor Inglorion about Gandalf from The Lord of Rings by J.R.R. Tolkien.

1. INTRODUCTION

In this report a presentation will be given of the computer program GANDALF (short for Generally Applicable Numerical Data Acquisition Laboratory Facility), which has been implemented at the Q-machine laboratory at Risø on an HP-9825 computer. GANDALF is designed for different types of data acquisition, e.g. for A/D numerical input, it has flexible data processing functions and it can produce various kinds of data output on an HP-9872A four-colour-plotter, in addition to lineprinter listing on an HP-9866B, storage on floppy disks by an HP-9885 M/S and D/A output by an HP-59303A.

The presentation is organized in the following way: in Sec. 2 the terminology used throughout the report is introduced, Sec. 3 and Sec. 4 briefly describe the input and output facilities, respectively, while the data processing possibilities are presented in Sec. 5. After these introductory sections a more detailed description of GANDALF's capabilities and guidelines for extending the basic set of functions by user defined sub-functions is given in section 6.

It should be stressed that GANDALF is an interactive program with many facilities, so it is almost impossible to write a complete user guide to the program, since the actual sequence of questions asked by GANDALF is dependent on the user's answers to all previous questions. However, the very fact that GANDALF is interactive makes such a user guide almost superfluous since the program itself (in principle, at least) asks the right questions or gives the right commands at the right moments. Naturally, a certain amount of basic information is needed in order to understand GANDALF's questions. Also, a superficial knowledge of the widespread capabilities of the system is necessary in order to fully enjoy GANDALF's powerful wisdom. It is the idea with Secs. 2-5 to give the reader this fundamental information in a comprehensive form. By use of Sec. 6 and the appendices the interested and intelligent reader should also find it possible to go into the finer details of GANDALF's anatomy.

2. TERMINOLOGY

In the following a definition of some of the special terms used in this report is listed.

Variable. Different sources of input data give rise to different variables in GANDALF. In this way a probe position, a probe potential, and a sampling time may give rise to three variables: POSITION, POTENTIAL and TIME, say. The maximum number of variables is 10 and each variable can be given a name consisting of max. 16 letters. In general there is no formal distinction betweeen dependent and independent variables. Further details of the possibilities for the use of different types of variables (e.g. manual and A/D input) are discussed in Sec. 3.

Observation. A set of simultaneous values of the variables is called an observation (e.g. POTENTIAL at a certain TIME and a certain POSITION).

Investigation. The total set of observations describing a particular measurement (or numerical analysis) is called an investigation. The maximum number of observations multiplied with the number of variables which an investigation can contain is approx. 6000.

Another way to explain the meaning of the three terms just introduced is: If K variables ($1 < K < 10$) describe the unit vectors of a K-dimensional space, then an observation is a point in this space and an investigation is a (discretized) curve.

Parameter. Characteristic numerical data which are common for all the observations in an investigation are called parameters. Examples of parameters are: B-FIELD and CATHODE POWER. Each parameter can be given a name consisting of max. 16 letters, and the maximum allowed number of parameters is 10. (Under certain circumstances the maximum number of parameters is 7 or 8, for details see Sec. 3.1).

Job. The collection of one or more investigations, each with their own parameter values is called a job. One job can be to measure a dispersion curve while another can be determination of the two-dimensional structure of an electrostatic double-layer. A job can be given a name consisting of max. 16 letters (e.g. SISYPHUS).

Namefile. Information common to all the investigations in a job is stored in a namefile. The namefile contains the job name, the number of parameters, and their names, the number of variables and their names, and other information useful for GANDALF. A total list of the data stored in a namefile is contained in App. I. After the namefile has been produced, it is stored on a floppy disk. The disk filename of the namefile is automatically provided by GANDALF as "NFnn", where nn designates the namefile number with a leading zero, if necessary, so that a total of two digits is obtained. When the user is asked by GANDALF to refer to a previously stored namefile, only the namefile number without any leading zero is necessary.

Datafile. After conclusion of an investigation the parameter values and the values of the variables in all the observations, together with some extra informative data can be stored in a datafile on a floppy disk. A total list of the data stored in a datafile is contained in App. II. The disk filename of the datafile is automatically provided by GANDALF as "Dnndd" where dd is the datafile number with leading zeros and nn is the corresponding namefile number. In general only the datafile number without leading zeros is needed when referring to a datafile, except in cases where GANDALF specially asks for five digits, then nnddd is needed.

Single or block data processing refer to the cases of action after each observation or after a whole investigation, respectively. Examples are SINGLE PRINT or BLOCK PRINT for further details on these facilities see Sec. 4.

3. INPUT

3.1 Variable types

When discussing the input possibilities to GANDALF it is useful to list what is called the variable types and their respective numbers, T.

Type No. T	Variable type
1	Manual input
2	Computed variable
3	Digitized X-position
4	Digitized Y-position
5	File input
6	File transformation
7	Scanning of files
8	D/A type
9	Time variable
10	A/D input

Manual input is simply data entered into the machine via the keyboard.

Computed variable makes use of a user defined subfunction "SUBFnn" (where nn is the namefile number) in order to calculate the value of the variable X[I]. This value can be a function of all the different variables, including X[I]. The simple case when X[I] is a function of the instantaneous values of the variables, i.e. the values of the variables at the present observation number, can easily be programmed. However, the more special cases when X[I] is a function of the past (or even the future!) observations can also be handled. A detailed description of how to make a subfunction "SUBFnn" is contained in Sec. 6.3. If the first variable, X[1], in a job is a computed variable, this variable is considered as an independent variable, and all the other variables to be functions of X[1]. In this case GANDALF automatically includes three extra parameters in the end of the parameter list, meaning that the user in this case can insert 7 free parameters only. The three additional parameters are X1, X2 and DX, designating the first value, the last value, and the step length, respectively for X[1]. X1 may be larger than X2, but then DX must be negative.

Digitized X-position. The HP-9872A plotter can be operated in a digitize mode, which provides the possibility of reading the current location of the digitizer (a special "looking glass" which can be mounted in the pen holder). The digitized X-position is obviously the abcisse value of the digitizer in the current scale. When a variable is digitized, three additional parameters are included as in the case mentioned previously of a computed first variable. If the user wants to digitize a curve with a fixed X-step the digitizer will automatically be moved from X1 to X2 by steps of DX.

Digitized Y-position. If a variable has variable type 3 (Digitized X-position) the next variable must, naturally, be a digitized Y-position with variable type 4.

File input designates data fetched from one or more datafiles already stored on the disk. There are three different ways to fetch the desired data, and among other possibilities the user can use file input variables to either expand or contract the number of variables in each observation, or to scan through a series of investigations to find a dependent variable as a function of a parameter for a fixed value of another variable. Further description of file input is given in 6.2.8.

D/A-type. Actually, a D/A (Digital/Analog) variable is not an input variable, but is included here to complete the list of variable types. A D/A variable determines the output voltage from the D/A converter and if the first variable is of this type the three additional parameters, X1, X2, DX are included to control the output voltage in steps of DX from X1 to X2. If the D/A variable is not the first a user defined subfunction is called (see 6.3.2) in which the user must specify what the D/A converter has to output.

Time variable. The actual time in seconds after starting a new investigation is obtained from the internal clock and put into the time variable. If the first variable is of this type two parameters, "NUMBERS OF OBS." and "TIME INTERVAL" is set up giving a maximum number of 8 for the other parameters. A new observation is then entered every "TIME INTERVAL" until the specified "NUMBER OF OBS." is obtained.

A/D variables are used for input from the A/D converter in units of volts. If the first variable is of this type the usual three parameters, X1, X2, and DX are introduced. When an investigation is initiated the computer reads in observations starting at X1 and continuing to X2 by steps of DX, if possible (see 6.2.5).

3.2 Variable constants.

If the variable is of the type 1,3,4 or 10 the input value is multiplied by a variable constant before the value is stored in the computer. For further details on the point of variable constants as well as on the descriptions of the variable types the reader is referred to Sec. 6.1.5.

3.3 Auto mode.

When GANDALF is in the auto-mode a specified parameter, P, is scanned from P1 to P2 in steps of DP, and for each P-value an investigation is stored in a data file when completed. It is obvious that the auto-mode only has a sensible application when the first variable is of the type 2,8,9 or 10 so that each investigation is automatically performed. The auto-mode can be set on or off in the beginning of each new investigation when GANDALF asks to change the auto-mode. A more detailed description of the auto-mode is contained in Sec. 6.2.

4. OUTPUT

The output facilities offered by GANDALF are, naturally, dictated by the hardware connected to the system. At present, this hardware consists of: 1) a disk drive, 2) a four colour plotter, 3) a line printer, 4) an internal printer, 5) a D/A converter, and 6) a programmable switch function in the A/D converter. The different output possibilities are briefly described in the following.

4.1. Disk storage.

The storage system consists of two floppy disk drives, a master drive and a slave drive. To run GANDALF, a floppy disk containing the GANDALF PROGRAM, called GANDALF disk or main disk, must be placed in the master drive (drive Ø) and a data disk must be placed in the slave drive (drive 1).

As mentioned in Sec. 2, GANDALF is capable of storing two different types of files on a floppy disk, i.e. namefiles and datafiles. The namefiles contain information common to all the investigations in a job (a complete description of this information is given in App. I). Information about the storage of the namefile is given to the user immediately after the creation of a new namefile, for further details, see Sec. 6.1.5. The namefiles are stored on the same disk as GANDALF itself. The data, however, are stored on a data disk. A datafile contains all the data belonging to an individual investigation and the storage can be executed by the key-function BLOCK → DISK any time the text "SELECT KEY" is displayed. A table of contents of the information stored in the namefiles and their respective datafiles is provided through the key-function LIST as described in section 6.7.3.

4.2. Plotting

4.2.1. Two dimensional plots

GANDALF can plot one or more variables as ordinates with any other variable as the abcisse. The ordinates, as well as the abcisses, can be shown in either a linear or a logarithmic representation. The data points can either be marked by a user defined letter or symbol, or be connected by straight lines with the possibility of using seven different linetypes.

The process of plotting can either be executed during the data sampling in an investigation or after a whole investigation is completed. In the first case the user must, naturally, set up the plot scale before the data sampling is started (the user is automatically guided to perform this scaling). The insertion of new data points on the plot is performed immediately after each observation and this way of plotting the data is called the "SINGLE PLOT" mode, which the user can set on or off at the beginning of a new investigation. The other alternative of plotting the data after completion of an investigation is provided through the key-function "BLOCK PLOT". A scaling of the plot which allows the representation of all the data points is automatically suggested by BLOCK PLOT, but the user is free to choose his own scaling. Both the SINGLE PLOT and the BLOCK PLOT functions allow for additional information such as the parameter names and actual values to be written on the plot. Figure 1 shows an example of what a two dimensional plot may look like, and further information on the SINGLE PLOT and BLOCK PLOT functions are given in section 6.6.2 and 6.6.1, respectively.

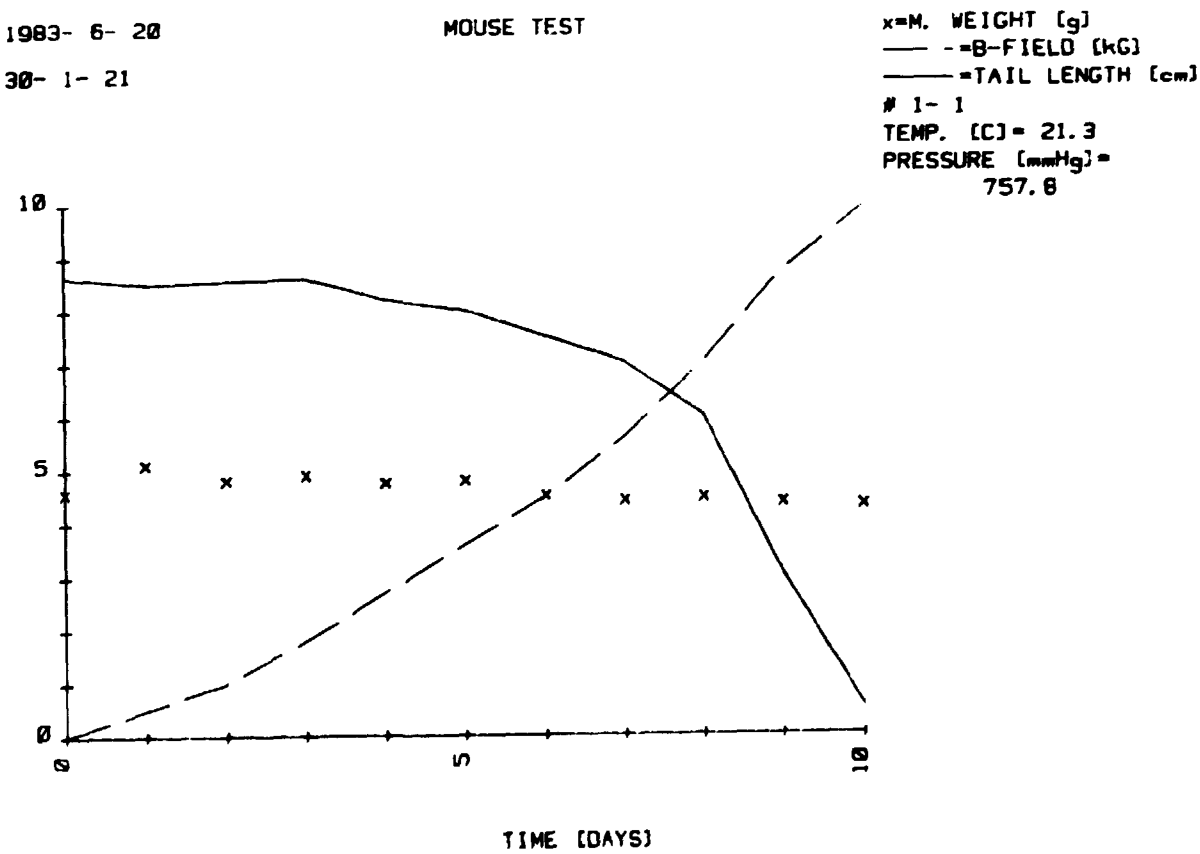


Fig. 1. Example of a two-dimensional plot by SINGLE PLOT or BLOCK PLOT.

4.2.2. "Three dimensional" plots.

GANDALF contains various sub-programs which can plot dependent variable as the function of two independent variables in different "three dimensional" representations.

3-DIM-PLOT. The simplest representation is provided by the 3-DIM-PLOT function illustrated in figure 2. Here a number of investigations which are saved as data files, are plotted as a sequence of ordinary two-dimensional curves, like those described in 4.2.1 with x and y offsets determined by one of the parameters. The 3-DIM-PLOT offers no removal of hidden lines, but it is the most flexible of the three dimensional plotting functions since it does not require equidistant data points. For further discussion see 6.6.3.1.

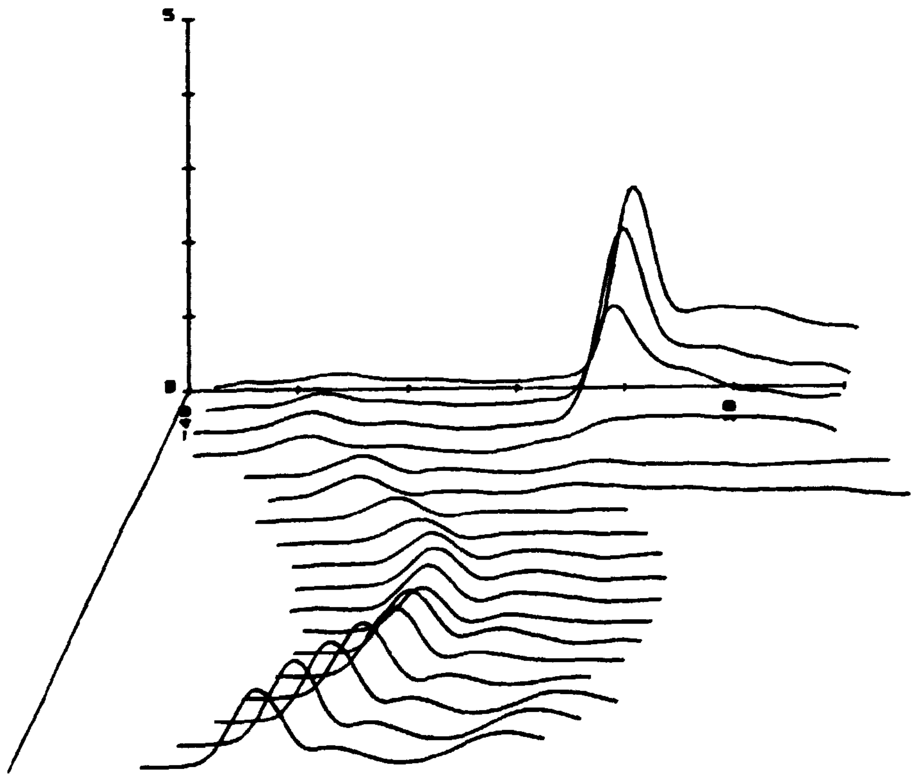
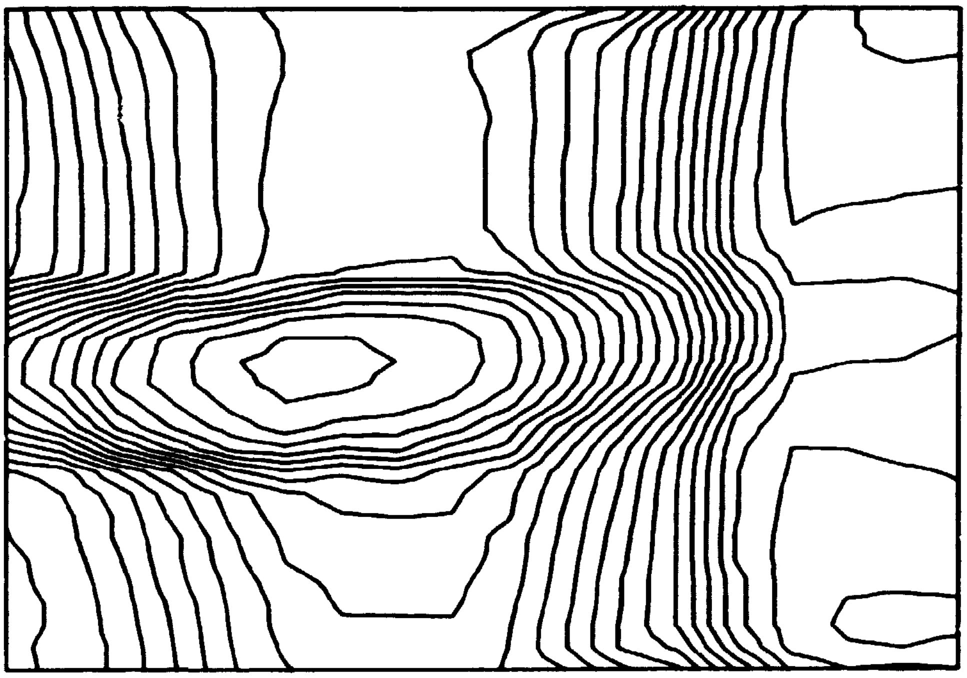


Fig. 2. Example of a 3-DIM-PLOT.

a)



b)

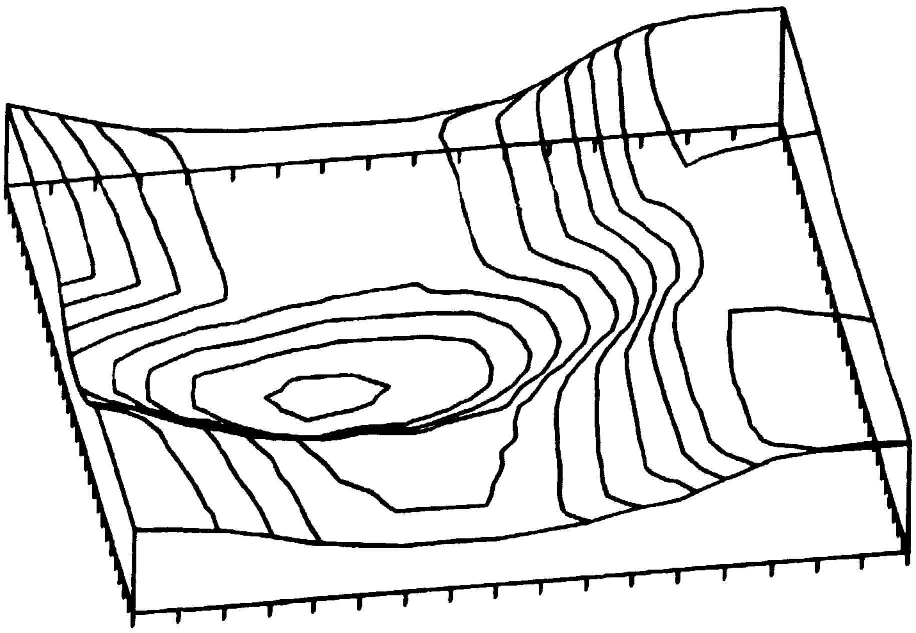


Fig. 3. Examples of CONTOURPLOT. a) Two-dimensional presentation and b) three-dimensional presentation.

CONTOURPLOT. The two different types of plots offered by CONTOURPLOT are illustrated in Figs. 3a and b. The input data to CONTOURPLOT are taken from a number of data files described by the user, covering a range of observations within user defined boundaries. The data are assumed to be equidistant, i.e. the distance between the independent observation variable values are the same as well as the difference in the values of the independent file parameter. Figure 3a shows the contour lines of the three dimensional landscape seen from above as in ordinary geographical maps, while Fig. 3b shows the same lines seen "sideways" with a user defined tilt and rotation angle. In the last type of plot hidden lines are not removed. The direction of the axis of each of the independent variables may be inverted. For further details see €.6.3.2.

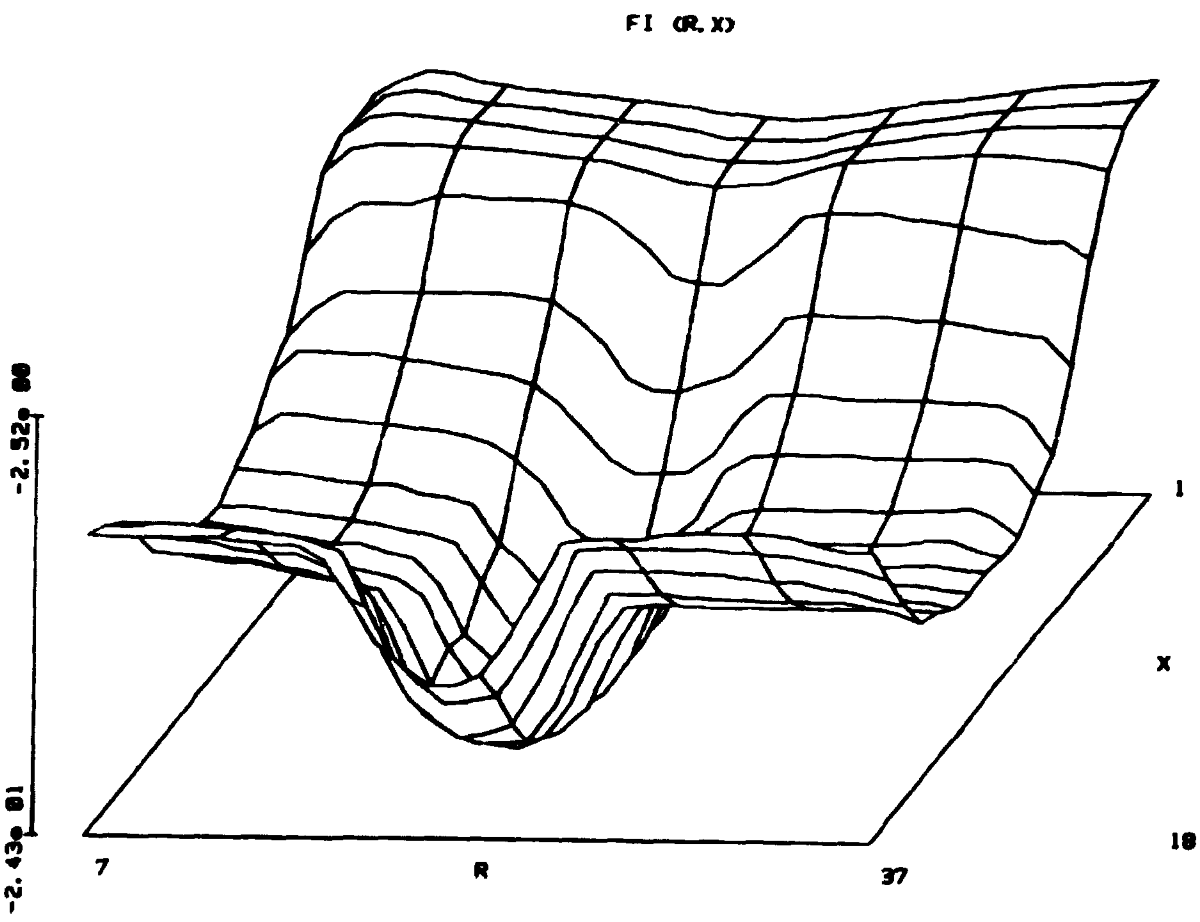


Fig. 4. Example of a NETSURFACE plot of the same data as shown in Fig. 3.

NETSURFACE. NETSURFACE plots the projection of an axi-parallel rectangular net on the three dimensional surface determined by the dependent variable. The subprogram give the possibility of removing hidden lines and an example of a NETSURFACE plot is shown in Fig. 4.

The data can either be taken from previously stored data files, as in CONTOURPLOT, or be generated by a user defined arithmetic function which may be stored in a separate disk file for later reference. A thorough description of NETSURFACE is given in 6.6.3.3.

4.3. Listing

SINGLE PRINT. As in the case of SINGLE PLOT, which was described in 4.2.1 the user has a possibility of monitoring the data sampling during a investigation by using the SINGLE PRINT mode. In this mode the variables are listed on the internal printer on the 9825 and an example of the output is shown in Fig. 5. The SINGLE PRINT mode can be set on or off in the beginning of each new investigation.

```
   1983-7-6
================
   MOUSE TEST
================

MAIN DISK#:    30
NAMEFILE#:     21
INV.#:          1

TEMP. [C]=
       2.130E 01
PRESSURE [mmHg]=
       7.578E 02

X1=TIME [DAYS]
X2=M. WEIGHT [g]
X3=B-FIELD [kG]
X4=TAIL LENGTH [

OBS#1
X1=0.000E 00
X2=4.750E 00
X3=0.000E 00
X4=6.750E 00

OBS#2
X1=1.000E 00
X2=5.120E 00
X3=4.530E-01
X4=8.720E 00

OBS#3
X1=2.000E 00
X2=4.987E 00
X3=1.012E 00
X4=8.806E 00
```

Fig. 5. Example of SINGLE PRINT output.

BLOCK PRINT. BLOCK PRINT creates a complete list on the line-printer of all the data in an investigation. An example is shown in Fig. 6. The output is segmented to be presented on A4 size paper. If the user wants to remove an unfilled page from the printer, the key function PAGE OUT will insert the correct number of blank lines to ensure the A4 format.

```
30- 1-21- 0                    MOUSE TEST                    1983-7-6
=========================================================================

INV#: 1
      TEMP. [C] =  2.130E 01          PRESSURE [mmHg] =  7.578E 02

-------------------------------------------------------------------------
```

OBS.#	TIME [DAYS]	M. WEIGHT [g]	B-FIELD [kG]	TAIL LENGTH [cm]
1	0.000E 00	4.750E 00	0.000E 00	8.750E 00
2	1.000E 00	5.120E 00	4.530E-01	8.720E 00
3	2.000E 00	4.987E 00	1.012E 00	8.806E 00
4	3.000E 00	5.122E 00	1.987E 00	9.012E 00
5	4.000E 00	4.852E 00	3.052E 00	8.001E 00
6	5.000E 00	4.753E 00	3.564E 00	8.895E 00
7	6.000E 00	4.568E 00	4.781E 00	7.563E 00

Fig. 6. Example of BLOCK PRINT output.

4.4. Analog output

D/A-OUTPUT. The D/A-output possibility has been mentioned previously in Sec. 3 (under the title "Input"!, but it was included for completeness of the variable type description). It was explained that if the first variable in each observation is of the D/A type, GANDALF automatically controls the D/A output from X1 to X2 in steps of DX, where X1,X2, and DX are the last three parameters. On the other hand, if the D/A variable is not the first then the user must include a subfunction "SUBFnn" (see 6.3.2) in which the D/A control must be programmed.

A voltage of V volts, where $-9.99 < V < 9.99$, is set up on the D/A output terminals by the programline

```
fmt f.2; wrt 7ø8, V.
```

Note that the maximum current load of the D/A converter is 10 mA corresponding to a minimum load resistance of 1kΩ. Any time the display shows "SELECT KEY", a D/A output can be obtained by pressing the D/A-key.

Switch control. The control of the programmable switch function is not included in the GANDALF program, but as usual the user is free to include his own subfunction "SUBFnn" (see 6.3) to take care of the switch function. The switch output of the A/D converter is located on the rear end of the 59313A and denoted "RVS CHAN" (which is short for the Reverse Channel function, see Ref. 1).

The switch is opened by the command

```
wrt 7ø6, "O"
```

and closed by

```
wrt 7ø6, "N".
```

The controlled voltage must be less than 80 V (for the correct polarity, check the Ref. 1) and the maximum current sink is 200 mA.

5. DATA PROCESSING

The general strategy which GANDALF is intended to follow is to enable the user to make simple data processing in a, hopefully, easy manner and, at the same time, to allow for complex data processing schemes limited only by the user's fantasy (and skill!).

The simple processing includes features like 1) performing $C + X$, $C * X$, and $X \uparrow C$, where C is a constant and X is a certain variable, in all the observations in an investigation, 2) adding a weighted calibration investigation, i.e. a whole curve, to another investigation (called a two-file operation), and 3) performing regression analysis of two of the variables in an investigation by using one of 8 standard models (including Trivelpiece-Gould dispersion, and Langmuir probe characteristics).

If the simple processing functions are not sufficient the user has the possibility of performing 1) arbitrary variable transformations during an investigation (if one of the variables is a computed variable), 2) arbitrary variable transformations after an investigation is completed (regardless of the variable types), and 3) arbitrary regression analysis with up to 5 fitted parameters.

In the following subsections the various data processing possibilities are discussed in further detail, while a more complete description of some of the advanced possibilities of the data processing in GANDALF are found in Sec. 6.

5.1 Simple block transformation

As mentioned in Sec. 2 a Block (of data) is synonymous to a complete investigation. The Block of data can either be newly generated, non-stored data or previously stored data read into the computer in the beginning of a new job. By pressing the key labelled "Bl.-Trans." the user is first asked if he wants a simple transformation. Answering yes gives the possibility of performing

$$X_i \cdot C \rightarrow X_i \text{ , or}$$
$$X_i + C \rightarrow X_i \text{ , or}$$
$$X_i \uparrow C \rightarrow X_i$$

where X_i denotes the variable with variable number i and C is a constant.

If the user does not want a simple transformation, he is given the opportunity to perform the two-file operation

$$X_i + C \cdot Y_i \rightarrow X_i ,$$

where Y_i denotes the variable with variable number i in an investigation already stored on a disk.

If the user neither wants the simple transformation nor the standard two-file operation, he must specify what he desires in a subfunction SUBFnn, to be described in the following subsection.

5.2. User defined subfunctions "SUBFnn"

The SUBFnn subfunctions where nn designates the namefile number with a leading zero must be created and stored on the GANDALF disk under the following circumstances 1) if one of the variables is a computed variable, 2) if a variable, which is not the first in an observation, is of the D/A type, 3) if the variables are of the file transformation type or file scanning type with variable type number 6 or 7 (see 6.2.8), 4) if a non-standard Block Transformation is desired.

The user defined subfunctions must start with the label "SUBF": in the first programme line and end with the ret statement. The parameter values are available as array variables P[i], i = 1,2 ... 10, and the variable values within the current observation number, I, are present as array variables X[J] , J = 1, 2 .... 10.Furthermore, the user is free to use to variable names from A to G and, naturally, the p-variables, the latter for references inside the subfunction only.

A typical example of a user defined subfunction is

```
0: "SUBF":
1: deg
2: cos(X[1])+X[2]
3: sin(X[1])+X[3]
4: ret
```

which either can be used during the investigation if the second and fourth variable are computed variables, or as a non-standard Block-Transformation.

It is easy to see that if all the variables are computed variables, it is a simple task to make quick calculations of purely arithmetic functions, since the first variable, X[1], automatically is given values from X1 to X2 in steps of DX, where X1, X2, and DX are the last three parameters. Function tables or plots are then simply performed by the Block Print or Block Plot key functions. A detailed description of the subfunction possibilities and restrictions is given in Sec. 6.3.

5.3 Regression analysis

The regression models in GANDALF, which will be available by pressing the key function "REG-AN", are

```
***REGRESSION***
***ANALYSIS ***

NO.   MODEL
0 NONE
1 Y=A*exp(B*X↑2)
2 Y=A*exp(B*X)
3 Y=A+B*ln(X)
4 Y=A*X↑B
5 Y↑N=A+B*X↑M
6 Y=A+B*X+C*X↑2
7 PROBE CHARACT.
8 DISP(fp,a,Vt)
9 GENEREL FUNC.
```

where X and Y are the two regression variables, A, B, and C are the parameters to be fitted, and N and M are constants. The user may choose to fit any number of the parameters in a given model keeping the rest of the parameters at a constant value.

Model 7 performs a regression analysis of a Langmuir probe characteristic where the user can define the voltage region, in which the best linear ion current is found, and the voltage region, in which the best exponentiel fit to the electron current (determined as the total current minus the linear ion current) is performed. As a result, the electron temperature, T_e, is printed.

Model 8 performs a regression analysis of the Trivelpiece-Gould dispersion relation

$$f^2 = f_p^2 \frac{(ka)^2}{1 + (ka)^2} + \frac{1}{(2\pi)^2} 3k^2 v_t^2,$$

where f and k are the measured frequency and wavenumber, a = plasma radius/2.4, and $v_t = T_e/m$ is the electron thermal velocity; T_e and m being the electron temperature and mass, respectively.

In all the regression models, the user has to guess starting values of the parameters to be fitted. These values must as a minimum have the correct sign, and in most cases the correct order of magnitude is useful.

The user is free to make his own regression models for fitting up to 5 parameters. The user model must be stored on the disk under the name "REGFnn" (where n is the namefile number with a leading zero) and this model is applied by using model number 9.

Further details on how to use the standard regression analysis models and how to construct more sophisticated models are given in Sec. 6.8.1.

5.4 Data interpolation

The subroutine "SPLINE", which is accessible in the beginning of a new job, uses cubic spline interpolation to compute a smooth curve $s(x)$ through n unequally spaced data points (x_i, y_i). SPLINE produces m equally spaced data points (t_j, s_j) where $s_j = s(t_j)$, in addition to the derivatives $s'(t_j)$ and the integral of $s(x)$ from x_1 to x_n.

The subroutine SPLINE includes the HP "General Utility Routine" called "DERINT" and described in Reference 2. Further details on the GANDALF subroutine SPLINE are given in Sec. 6.8.2.

6. HOW TO USE GANDALF

GANDALF is an interactive program with many branches. The user determines which branch he wants to follow either by using the key-functions or by answering the "branching questions" which GANDALF asks via the display. Some questions have an immediate effect without doing any branching in the program. This section describes the effect of the various key-functions, of the "branching questions" and of some of the essential "non-branching" questions where additional explanation is necessary. Many of the "non-branching" questions are self-explanatory so no extra discussion is needed.

The GANDALF program is divided into a number of program files, placed physically on a disk called the GANDALF disk, and only one or two of these files are in the machine memory at a given time. Each program file is divided into a number of program blocks, each of which has a label as a headline. A list of all the program files can be found in Appendix III. A program block will be referred to by its headline, e.g. "INVEST". The questions GANDALF asks are shown in brackets, e.g. [PEN # ?], while the allowed answers are shown as e.g. <Ø,1>, which means that the user has to type-in either a Ø or a 1 followed by a continue.

The key functions should only be used when the display shows: "SELECT KEY", and even in this case only relevant key-functions should be used. The keys are listed in Table 1, together with the number of the subsection where they are described.

When GANDALF asks a question, it can in most cases be answered with a number followed by a "continue", or in some cases simply by pressing "continue". A yes or no question should be answered with a <1> (= yes) or a <Ø> (= no). If a question shown on the display ends with e.g. (1 = YES) any other answer than <1> will give the same result as answering with <Ø>.

Table 1. This table contains all the key-functions and a reference to the subsection in which a description can be found.

Key function	Section number
BLOCK PLOT	4.2.1. and 6.6.1.
SCALE	6.6.4.
X-OFS	6.6.5.
A/D-DSP	6.2.5
TERMINATE	6.3.6.
BLOCK PRINT	4.3. and 6.7.1.
DATA-EXIT	6.2.2.
NO-DATA	6.2.2.
NEW INVEST	6.2
REG.AN.	5.3. and 6.8.1.
BLOCK TRANS.	5.1 and 6.4.
SAVE PLOT	6.6.7.
3D-PLOT	4.2.2 and 6.6.3.
Y-OFS	6.6.5.
SET D/A	6.2.6.
OPTION	6.3.5.
PAGE-OUT	6.7.1.
LIST	6.7.3.
CORRECTION	6.2.2.
NEW JOB	6.1.2.
CONSTANTS	3.2. and 6.1.4.2.
BLOCK→DISK	4.1. and 6.5.

To make the description of the questions precise, we will use the following semi-symbolic form:

X.X.X. [QUESTION] <cont.,1,2,3,4>
<cont.>: See Y.Y.Y.
<1,2,3,4>: Comment

The number X.X.X. is a subsection number which gives a reference to the question we are dealing with. The word QUESTION means the question GANDALF presents on the display. The numbers or letters in the <> brackets define the possible answers the user can give. and cont. means that just a continue can be typed-in. Below this first line some comments to the various possible answers follow. The number Y.Y.Y. is a subsection number, under which a description can be found of the next question GANDALF will ask you. If SELECT KEY is written, the meaning is that with this answer the next display will be SELECT KEY.

To describe the branching in the program some block diagrams are given in the following subsections. The numbers in the blocks refer to the subsection where the questions are described. Only questions of the branching type are shown in these block diagrams.

6.1. Start procedure

To start the GANDALF program, place the GANDALF disk in the master disk drive (drive Ø) and a corresponding data disk (see Sec. 6.9) in the slave disk drive (drive 1). Then type in:

<get "GANDAL", Ø, Ø>

exactly as written in the brackets and press EXECUTE. The file GANDAL is then transferred to the machine and is run from line Ø (for details see the block diagram shown in Fig. 7).

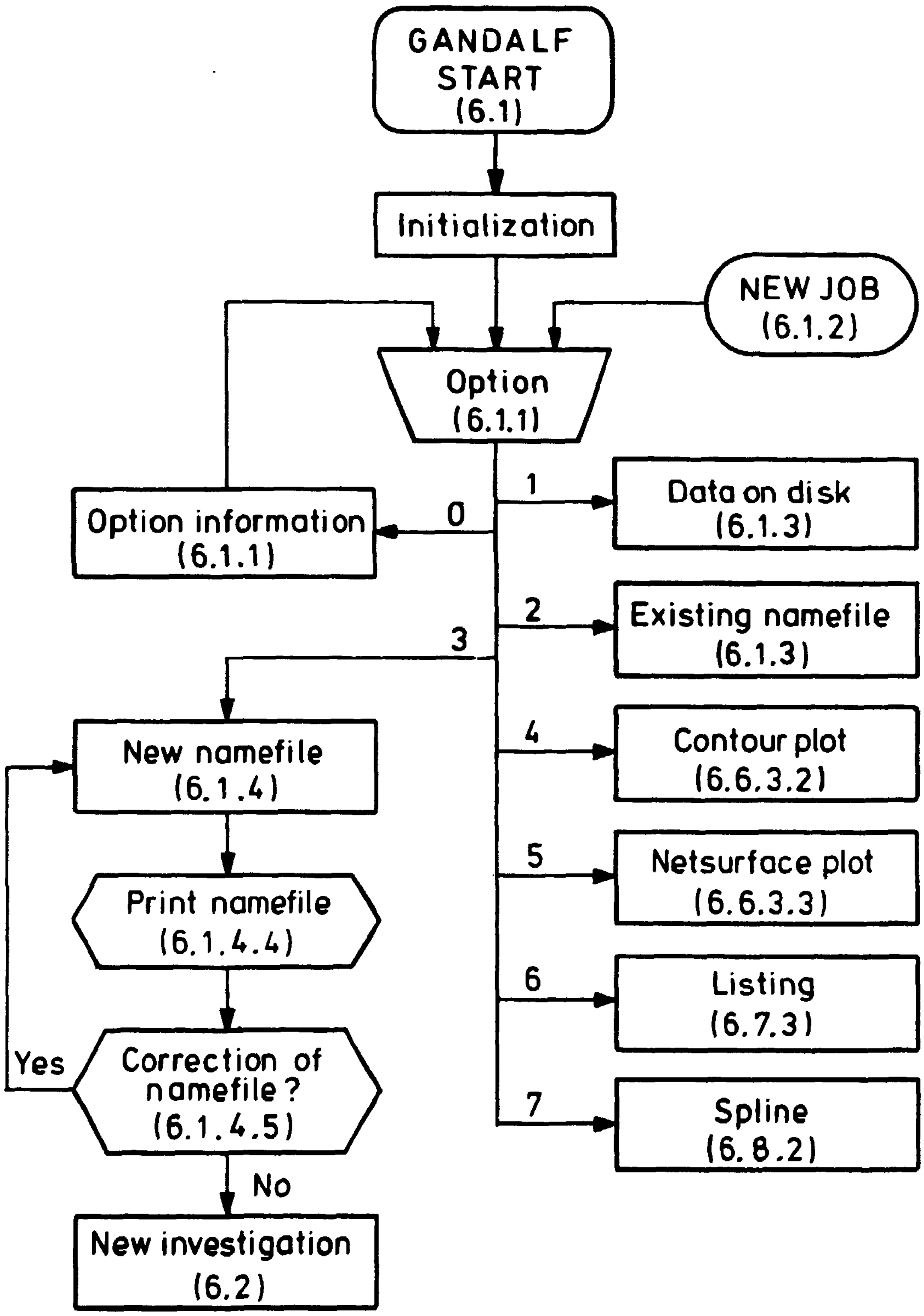


Fig. 7. Block diagram showing the general structure of GANDALF.

6.1.1 "OPTIONS".

After the initialization GANDALF prints out the following message:

```
     GANDALF
================
VERSION:       3
MAIN DISK#:   30
NAMEFILES:    21
DATA DISKS:    1
DATA DISK#:    1
    1983-7-6
================
```

The information given in this message is the following: a) VERSION states the current version of the GANDALF program, b) MAIN DISK# is the number of the GANDALF disk, c) NAMEFILES tells you how many namefiles are stored on the GANDALF disk, d) DATA DISKS indicates how many data disks that belongs to the GANDALF disk, e) DATA DISK# is the number of the current data disk in disk drive 1. Then GANDALF displays:

[CHOOSE OPTION, Ø=INFORMATION] <Ø,1,2,3,4,5,6,7 >

<Ø>: You get the following list with information about the other possibilities:

```
     OPTIONS
================
DATA ON DISK:   1
EXISTING
   NAMEFILE:    2
NEW NAMEFILE:   3
CONTOURPLOT:    4
NETSURFACE:     5
LISTING:        6
SPLINE          7
```

<1>: Enters data which are stored on the data disk into the machine (see 6.1.3).

<2>: Allows new data to be input with a namefile already on the GANDALF disk (see 6.1.3).

<3>: Initiates creation of a new namefile (see 6.1.4).

<4>: Performs a contour plot from a series of investigations (see 6.6.3.2).

<5>: Makes a netsurface plot from a series of investigations (see 6.6.3.3).

<6>: Allows listing of namefiles, datafiles and program files, (see 6.7.3).

<7>: Constructs a smooth curve through all points of a variable by use of a cubic spline function (see 6.8.2).

6.1.2. NEW JOB.

By using the key function NEW JOB all the options described in 6.1.1. can be obtained.

6.1.3. [NAME FILE # ?] <1,2, ... 99>. When you have typed-in the namefile number, the namefile is transferred to the machine. If you have chosen option 2, GANDALF will continue with 6.2, otherwise you are asked:
[DATA FILE #?] <1,2,...999>. The complete datafile name is nnddd, where nn is the namefile number and ddd is the datafile number. This datafile name is printed out when a data file (an investigation) is stored on the disk. When you have typed-in the datafile number you get the message: "SELECT KEY!"

6.1.4 NEW NAMEFILE

6.1.4.1. [NUMBER OF VARIABLES?] <1,2,...1Ø>. This is the first question when you make a new namefile. The following questions are self-explanatory.

[NUMBER OF PARAMETERS?] <ø,1,...,1ø>. In some cases the maximum parameter number is 7 or 8. See Sec. 3.1.
[NAME OF JOB?] <A string of no more than 16 symbols>
[NAME OF Nth PARAMETER?] <max. 16 symbols>
[NAME OF Nth VARIABLE?] <max. 16 symbols>
[VARIABLE TYPE?] <1,2,.....,1ø>

The various possibilities are listed on the printer if the user answers yes to the question [VARIABLE TYPE INFORMATION?]:

```
VARIABLE TYPES:

MANUEL INPUT:   1
COMPUTED VAR.:  2
DIGITIZED X:    3
DIGITIZED Y:    4
FILE INPUT:     5
FILE TRANSF.:   6
SCANNING:       7
D/A TYPE:       8
TIME VAR:       9
A/D INPUT:     10
```

<1>: Manual input means a variable that is typed-in from the key board.

<2>: Computed variables are variables which are computed in a subfuction made by the user (see 6.3). If the first variable is of this type three new parameters are added to the ones the user has already defined. These new parameters automatically get the names X1,X2, and DX. When an investigation is started the first variable will step from X1 to X2 by steps of DX.

<3>: Digitized X is a variable which gets the input from the abscisse value of a digitized point. Also in this case three new parameters are introduced as for computed variables. If the user wants a fixed X-step, the digitizer unit will step from X1 to X2 by steps of DX.

<4>: Digitized Y is a variable which gets the input from the ordinate value of a digitized point. If a variable is of type 3, the next one must be type 4.

<5>: File input is an input type where the data are read from a datafile already stored on the disk under another namefile. The purpose of the variable type is to allow for a change in the number of variables from an old datafile to a new one. For instance, it may be necessary to expand the number of variables in order to get space for some new variables calculated by a "BLOCK TRANS" (see 6.4), or it may be an advantage to decrease the number of variables to save space on the disk. (For further information see 6.2.8).

<6>: This is similar to type 5, except that a user defined subfunction is used, so a transformation of the variables may be performed after they have been read from the datafile on the disk. (Concerning subfunctions, see 6.3).

<7>: A scanning variable also takes data from datafiles already stored on disk, but as the name indicates the input taken during a scan through several datafiles. The data from each datafile can be transformed in a subfunction to new variables in one observation, i.e. each observation in the current investigation takes input from one datafile on the disk (see also 6.2.8). Two parameters are introduced in this case, FIRST FILE # and LAST FILE #, which specify the range of the data files.

<8>: If the first variable is of the D/A type, three new parameters X1,X2 and DX are introduced as in case <2>. When the investigation is started the D/A converter will give a voltage output starting at X1 and stepping up to (or down to, if DX<Ø) X2 in steps of DX. If another variable than the first one is of the D/A type a user defined subfunction is called (see 6.3). In this subfunction the user must specify what the D/A converter has to output.

<9>: The time variable reads the time from the internal clock. The time starts when an investigation is started and the unit is one second. If the first variable is of this type, two parameters, "NUMBER OF OBS" and "TIME INTERVAL", are set up. A new observation is then input every TIME INTERVAL until the specified NUMBER OF OBS is obtained.

<1Ø>: The variables of this type are input from the A/D converter directly in volts. If the first variable is of this type, the three parameters X1,X2 and DX are introduced. When an investigation is started the computer reads-in data starting at X1 and continues until X2 in steps of DX, if possible (see 6.2.5).

6.1.4.2 [VARIABLE CONSTANT?] <Any number the machine can accept>
If the input variable is of the type 1,3,4 or 1Ø the input value is multiplied by this constant to give the final input. For other variable types the constants are not used, but the user may utilize them in subroutines. The constants may be changed by the user by pressing the key-function: CONSTANTS. The constants which are input during the process of making a namefile will be stored on the disk in the namefile. This means that these constants are available next time the namefile is fetched from the disk, even if the key-function CONSTANTS has been used last time this job was performed.

6.1.4.3 [DO YOU INTEND TO PLOT?] <Ø,1>

<Ø>: see 6.1.2.6.

<1>: In this case information about which variable should be the independent variable and which variable(s) should be the dependent variable(s) in a plot later on is stored in the namefile. However, it is possible to change the dependent and independent variables when a plot is produced.

[X-VAR ?] <cont., 1,2, ...K>. K is the number of variables used in this job.

<cont.>: see 6.1.2.6

<N>: N determines which variable is going to be plotted in the x-direction. If N is negative, a logarithmetic x-axis is assumed.

[Nth.Y-VAR ?] <cont. 1,2, ...K>.

<cont.>: See 6.1.2.6.

<N>: N determines which variable is going to be plotted in the y-direction. If N is negative, a logarithmetic y-axis is assumed.

6.1.4.4 [PRINT NAMEFILE?] <cont. Ø,1>

<cont.>=<Ø>: See next question.

<1> All data in the namefile is printed out in the following way:

```
   1983-7-6
================
DISK: 30 NF: 21
================
   MOUSE TEST
================
PARAMETERS:
TEMP. [C]
PRESSURE [mmHg]
VARIABLES:
TIME [DAYS]
M. WEIGHT [g]
B-FIELD [kG]
TAIL LENGTH [cm]
 I   TYPE   PLOT
================
 1       1      1
 2       1      2
 3       1      3
 4       1      4
================

 I    CONSTANTS:
================
 1     1.0000E 00
 2     1.0000E 00
 3     1.0000E 00
 4     1.0000E 00
```

If a variable is of the A/D type, the type number has been changed since the final type number is = 1Ø+channel number.

6.1.4.5 [CORRECTION OF NAMEFILE?] <cont.,Ø,1>

<cont.>=<Ø>: GANDALF is now ready for data input (see 6.2).

<1>: The program starts again with question 6.1.4.1 and a new namefile can be produced by answering the questions from 6.1.4.1 to 6.1.4.3. Only the answers which have to be corrected has to be typed-in. If no correction is necessary, just press continue.

6.2 NEW INVEST

A new investigation starts automatically when a namefile has been produced or after entering a namefile from the disk by the EXISTING NAMEFILE option, or when the key-function NEW INVEST is pressed. All data input is performed by the program file INPUT. The general input scheme is shown in the block diagram in Fig. 8. An investigation consists of some parameters and some observations. There are two kind of parameters. The first kind is calculated or measured by GANDALF and stored in the r-variables rØ-r9 (see Appendix III). The other kind is entered via the keyboard in the beginning of each investigation, and are the ones defined by the user in the namefile. These parameters are stored in a P [*] array and also as r1Ø - r(P-1) where P is the total number of user defined parameters.

The I'th observation consists of the variable-values $X_{i,j}$, which are input according to the variable type defined in the namefile. These variable values are at first put into the array X[J], but are transferred to r-variables according to: X[J] → r(9+P+(I-1)K +J) after each observation is completed.

In the "AUTO-MODE" many investigations can be performed automatically, without communication between GANDALF and the user. In this mode one of the user defined parameters is stepped from a start value P1 to a stop value P2 by a DP step. The other parameters are fixed. After each investigation the data are stored on the disk, and a new inverstigation starts. In the following paragraph the questions, which GANDALF asks in the beginning of an investigation, are discussed. In the next paragraphs questions and problems connected to the specific variable types will be mentioned.

6.2.1. General input scheme

Several of the questions asked during the start of an investigation are only asked under certain circumstances. In case of auto-mode operation the questions are only asked in the beginning of the first investigation, while the following are performed automatically.

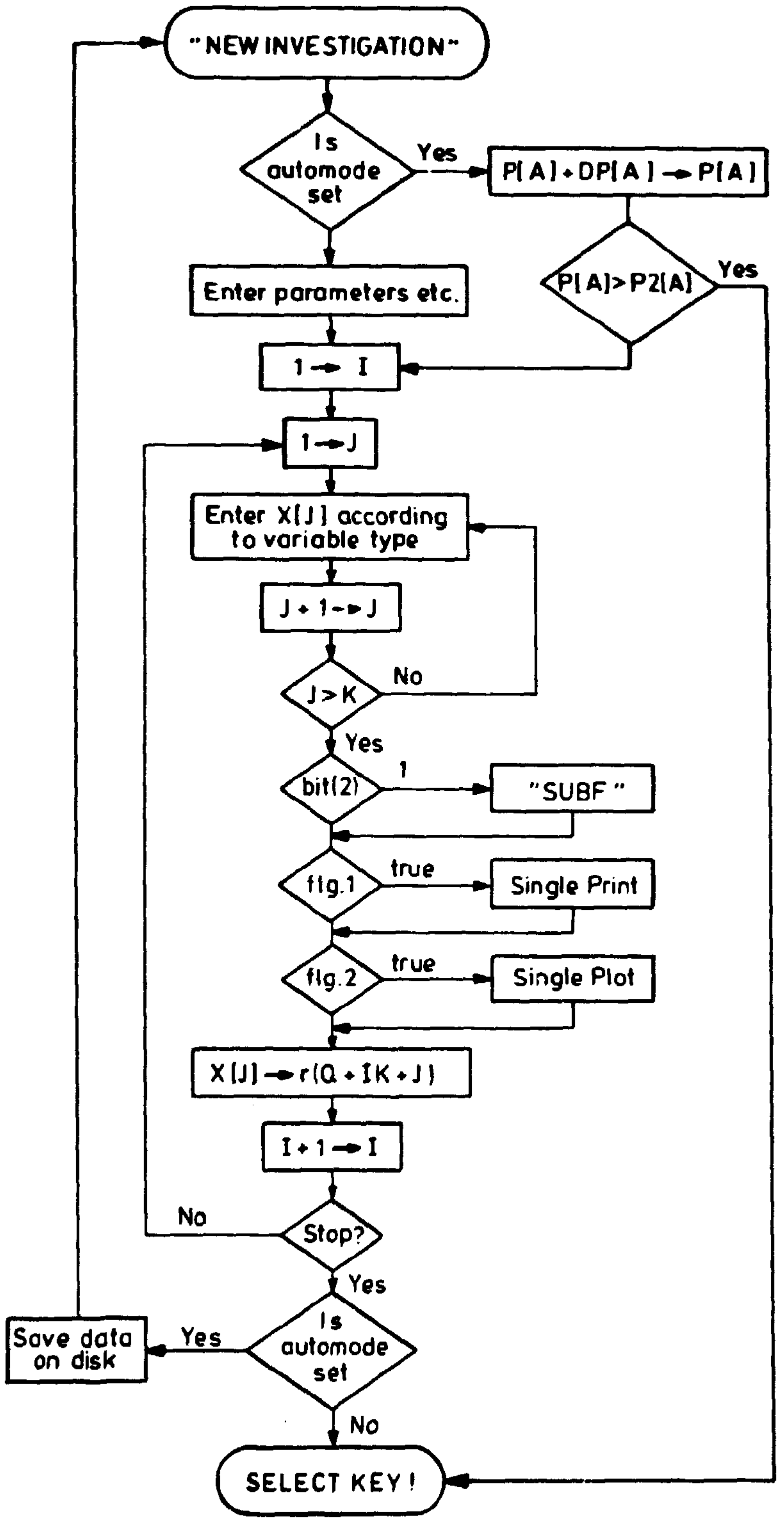


Fig. 8. Block diagram showing the general input scheme of GANDALF.

SAVE DATA ON DISK? <Ø,1>

This question is only asked if data already have been read but not saved.

INVESTIGATION#? <cont., any number>

<cont.>: In case this is the first investigation the INV.# will be 1, else it will be the last INV.# + 1. The INV.# is stored in r4.

"PARAMETER NAME?" <parameter value, cont.>

<parameter value>: The appropriate parameter value corresponding to the PARAMETER NAME shown on the display is typed-in.

<cont.>: In this case the parameter value used in the last investigation is assumed.

This question is repeated for each of the parameters.

CHANGE PRINT, PLOT OR AUTO-MODE <cont.,Ø,1>

<cont.> = <Ø>: go to 6.2.1.2.

SINGLE PRINT ON? <Ø,1>

<1>: In this case all observations will be printed out on the internal printer during the investigation.

SINGLE PLOT ON? <Ø,1>

<1>: The observations will be plotted during the investigation according to the plottype. You will be asked if you want to plot axes or text. (If this is desired, see 6.6 for further explanation).

AUTO MODE? <Ø,1>

<1>: Sets the auto-mode and you will be asked which parameter you want to step, the last value of this parameter, and the step size you want to use.

6.2.1.2. "START". At this point the input of observations starts according to the variable type. The special questions and responses for the various variable types are treated in the following.

6.2.2. Manual input

When the variable is of this type, data are enterend via the keyboard according to the information given in the display. If you want to skip a point, press the key-function NO-DATA. The effect is that the number 1e99 is entered. This particular variable value will be skipped if data are plotted or a regression analysis is performed. When all the data have been typed-in, press the key-function DATA-EXIT. If some data have to be corrected, use the key-function CORRECTION.

6.2.3 Computed input

In this case a subfunction will be called after each observation. See section 6.3 for how to make a subfunction.

6.2.4 Digitized input

Data are digitized using the plotter-digitizer function. When two identically points are digitized, the input stops. Afterwards the correction key may be used. In the beginning of each investigation you are asked if you want to define a new coordinate system by the question: [NEW AXES?] <Ø,1>.

6.2.5 AD input

Variables of the AD-type data are read from the analog-digital converter. If the first variable is of this type two conditions must be satisfied before the input starts. The user must press CONTINUE when GANDALF displays "PRESS CONTINUE TO START!". The input (signal) must be below X1 if DX>Ø and above X1 if DX<Ø. When this has been satisfied the input starts when the signal passes X1, and stops either when the signal passes X2 or when the key-function TERMINATE is used.

The key-function AD-DISPLAY may be used when "SELECT KEY!" is on the display to show the direct output from the four channels of the A/D-converter. This mode is stopped with the key-function TERMINATE.

6.2.6 DA-variable

If the first variable is of D/A-type, this variable will start with the value X1 and then step by DX until X2. The output voltage in Volts from the D/A converter will be equal to the current value of this variable. If a variable different from the first one is of this type, a subfunction "SUBF" will be called for each observation. The user must define what the DA-converter shall output in this subfunction. See section 6.3. The D/A-converter can be set manually by using the key-function SET D/A when the "SELECT KEY" is on the display.

6.2.7 Time variable

If the first variable is of this type, two parameters have been defined when the namefile was produced: NUMBER OF OBS. and TIME INTERVAL[s]. The values of these two parameters determine the total number of observations and the time interval in seconds between two observations, respectively. If another variable than the first is of this type, the value of the variable will be read from the internal clock in units of seconds. The clock is started when the user presses CONTINUE after the display shows "PRESS CONTINUE TO START!". The TIME INTERVAL must be large enough to allow for the various actions evolved with the rest of variables.

6.2.8 File input

If one of the variables is of type 5,6 or 7, all the variables must be of the same type. These variable types are used if one already has some data stored on disk files but wants to connect them to another namefile. The file input can be used in three ways as mentioned in section 6.1.5.1. If you, for example, have measured some data X_i,Y_i and stored them on the disk, and you now want to calculate a function Z(X,Y) you may create a new namefile with the three variables X,Y,Z. Then, when you start an investigation, GANDALF will read the data from a specified data file and place the values of X_i and Y_i in the r-variables, but for each dataset X_i, Y_i a r-variable is reserved for the value of the Z-function. The Z-function can then be calculated by using a block transformation (see section 6.4)

When making the namefile you are asked which way you want the file input. The three possible file input types are called:

"FILE INPUT" (type 5)
"FILE TRANSFORMATION" (type 6)
"SCANNING OF FILES" (type 7).

The example above is a case of a file input operation. The second type is similar but a transformation of variables is performed during the input in the same way as when the variable is of the computed type, i.e. for each observation a subfunction is called and the variable transformation defined by the user will be performed. The method of producing a subfunction is described in section 6.3.

The "SCANNING" is also performed by using a user defined subfunction. The idea is that if you have performed a series of investigations i.e. with one parameter varying for each investigation, you can produce a new investigation where the parameter is one of the variables and one or more of the variables from a fixed observation number are variables in the new investigation (see examples in sec. 6.3.4).

6.3. How to make subfunctions

When one of the variables other than the first is of the computed or the D/A type, or if all the variables are of the file transformation or scanning type, a subfunction "SUBF" will be called after each observation, i.e. after the "simple" variables, if any, have received their values. In the subfunction the missing "non-simple" variables are calculated.

A certain subfunction belongs to a given job. If the job has a namefile "NFnn", the subfunction must be stored on the GANDALF disk under the name "SUBFnn", where nn represents the namefile number. The subfunction should be made and stored on the disk before starting GANDALF. If a subfunction is missing, GANDALF will print the following message before an investigation can begin: "SAVE SUBROUTINE!". The user should then clear the memory and make the subroutine, save it on the disk and start GANDALF from the beginning.

In Fig. 8 a block diagram is shown which demonstrates the main functions performed during an investigation. There are two main loops controlled by the two variables I and J. I is the observation number and J is the variable number. K is the total number of variables in the problem. If a variable is of a type which requires a subfunction, this subfunction is called at the end of each observation, i.e. when J=K. During each observation the input data are stored in the variables X[J]. After each observation and after the call of the subfunction the X array is transferred to r-variables according to the formula:

$$X[J] \rightarrow r(Q+IK+J). \qquad \text{(Eq. 1)}$$

where Q = 9 + P - K.

The subroutine must satisfy certain requirements concerning the variables in order to communicate with the main program in the right manner. The first program line must contain the label "SUBF":, and the subfunction must contain a ret statement. The following principles should be observed:

The X[*] array: This array contains the current input data as discussed above.

The P[*] array contains the values of the parameters.

The simple variables from A to G can be used in the subroutine, and they are not changed from one observation to the next.

p-variables: For internal use p1,p2,... can be used but their values disappear when the subroutine is left, i.e. between each observation.

r-variables: Contain data from all earlier observations and can be used according to Eq. 1 (see also Appendix II).

The following arrays, which are dimensioned in GANDALF: A[5,6], D[10], Q[10], R[5], Z[5] can be used in subroutines. Their values are not changed during the input of data to an investigation, but they may be changed e.g. during a regression analysis.

The following arrays are not dimensioned by GANDALF and can be used when they are dimensioned: E,F,G,H,I,J,M,N,O,U,W. For a list of variables see App. V.

Flags: The flags 10,11, and 12 may be used in subroutines. If flag 6 is set, the subroutine is called during the investigation, if it is cleared, the subfunction is used by BLOCK-TRANS, see 6.4. A list of the effects of the other flags are given in App. IV.

In the following paragraphs a few examples of subfunctions are given.

6.3.1 Simple function

Assume that you want to calculate and to plot f as a function of k from the following expression

$$f^2 = f_p^2 \frac{(ka)^2}{1+(ka)^2} + 3(ck/2\pi)^2$$

where f_p, a and c are parameters. This task can be performed by making the following namefile and subroutine:

```
   1983-3-10
================
  T.G.DISP.REL.
=================
PARAMETERS:
a
c
fp
X1
X2
DX
VARIABLES:
k[cm-1]
f[Hz]
 I    TYPE    PLOT
=================
 1       2       1
 2       2       2
=================

 I     CONSTANTS:
=================
 1     1.0000E 00
 2     1.0000E 00
```

```
0: % "SUBF11"
1: "SUBF":
2: if I=1;3(P[2]/2π)↑2→A;P[3]↑2→B
3: (P[1]X[1])↑2→p1
4: √(Bp1/(1+p1)+AX[1]↑2)→X[2]
5: ret
```

As it can be seen from the namefile the two variables, k and f, are of the computed type and GANDALF therefore has added the three parameters X1, X2 and DX, which determine the interval and steplength for the variable k.

Line "Ø" of the subfunction is a comment to remind the user that this subroutine is stored on the disk under the filename "SUBF11". In line "1" the label "SUBF" is found, and GANDALF will jump to this when necessary. In line 2 some constants are calculated to save time, since this line only is performed in the beginning of the first call of the subroutine (when I = 1). In line 3 to 4 the function is calculated and the return statement is in line 5. When this 5-lines program has been written and stored on the disk, the function can be calculated, plotted and written out as a table as you wish.

6.3.2 D/A Command.

Now an example is given which demonstrates the use of a D/A command and value assignment directly to r-variables. Assume you want to measure two signals: SIGNAL1 and SIGNAL2 which you have in the A/D converter channel 1 and channel 2. The signals are measured as a function of a time delay, and this time delay can be regulated by the analog output of the D/A-converter which should be equal to the time delay multiplied by a constant. The name file and the subroutine could be the following:

```
   1983-3-10
================
   D/A-DEMO
================
PARAMETERS:
POSITION
DENSITY
X1
X2
DX
VARIABLES:
DELAY[ms]
D/A-OUTPUT
SIGNAL1
SIGNAL2
 I    TYPE   PLOT
================
 1       2      1
 2       0      2
 3      11      3
 4      12      4
================

 I     CONSTANTS:
================
 1     1.0000E 00
 2     1.0000E 00
 3     1.0000E 00
 4     1.0000E 00
```

```
0: % "SUBF12"
1: "SUBF":
2: C[1]X[1]→X[2]
3: fmt 9,wrt708.9,X[2]
4: ret
```

The first two lines of the subfunction is similar to the first example. In line 2 the first variable DELAY is multiplied by the constant C[1] to give the second variable, which in line 3 sets the D/A converter to give a signal in volts equal to X[2]. The constant C[1] is according to the namefile equal to 1, but can at any time the display shows: "SELECT KEY" be changed by the user by pressing the key function CONST.

6.3.3 File transformation

The next example demonstrates the use of file transformation variables (variable type 6).

Assume we have performed a lot of measurements of the type discussed in the last example (6.3.2) and that we have stored all the investigations on the disk. Now we want to calculate the amplitude and the phase according to the following formulas:

$$\text{AMPLITUDE} = (\text{SIGNAL1}^2 + \text{SIGNAL2}^2)^{1/2}$$
$$\text{PHASE} = \text{arc.tan.}(\text{SIGNAL2}/\text{SIGNAL1})$$

together with the time delay, but we do not need the second variable (the D/A) from the last example.
We can perform this task by making a new namefile and a subroutine as follows:

```
   1983-3-10
=================
  FIL-VAR.DEMO
=================
PARAMETERS:
VARIABLES:
DELAY[ms]
AMPLITUDE
PHASE
 I    TYPE    PLOT
=================
 1       6       1
 2       6       2
 3       6       3
=================

 I     CONSTANTS:
=================
 1     1.0000E 00
 2     1.0000E 00
 3     1.0000E 00
```

```
0: % "SUBF13"
1: "SUBF":
2: r(X[3]↑2+X[4]↑2)→X[2]
3: atn(X[4]/X[3])→X[3]
4: ret
```

When an investigation is performed under this new namefile, GANDALF asks from which datafile (data file #) the input should be taken. Then for each observation of this datafile the variables are transformed according to the expressions in the subroutine and the new variables are the input to the current investigation.

As it can be seen from the subroutine, X[1] is not changed but the AMPLITUDE goes to X[2] and the PHASE to X[3].

If the variables are of the type 5 (file input), data from the disk belonging to another namefile can be input to an investigation under the present namefile similar to the example shown above, but without the transformation in the subroutine. The idea is that the number of variables in the new namefile may be different from the number in the old one. It could be smaller in order to save space on the disk, if all the variables do not have to be stored, or it could be larger, if some new variables are needed. These can be calculated afterwards from the old variables using a Block transformation (see 6.4).

6.3.4. Scanning

The last example presents the case when the variables are of type 7, i.e. the scanning file type.

Assume again that we have used the job "DEMO" from 6.3.2 to perform some investigations and that all the results are stored as data files on the disk. In each of these investigations we have measurements of the two signals as function of a delay, and for each investigation we have a certain value of the parameter called "POSITION". Now assume that we want to have the signals of function of POSITION with the delay as a paramter instead. That is, we have to scan through many investigations to get the necessary values. To perform this task we use the following namefile and subfunction:

```
   1983-3-10
================
  SCANNING DEMO
================
PARAMETERS:
DELAY[ms]
FIRST FILE #
LAST FILE #
VARIABLES:
POSITION
SIGNAL1
SIGNAL2
 I    TYPE    PLOT
================
 1      7       1
 2      7       2
 3      7       3
================

 I     CONSTANTS:
================
 1     1.0000E 00
 2     1.0000E 00
 3     1.0000E 00
```

```
0: % "SUBF15"
1: "SUBF":
2: if I=1;p(11+4P[1]→A)→P[1]
3: p(A+2)→X[2]
4: p(A+3)→X[3]
5: p10→X[1]
6: ret
```

When the variables are of this type two parameters are automatically added to the parameter list: "FIRST FILE #" and "LAST FILE #". The files have to be in consecutive order, but it doesn't matter if some files are missing. That is, if some data files in between contains wrong information they could simply be killed.

For each observation in an investigation of this type a subfunction "FILE INPUT" is called and all the data from a data file, starting with the "FIRST FILE #" and ending with the "LAST FILE #" are transferred to p-variables. After this the user subfunction is called and the transformation specified by the user is performed. In other words the data from the original investigation are now available as p-variables corresponding to the original r-variables in which the data were saved on the disk. In the subroutine a function of these p-variables can be assigned to the normal X[J] variables, or in a more advanced case to new r-variables.

For the use of the scanning transformation the formula (1) and the table of r-variables in App. II are essential.

In the example shown the parameter "DELAY" is used in two ways. When the scanning is performed this parameter is constant, but GANDALF must know in which observation of the original investigation this DELAY can be found. The user therefore must key-in the appropriate observation # when the question "DELAY?" is asked from the display. Then, in the second line of the subfunction, the right value for the delay is found and the constant A is calculated. A is the number of the r-variable in which the first variable in the observation containing data corresponding to the correct delay is stored. In line 3 of the subfunction the SIGNAL1 of the original data file is assigned to the second variable (X[2]), and in line 4 the SIGNAL2 is assigned to the third variable (X[3]). In line 5 the parameter POSITION is assigned to the variable "POSITION" of the new investigation.

6.3.5. OPTION

If the user wants to use a certain program without interrupting GANDALF, for example if he wants to do some calculations between each investigation, he can use the "OPTION" facility. If the subfunction has a label OPTION, GANDALF will continue at this label when the key function OPTION is pressed. The following example shows how it can be used.

```
0: % "SUBF16"
1: "SUBF":
2: 'ERF'(X[1])→X[2];ret
3: "OPTION":
4: ent "X?",A
5: prt "X=",A," Erf(X)=",'ERF'(A)
6: gto "CHOOSE"
7: "ERF":
8: % "HERE COMES A SUBFUNCTION WHICH CALCULATES THE ERROR FUNCTION"
9: ret
```

This subfunction calculates the error function. If you just want one value of the error function, it is enough to press the key function OPTION, then enter the x value, and erf(x) will be printed out on the line printer. A subfunction using the OPTION option must contain the label "OPTION": and a - gto "CHOOSE" - statement, in order to return to normal GANDALF operation. The OPTION possibility is especially useful in connection to A/D and D/A operations.

6.3.6. TERMINATE

If it is necessary to stop GANDALF during a task, the key function TERMINATE should be used instead of the stop-key, which only should be used in emergency cases. The effect of the key function TERMINATE is that flag 8 will be set and this will stop GANDALF during an investigation when the current observation is finished.

If the key function TERMINATE does not stop GANDALF, the normal stop-key will stop the program, and then the key functions can be used to proceed. However, if GANDALF is stopped in the middle of a subroutine it will miss the return statement, and next time a new program file is going to be loaded into the memory, you will get error 63 (see appendix VI).

6.4. Block transformation

In section 6.3 it was discussed how to transform some variables into new variables during an investigation by using a transformation defined by the user in a subfunction. After an investigation has been performed it is still possible to transform the variables. There are three kinds of transformation called: 1) simple transformation, 2) two file operation, and 3) user defined block transformation. Each of these are discussed in the following subsections.

To start a block transformation use the key function BLOCK TRANS and you will get the question:

6.4.1 [SIMPLE TRANSFORMATION?] <Ø,1>

<Ø>: see 6.4.2

<1> With this answer the next question will be:

[TYPE? (X+C:1, X*C:2, X↑C:3)] <1,2,3>

<1>: The transformation X[J] + C → X[J] is performed when GANDALF has asked what C and J should be.

<2>: The transformation X[J] * C → X[J] is performed.

<3>: The transformation X[J] ↑ C → X[J] is performed.

6.4.2. [TWO FILE OPERATION?] <Ø,1>

<Ø>: See 6.4.3.

<1>: A general two file operation will be of the type: $F(X_{i,j},Y_{i,k}) \rightarrow X_{i,l}$, where $X_{i,j}$ represents the variables of the i^{th} observation of the investigation already in the machine memory, while $Y_{i,k}$ represents the corresponding variables in an investigation produced with the same name-file and stored on the disk. The same transformation is performed for each observation. It can for instance be used if the data have to be calibrated according to some calibration curve. In a subfunction the variables from the investigation fetched from the disk is accessible in the Y-array, i.e. one observation at a time is assigned to the Y-array.

After a question about the disk file # of the second investigation the user gets the following question:

[X[J] + C* Y[J] → X[J]?] <Ø,1>

<Ø>: If you cannot use this simple two file operation you must define your own transformation by making a subfunction. This should be stored on the disk before starting GANDALF, except if it is a very simple one which only is going to be used once. In that case it could be stored in the end of the "DISPRT" program according to the instructions given by GANDALF.

<2>: In this case the simple two file operation: X[J] + C* Y[J] → X[J] will be performed for each observation when J has been entered.

When the transformation is finished GANDALF displays: "SELECT KEY!".

6.4.3. User defined transformation

If the transformation is not of the simple type, it requires a subfunction. There is no difference between subfuctions used in connection with block transformation and subfunctions used during an investigation. Therefore, the description in Sec. 6.3 about how to make a subfunction is also valid when making a subfunction for a block transformation. A certain job can only have one corresponding subfunction. If a job needs both a subfunction for the investigations and one for block transformation, everything must be specified in the same subfunction. During an investigation flag 6 will be true while it will be false during a block transformation.

6.5. Data to disk

After an investigation or whenever GANDALF displays SELECT KEY, the data in the memory can be stored on the disk by pressing the key function BLOCK→DISK. When the data have been stored, a message is written on the internal printer about which disk and under which filename the data were stored. This filename has the general form: Dnnddd where nn represents the name file number and ddd the data file number. For each new data file stored on the disk the data file number is automatically increased by one. No more than 999 data files under the same name can be stored on one disk.

If some data already stored on the disk are wanted back for further processing the key NEW JOB should be used, see 6.1.2.

6.6. Plotting

As described in Sec. 5.2., GANDALF has several plotting facilities. Each of these are described in this section in some details. Especially, the questions GANDALF asks, which are not self-explanatory, will be discussed, using the symbolic form introduced in the beginning of Sec. 6.

6.6.1 "BLOCK PLOT"

A plot similar to the one shown in Fig. 1 of one or more of the variables as a function of one of the variables can be obtained by using the key function "BLOCK PLOT". The first question, the user gets, is:

[FORMAT * ? (IF A4 PRESS CONTINUE)] <continue, Ø,1,2,N>

<continue>: You get the plot on a standard A4 page.
<Ø>: P1 and P2 will be the same as the last plot.
<1>: The user is asked both to set P1 and P2 and to define the exact scale.
<N>: If N is anything else, you get some user instructions printed out.

[NEW VARIABLES?] <Ø,1>

<Ø>: The variables are plotted as in the last plot, if any, otherwise as specified in the namefile.
<1>: You can now change the plot type, see 6.1.5.3.

The plotting of the axes is performed rather automatic. GANDALF will suggest some limits for the axes, but you can change them if you wish. However, even if you change the limits for the axes GANDALF may perform some round-off in order to get reasonable tic-marks.

Three questions which need some explanation are:

[PLOT SYMBOLS?] <Ø,1>

<Ø> If your answer is no, the curve(s) will be plotted with the line type or the symbol which you have used last time a plot was made, if any, otherwise with the default type. The default values are chosen so that the various line types are used starting with a solid line, except in the case where the first variable is of the type 1 (manual input) or type 5 or 6 (file input), in which case the symbols X,O,*,+, ,-,=,% are used in this order.

<1>: In this case the line type or symbol which will be used for plotting is shown on the right hand side of the paper. If the line type or the symbol is going to be changed, you must also give this answer, and you will then get the question:

[LINE # OR SYMBOL FOR "VARIABLE NAME"?]

<continue, -1,Ø,1,2, ...,6,character>

<continue>: The variable will be plotted with a solid line.

<-1>: The variable will be plotted according to the default value unless it has been changed during the job.

<N>: If N is a digit between Ø and 6, the curve will be plotted with the corresponding line type (see Ref. 3).

<character>: If any other character than the ones mentioned above is used each observation of the variable will be plotted with this character.

[PLOT PARAMETERS?] <continue,Ø,1,2, ...,P>

<continue>: The name and the value of the first parameter will be written on the right hand side of the plot.

<Ø>: No information about the parameters is written.

<1>: All parameter names and their values will be written.

<N>: If N is a number between 2 and P (the total number of parameters), the first N parameter names and their values will be written on the right hand side of the plot.

6.6.2 Single plot.

If the single plot mode is set (flag 2), the variables will not be plotted immediately after each observation. The single plot mode can be set on or off in the beginning of an investigation (see 6.2.1). If the single mode is on you get the following question in the beginning of the investigation:

[AXES(1), TEXT(2), OR CONT.!] <continue,1,2>

<continue>: You neither get axes on the plot nor any text about symbols or parameters. Note, it is necessary to make a scaling before any plotting of curves. The scaling can be performed either when axes are drawn or when the key function SCALE is used.

<1>: In this case you get both scaling, axes and the possibility for writing information about symbols and parameters on the plot. The further questions are the same as in BLOCK PLOT (see 6.6.1).

<2>: By using this answer you only get text on your plot.

6.6.3 Three dimensional plotting

There are three different ways of performing 3-dimensional plots, the simple 3-dim-plot, the contour plot and the netsurface plot. The netsurface plot is actually a separate program which can be used without going through GANDALF, and it does not require any namefiles. The contour plot and the netsurface plot are option possibilities when GANDALF is started. The simple 3-dim-plot requires a namefile and can only be used after a namefile has been loaded. All the 3-dim-plots can be started when the display shows: "SELECT KEY!" by pressing the key function: "3DPLOT". The main effect of the three 3-dimensional plotting options has already been explained in Sec. 4.2.2 and shall not be repeated here, but some details will be discussed a little further.

The data to be plotted are assumed to exist as investigations stored on the disk under the same namefile. However, the netsurface plot program can also be run completely separate from GANDALF and plot a user defined function, see Sec. 6.6.3.3. If investigations are stored on the disk with data file numbers in consecutive order, only the date file number of the first and last data file have to be specified, otherwise the data file number of each data file must be typed-in.

In principle all the 3 dimensional plot functions show a series of investigations as a function of a parameter. For the simple 3-dim-plot there are no restrictions on the variations of this parameter as a function of the disk file number. However, for the contour plot and the netsurface plot, the parameter, which now is an independent variable, must increase (or decrease) with a constant value for each new investigation to be plotted. If the disk file numbers are in consecutive order, this means that the parameter must be a linear function of the disk file number. However, it should be noted that it does not matter if some date files have been killed even when it is specified that the files are in consecutive order.

6.6.3.1 3-DIM-PLOT. The first question to answer after the user has decided to use the simple 3-dim-plot will be:

[PLOT OF AXES?] <Ø,1>

<Ø>: May only be used if a scaling already exists. If scaling without axes is wanted, use key function "SCALE" (see 6.6.4).

<1>: The axes, the scaling and possible text can be performed and plotted as in "BLOCK PLOT" (see 6.6.1).

[PARAMETER #?] <N>

<N>: The number of the parameter, which is going to be one of the two independent variables, must be entered. Ø<N<=P, where P is the total number of parameters.

[X-OFS/PARAMETER UNIT?] <A>

<A>: The number A gives an offset in the X-direction determining where the variables are plotted with respect to the original scaling. The simple 3-dim-plot is similar to a block plot performed for each data file, and when a new data file (investigation) is plotted the X-coordinates will be off-set by A×P[N], where P[N] is the varying parameter. The X-off-seet is measured in the same unit as the x-axis. Note also that the variables are plotted according to the plot type as given in the namefile. If the plot type of the variables has to be changed, it can be done when the axes are plotted.

[Y-OFS/PARAMETER UNIT?] <B>

<B>: Gives the off-set in the Y-direction, see the comments above. In the end of the 3-dim-plot there is a possibility for plotting the Z-axis. If a Z-axis is wanted, GANDALF asks about the "X- and Y-cordinates", which are the coordinates for the starting point of the Z-axis, while "Z-length" is in the unit of the parameter.

6.6.3.2 CONTOUR PLOT. This program is one of the Hewlett Packard demonstration programs modified so it can read data from the disk file when these data come from an investigation performed with GANDALF. All questions should be self-explanatory when sections 4.2.2 and 6.6.3 have been read. The key-functions as defined by GANDALF are not available during the contour plot program, although some of them may work.

6.6.3.3 NETSURFACE. As mentioned previously, the three dimensional plotting program in the disk file named "NETSUR" is not really an integrated part of GANDALF, since GANDALF gets "NETSUR" instead of chaining "NETSUR". The difference is that getting a program file (contrary to chaining it) causes all the variables in the memory to be erased. Another consequence is that the key-functions are inactive. On the other hand, it is not necessary to use GANDALF at all in order to start a netsurface job: simply get "NETSUR" and run it. NETSURFACE produces 3-dimensional plots as described in Sec. 4.2.2 and shown in Fig. 4. Either data stored previously in datafiles or user defined arithmetical functions can be plotted.

In the case of previously stored data one variable, called the Z-variable, is plotted as a function of the observation number and the number of the file in which the data are stored. In order to distinguish between the two independent variables corresponding to the observation and file number, respectively, the user is asked to give a maximum 16 character name to what is called the "observation variable", and the "file variable". When NETSURFACE asks for the minimum and maximum values that the user wants to use for these two independent variables, it should be remembered that they both are integers. The value of the observation variable is quite obvious, since it simply is the observation number, and the value of the file variable refers to the number the file has in the order the files were read into NETSURFACE, i.e. the value of the file variable is an integer between 1 and the total number of files entered into NETSURFACE.

NETSURFACE shows the projection of an axiparallel rectangular net on the Z-surface and the distance in the "NN" direction (where NN denotes either X or Y) between two nearest lines perpendicular to the "NN" axis is called the "NN" DIVISION. If "NN" DIVISION = 1 a line is drawn through each of the Z-values. The user is free to specify any real value for "NN" DIVISION.

When producing the plot, NETSURFACE makes use of imaginary equidistant vertical resolution lines. The user is free to choose the number of resolution lines between a lower limit determined by NETSURFACE on basis of the specifications of the plot and an upper limit of 500 corresponding to a 0.5 mm resolution on an A4 size paper. The user must also specify the angle, V, between the axes of the independent variables on the plot and the "SCALE" of each of the three variables. NN-SCALE is a number multiplied on the NN-variable before the plot is performed. NETSURFACE automatically ensures that the plot is kept within the paper size by internally performing an additional scaling. Since the additional vertical and horizontal scale factor usually are not equal, the angle V on the final plot has different value from what was specified by the user. When performing a NETSURFACE plot of data for the first time it is recommended to start with divisions corresponding to approximately 5 lines in each direction, a low number of resolution lines, an angle of 60^{o}, and scale-factors, making the three variables vary over approximately the same range Then make several plots varying the scale-factors but with constant V, and, finally, decrease the division to the desired netsize and increase the number of resolution lines.

In case of a user defined arithmetical function, NETSURFACE works almost in the same way, but the two independent variables are now called X and Y. They may be given other names by the user. The subfunction for calculation of Z(X,Y) can be saved in a program file with any name and must have the following construction:

```
0: % "MYFUNC"
1: "Z-VAL":
2: p1→J;p2→I;if flg1;p1→I;p2→J
3:
4: % "CALCULATION OF Z"
5:
6: ret Z
```

In the subfunction I and J have the roles of the X- and Y-variable, respectively. The user is free to use the variables D,E,L,P,U and Z and the arrays C-G (after dimensioning) in the subfunction "Z-VAL".

After performing the 3-dimensional plot, NETSURFACE gives the possibility of drawing axes at a user defined Z-height and of writing some additional text on the plot, as shown in Fig. 4.

6.6.4 SCALE

If a scale of a plot is wanted without drawing axes the key-function "SCALE" can be used. This is especially useful if the plotter is used in the "XY-RECORDER-MODE" (see Sec. 6.6.6). The user gets the following questions which need some comments:

[FORMAT #?] <continue, -1,Ø,1, ... 2Ø>

<-1>: Information about the various format possibilities is written on the printer.

<continue>: The standard A4 format is assumed. The plotters P1 and P2-points will be in the corners of a horizontally placed piece of A4 paper, while scaling points will be somewhat inside. The scaling points X1,X2, Y1,Y2 are the points the user is asked to define in order to perform the scaling command.

<Ø>: P1/P2 are not changed, but the user can give new scaling points, which here are the coordinates of P1 and P2.

<1>: The user is asked to set P1 and P2. The scaling point will be somewhat inside the rectangular defined by P1/P2.

<2>: As <1>, but the scaling points are the coordinates of P1/P2.

<N>: If 2<N<20, P1 and P2 will be in the corner of a horizontally placed paper with a format AN, the scaling points are the coordinates of P1 and P2.

6.6.5 X-OFS and Y-OFS

The scaling can be off-set in the X-direction or in the Y-direction by using the key-functions "X-OFS" or "Y-OFS", respectively. The units are in % of the full scale. With a 100% off-set performed three times, it is possible to make 4 A5 drawings on a A3 piece of paper with just one scaling.

6.6.6 XY-Recorder mode

If the first variable is of the A/D-type, the parameter DX is set to zero, and the single-plot-mode is on, GANDALF will run the plotter as an XY-recorder. That is, the variables measured with the A/D-converter will be plotted simultaneously, but they are not loaded into r-variables, and the memory will therefore not be overloaded. To stop the recording use the key-function TERMINATE, and to start again, press "NEW INVEST". In this mode, the "SCALE" (see 6.6.4) and the "OFF-SET" (see 6.6.5) options are particularly useful for scaling of the plot.

6.6.7 SAVE PLOT

If the plotter has been turned off or disconnected, the information about P1, P2 and the scaling can be restored by pressing the key-function "SAVE PLOT".

6.7 Printing

6.7.1. BLOCK PRINT

All data in an investigation, i.e. all observations and parameters, can be printed as a table on the line printer by using the key-function BLOCK PRINT. The printing is stopped after each A4-sized page so the user can tear off the paper. If it is necessary to stop the printing before it has finished, the key function TERMINATE should be used. When the table is finished you can get a full A4-page out by pressing the key-function "PAGEOUT". If this is not done, the line counter is not set to zero, which may be a disadvantage later on. If necesary, the line counter can be reset by writing: "Ø→H" and pressing: "EXECUTE".

6.7.2 SINGLE PRINT

Information about parameters and variables can also be printed out on the internal printer during the investigation. The SINGLE PRINT mode can be set in the beginning of each investigation.

6.7.3 LIST

A listing of information of all or some of the name-files on the disk, or of the data files under a certain namefile, or of the program which is in the machine can be obtained by using the key-function LIST.

6.8. Data processing

6.8.1 Regression analysis

Since an introduction to the regression analysis was given in Sec. 5.3 and all the questions in the regression program should be self-explanatory, it shall only be discussed here how to make a regression function to be used in regression model 9, the general one, which has to be used if no other models are applicable. A regression function is a subroutine which defines the regression model. As an example, consider a case where some point should be fitted to a third order polynomium. The following subroutine will do this job:

```
0: % "REGF31"
1: "F":
2: ret ((Q[1]p1+Q[2])p1+Q[3])p1+Q[4]
```

From this example it can be seen that the independent variable is written as p1, and the parameters are denoted Q[1], Q[2], ... Q[4]. If is is necessary to use other variables in the subroutine only p-variables should be used, since the simple variables, A, B, ..Z may interfere with the regression program. The subroutine, must start with the label "F":. During the regression process the current parameter values until a maximum of three are shown on the display. If the measurements to fit are too bad or the guess values are too far off, the process may fail. In this case you get a message on the printer, and you can try again with another guess. Parameters in an exponent are more likely to cause divergence, when their guess values are too far off. If the message "DETERMINANT=O" is printed, the process is stopped because it is not possible to calculate either the standard deviation or the corrections for the parameters.

After each run the values of the parameters and their standard deviations are printed. Also the residual sum of squares and the coefficient of multiple determination (R↑2) are printed. For further information see Ref. 4.

6.8.2 SPLINE

The subroutine SPLINE uses cubic spline interpolation to compute a smooth curve s(x) through n unequally spaced data points (x_i,y_i). The x- and y-variable can be any variable in a previously stored datafile, as long as $x_i<x_{i+1}$ for $1<i<n-1$. SPLINE produces new data points (t_j,s_j) where $s_j = s(t_j)$ and $1<j<m$. The t-variables are equally spaced from x_1 to x_n and the user shall specify the total number, m, of the t-values (called the "NEW X-VALUES" by SPLINE).

In order to calculate the cubic spline function a set of coupled linear equations are solved by iteration. The absolute accuracy, EPSILON, to which these equations are solved, i.e. the maximum change in s_j between two iterations, is specified by the user.

In addition to the $s(t_j)$ values SPLINE calculates the derivatives $s'(t_j)$ and the integral of s(x) from x_1 to x_n. The values of t_j, s_j and s'_j are stored in the arrays T[M], B[M] and D[M], respectively. These data can be saved in a new datafile under either the same namefile as used for the input of x_i and y_i or a different namefile, with the same parameters as the input-namefile and a total number, K, of variables per observation greater than 1. If K=2, the new x-values, i.e. t_j, are stored in the first variable and the new y-values, i.e. s_j, are stored in the second variable. If $3<K<10$ the third variable contains s'_j, while the variables with higher numbers are zero.

SPLINE contains the Hewlett-Packard subroutine "DERINT" and for further references on the method used in the calculation of the spline function see Ref. 2, page 141.

As in the case of CONTOUR PLOT and NETSURFACE the special key-functions are inactive during a SPLINE job. When the SPLINE job is over, the user has the possibility to return to the ordinary GANDALF program.

6.9. Initialization of new disks

As discussed in the beginning of Sec. 6, the GANDALF program is placed physically on a disk called the GANDALF disk which must be placed in disk drive 0. Also the namefiles are placed on this disk. Subfunctions and regression functions must also be placed on the GANDALF disk. The data files, on the other hand, are stored on a data disk which has to be in disk drive 1. When a data disk is filled up with data files, or when the data file number exceeds 999, a new data disk must be initialized. First the disk must be initialized according to the HP-manual (Ref. 5). Then a GANDALF initialization is necessary. This is performed by the program "D-INIT", which is on the GANDALF disk. To start, place the GANDALF disk in disk drive Ø and the new initialized data in disk drive 1. Then type in:

<get "D-INIT", Ø, Ø>

press "CONTINUE" and answer the simple questions which GANDALF asks. The effect is that a special file called DATINF is opened on the data disk. This file contains information on how many datafiles are placed on the data disk and a few other things.

When 99 namefiles are stored on the GANDALF disk or when the GANDALF disk is filled up with subfunctions etc., it cannot be used any more and a new GANDALF disk must be initialized. This task can be performed in the same way as described above for data disks. That is, the GANDALF MASTER disk is placed in drive Ø, and the new initialized disk in drive 1, and the "D-INIT" program is run. On the new GANDALF disk, a file GANINF is placed. This file contains information about the total number of namefiles on the disk and some other things. When a new GANDALF disk has to be initialized, the GANDALF MASTER disk should be used for the initialization process, since this disk contains the official and updated version of GANDALF. When a data disk corresponding to the new GANDALF disk is initialized, the new GANDALF disk must be placed in drive Ø.

ACKNOWLEDGMENTS

The quotation in the introduction to "The Lord of the Rings" was brought to light by Dr. Roman Schrittwieser who is gratefully acknowledged.

We thank Peter van Doremalen for making the first version of the regression analysis program, while visiting Risø as an IASTE student in 1978. We are also grateful for the careful typing of the manuscript by Birthe S. Jørgensen.

GANDALF has been tested carefully by several of our colleagues, especially by Dr. Jens Juul Rasmussen who provoked the first version of GANDALF to make erroneous stops every time he used the program. The authors highly appreciate this effort, since it forced us to develope a nearly fool proof version of GANDALF.

REFERENCES

1. Analog to Digital Converter, Hewlett-Packard Manual No. HP 59313A pp. 3-6 to 3-7.

2. General Utility Routine; Hewlett-Packard Manual 09825-10001 p. 141.

3. I/O Control Reference, Hewlett-Packard Manual 09825-90210.

4. J. Beck and K. Arnold, Parameter Estimation in Engineering and Science, J. Wiley & Sons, New York 1977.

5. Disk Programming, Hewlett-Packard Manual 9885 M, p. 90.

6. HP 9825 Desktop Computer Operating and Programming Reference, Hewlett-Packard Manual No. 09825-90200

APPENDIX I

Information stored in a namefile

Variable name	Content
K	The number of variables
P	The number of parameters
I$[16]	The job name
X$[1Ø,16]	The variable names
P$[1Ø,16]	The parameter names
T[1Ø]	Variable type numbers
C[1Ø]	Variable constants
V[1Ø]	Plot type numbers

APPENDIX II

Information stored in a data file (r-variables)

r-variable	Content
rØ	The total number of r-variables used in the investigation
r1	The day the investigation was performed
r2	The month - " -
r3	The year - " -
r4	The investigation number
r5	Number of observations
r6	Number of variables = K
r7	Number of parameters = P
r8	Namefile number
r9	Datafile number
r1Ø - r(9+P)	Values of parameters
r(1Ø+P) - r(rØ)	Values of the variables

The J'th variable of the I'th observation is in r(Q+IK+J), where Q = 9+P-K

APPENDIX III

Program files

File name	Main task
GANDAL	Initializes GANDALF and makes new namefiles
GINPUT	Performs input tasks and transfers datafiles to disk
DISPRT	Contains BLOCK PRINT and BLOCK TRANS
DPPLOT	Contains BLOCK PLOT and also makes axes and text for SINGLE PLOT
REGAN1	Main program for regression analysis
REGMØ1 - REGMØ9	Contain the various regression models
3DPLOT	Simple 3-dimensional plotting
CONTUR	Contour plotting program
NETSUR	Netsurface plotting program
SPLINE	Constructs spline functions
LIST	Performs listing of other programs
NFLIST	Performs listing of namefiles
DFLIST	Performs listing of datafiles
D-INIT	Initializes new data disks and new GANDALF disks

APPENDIX IV

Flags

Several of the flags are used for various purposes in the different program files. Below is a list of the use of the flags during the input of data. (GINPUT program file).

Flag number	Effect
Ø	Digitized variables in this job
1	Single print mode
2	Single plot mode
3	Correction of data (for single print mode)
4	Correction of data (for digitized input)
5	XY-recorder mode
6	Subfunction is available
7	Single plot pen position
8	A/D-stop
9	Auto-mode
10-12	Can be used in subfunctions
13	Normal use (see Ref. 6)
14	Math. error eliminated (see Ref. 6)
15	Math. error observed (see Ref. 6)

APPENDIX V

List of variables

simple variables

A-F: Various use (can be used in subfunctions)

G: Label No. for file switching

H: Printer line counter

I: Current observation No.

J: -"- variable No.

K: Number of variables = r6

L: Last saved obs. No.

M-N: Various use

O: Contains global flags

P: Number of parameters = r7

Q: = 9+P-K

R-W: Various use

X: log (x min) for logarithmic plot of x-variable

Y: log (y min) for logarithmic plot of y-variable

Array variables

A[5,6]:	Regression analysis
B[10]:	Can be used in subfunctions
D[8]:	Disk information
K[10]:	Regression analysis
L[13]:	Plotter positions
P[10]:	Parameters
Q[10]:	Parameters for the general regression model
R[10]:	Regression analysis
S[5]:	- " -
X[10]:	Current variable
Y[0:31]:	Various use
Z[5]:	- " -
A$[10]:	- " -
C$[200]:	Disk information
F$[C]:	Name- and disk file names
H$[4,13]:	A/D input
K$[4]:	Various use
S$[9]:	Plot symbols or line types
V$[10,16]:	Variable names in block-print

See also Appendices I and II.

APPENDIX VI

Error 63

If GANDALF has been stopped by the STOP-key in the middle of a subroutine, it will miss the return statement. However, an error message:

"Error 63"

will not appear before the next programme file is fetched from the disk. To get out of this problem the user must type-in the following:

<fetch 999>, EXECUTE,

to get the last line in the programme. The display shows:

"nal"

where nal reprersents the line number of the "next available line". Then type-in

<ret>, STORE, <cont nal>, EXECUTE

Now you will get "error 28", but this does not matter. You can use the key-functions in order to proceed.

Risø National Laboratory **Risø - M - 2393**

Risø - M - 2393

Title and author(s)	Date January 1983
GANDALF - A Generally Applicable Numerical Data Acquisition Laboratory Facility J.P. Lynov and P. Michelsen Association Euratom - Risø National Laboratory	Department or group Physics Dept.
73 pages + tables + illustrations	Group's own registration number(s)

Abstract

This report contains a description of an interactive computer program, GANDALF, for general purpose data handling. GANDALF is designed for different types of data input, it has flexible data processing possibilities, and it can produce various kinds of data output. The program is written for an HP-9825 desk-top computer in HPL-language. However, the fundamental program structure can also be applied to other computer systems with other computer languages.

Copies to

Available on request from Risø Library, Risø National Laboratory (Risø Bibliotek), Forsøgsanlæg Risø), DK-4000 Roskilde, Denmark
Telephone: (02) 37 12 12, ext. 2262. Telex: 43116