SA(IR)-254

UNITED STATES DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

SAUDI ARABIAN PROJECT REPORT 254

G N A P

(GRAPHIC NORMATIVE ANALYSIS PROGRAM)

by

Roger W. Bowen

amended by

John Odell and Lester D. North

U. S. Geological Survey
OPEN FILE REPORT 79-1211

This report is preliminary and has not been edited or reviewed for conformity with U.S. Geological Survey standards and nomenclature.

PREPARED FOR
DIRECTORATE GENERAL OF MINERAL RESOURCES
MINISTRY OF PETROLEUM AND MINERAL RESOURCES
JIDDAH, SAUDI ARABIA
1979

U.S. GEOLOGICAL SURVEY

SAUDI ARABIAN PROJECT REPORT 254

G N A P

(GRAPHIC NORMATIVE ANALYSIS PROGRAM)

by

Roger W. Bowen

amended by

John Odell and Lester D. North

U.S. Geological Survey
Jiddah, Saudi Arabia

1979

CONTENTS

ILLUSTRATIONS

TABLES

LISTINGS

G N A P

(GRAPHIC NORMATIVE ANALYSIS PROGRAM)

by

Roger W. Bowen[1/]

amended by

John Odell[2/] and Lester D. North[3/]

ABSTRACT

A user-oriented command language is developed to provide direct control over the computation and output of the standard CIPW norm. A user-supplied input format for the oxide values may be given or a standard CIPW Rock Analysis format may be used. Once the oxide values have been read by the computer, these values may be manipulated by the user and the "norm" recalculated on the basis of the manipulated or "adjusted" values. Additional output capabilities include tabular listing of computed values, summary listings suitable for publication, x-y plots, and ternary diagrams. As many as 20 rock analysis cards may be processed as a group. Any number of such groups may be processed in any one computer run.

INTRODUCTION AND USE

GNAP (Graphic Normative Analysis Program) is designed to provide the user with a flexible input format capability, a summary output of computed normative values, more direct control over the values plotted on ternary diagrams, and a flexible x-y plot capability. The original program was

---

1/ U.S. Geological Survey, Reston, Virginia, USA

2/ Directorate General of Mineral Resources, Jiddah, Kingdom of Saudi Arabia.

3/ Woods Hole Oceanographic Institution, Woods Hole, Mass.

published by the USGS as Program 542 (Bowen, 1971) in its Computer Contribution Series. In 1977, a version of this program was adapted by Lester North to run on the PDP-11/45 machine in the computer center of the Directorate General of Mineral Resources (DGMR). The version described here is a modification of this program to make full use of the DGMR's mini-computer. This document follows closely the format of the original write-up (Bowen, 1971).

GNAP accepts a group of sample-analysis cards giving weight percents of oxides. From these values the normative mineral values are determined. These normative mineral values and the original oxide values are stored (i.e. saved). They may then be referenced by the user to create plots and diagrams or to define new values that may then be used in the specification of plots and diagrams.

The standard-analysis card format is shown on page 10; if this format is used, then no format specification is necessary. However, the user may specify his own input format (see p. 12) for the oxide values. For example, he may find it more convenient to use an analysis-card format that can be directly used in other programs involving petrological-chemical calculations or plots.

The CIPW norm may be calculated when the rock analysis cards are read or the oxide values may simply be stored (within that run) for norm calculations later in the program sequence. Such storage allows the user to manipulate or adjust oxide values prior to the norm calculation. The norm

calculation results in a horizontal listing of the input and computed values for one sample per printout page unless suppressed by the use of a 'nonorm' command (see p. 15).

The summary output option available in GNAP can produce a table of input oxide data (original and/or adjusted) and of calculated normative and/or chemical parameters specified by the user. The individual samples and their associated values (12 samples per page) appear across the printout page in tabular form. This type of tabular output permits easy comparisons of values between samples and, with little or no additional modification, direct use in manuscripts.

GNAP requires the user to specify which ternary diagrams are desired. This specification is accomplished by giving the complete formulation of the expressions to be used for the three apices of each ternary diagram. These formulations are expressed in terms of defined variables and common arithmetic expressions that the user supplies. These expressions are evaluated by GNAP, and the corresponding diagram is constructed.

In a similar fashion, GNAP provides x-y plot capability. The user supplies arithmetic expressions, formulated from the input oxide data and/or computed normative values, that define the abscissa and ordinate. GNAP evaluates these expressions and constructs the corresponding plots.

In addition to the above plotting capabilities, the user may supply arithmetic expressions that he would like evaluated and printed. GNAP evaluates the expression for each sample

data set in storage and prints the sample identification and expression value.

Values are normally stored as weight percents. It is often desirable to be able to have values available in molecular amounts. GNAP provides the capability of converting from weight percents to molecular amounts.

Plotting may take place on either the printed output or a Calcomp plotter of the digital-drum type. If the plotter is specified, the user may set a scale factor that determines the size of the plot generated. If no plotter specification is made, the lineprinter is used and the plot occupies a full printed page of output.

If desired, a problem title is printed at the top of each page of printout.

This report is a product of a cooperative program conducted by the Saudi Arabian Directorate General of Mineral Resources and the U.S. Geological Survey.

PROGRAM DESCRIPTION

GNAP is composed of a main driver, 13 subroutines, and the Calcomp software package. All coding is in PDP-11 Fortran IV-Plus with the exception of parts of the Calcomp package, which are in assembler language.

The main driving routine scans the input for recognizable commands prior to calls to the other subroutines. Statements are constructed from the input stream; blanks are ignored. The statements are then identified and the indicated action is performed by the appropriate subroutine.

Subroutine NORM, which is used to calculate the CIPW norm (Washington, 1917), contains two additional entries.

The first entry (RECALC) is used to recalculate a norm from previously stored oxide values. The other (CONVER) is used to convert weight percents to molecular amounts.

Subroutine EVAL is used for the evaluation of all arithmetic expressions. EVAL is a Fortran version of a procedure previously developed and documented (Bowen, 1969). A transition-matrix technique is used to parse the expression to be evaluated. This yields the Reverse Polish form of the expression, which is then evaluated using a push-down stack. EVAL has the advantage of not requiring actual comparisons to accomplish the parse; hence execution time is considerably improved over procedures using a more brute-force technique.

Subroutine SUMPNT is used to create the summary printout. An area of output is constructed according to instructions supplied by the user. After this area has been constructed, the summary is printed.

Subroutine PRNT is used for creation of x-y plots. A standard grid is determined from the range of values to be plotted. Subroutine TRIANG is used for the construction of ternary diagrams. If the Calcomp plotter is specified, then calls will be made to the Calcomp software package to generate the plotter commands; otherwise a printer plot is drawn.

Subroutine PRPLOT is a slightly modified version of a subroutine developed at the University of Michigan (Smidinger, 1966). PRPLOT has entries PLOT2, PLOT3, and PLOT4. PRPLOT is used to create printer plots.

Subroutine SIDE is used to construct one side of a ternary diagram when the Calcomp plotter has been specified. It is called from the TRIANG subroutine.

The remaining subroutines provide a degree of character manipulation and conversion in Fortran. MOVE is used to move characters from one string to another. CONV converts from character form to numeric form. INDEX determines the position of a given character in a given string. NOTEQ determines if two strings are equal (i.e., contain the same characters). CLEAR provides for the construction of pages of output prior to printing. PREPRO sets up the operating environment for the program on the DGMR's PDP-11 computer.

INPUT

This program has two general types of input. The title card and analysis cards are punched and read in a fixed format where blanks are read as blanks in the title card and interpreted as a 'Ø' (zero) in the analysis cards. All other input cards are free-form, i.e., the cards are punched beginning in any column, blanks are ignored, and the end of input is indicated by a semicolon.

Title card

A title card is identified by the word TITLE punched in columns 1-5. Any alphanumeric information in columns 6-80 that identifies problems is then printed on each succeeding page of output (excluding pages on which error messages are printed). The title occupies one card and is usually the

first card in the input deck. More than one title card may be used in a problem if desired. For example, if it is desired that the summary of oxides and normative parameters have one title and the plots have a different title (perhaps one in which symbols are identified), then the appropriate title card should be placed just before the command cards controlling the output to which the title is to apply.

Analysis card

Rock analysis cards contain values of the standard set of oxides. These cards should contain the letters NRM punched in columns 1-3 if the CIPW norm is to be calculated when the card is read. When the norm is calculated, the normative minerals, Niggli values, Barth's cations, as well as various totals and ratios, are printed and stored. If the user wishes to store the oxide values without calculation of the norm, the letters STO should be punched in columns 1-3. Column 4 should contain the character that is to be used when plotting values from this sample. Any printable character may be used for plotting, but the user is advised not to use the characters "+" or "-" because these characters are used to demarcate plot outlines and in the subdivisions of units of axes in x-y plots or sides of ternary plots and thus may result in possible confusion. Columns 5-10 should contain the six-character sample identifier.

In addition analysis card data may be input interactively by means of the NRM pseudo-arguement PROMPT (see below).

MODES OF OPERATION

The DGMR/USGS version of GNAP operates in one of two modes:

BATCH mode: All input to the program, commands, analytical data, and titles comes from a file or from cards. This mode is equivalent to off-line operation.

TERMINAL mode: Interactive operation, commands, titles, and optional analysis data all come from the console. Analysis data may be input interactively from a file or cards or typed in as card images at the console.

When the program is started, the mode is selected by request from the keyboard (see Running GNAP p. 30).

INTERACTIVE OPERATION

In terminal mode, output may be directed to the printer or to the console; output for the graph plotter is always sent to the disc. Error messages always come back to the console as does output from the PRINT command.

Analysis data may be input in one of three ways:

1. Typed in the usual way for analysis cards (see p. 10).
2. From an external file, or device, in which case the file should contain the usual analysis card images. This method is invoked by the pseudo-arguement BATCH in columns 5-9, followed by the file title in column 11-42. Columns 1-3 should contain either NRM or STO.

For example,

NRM BATCH <file title>

cards may be read from the card reader by;

NRM BATCH CR:

Note that no semi-colon should be placed at the end of the 'command'. The program will read records from the named file until an end of file marker is encountered. If no file title is given, the program will attempt to open a default file named GNAP.DAT.

3. Interactive input. In this method the program prompts for oxide values at the console from the current oxide list (see p. 15). This method is invoked by the pseudo-arguement PROMPT in columns 12-17. Columns 1-10 should contain "NRM" or "STO" in columns 1-3, plotting symbol in column 4, and sample identifier or laboratory number in columns 5-10, in the usual manner--for example:

NRM$AAA123 PROMPT

The program will then prompt for oxide values, one per line, which should be given as four digit numbers representing hundredths of a percent, with no decimal points (see table 1).

Each weight percent must be given to the hundredth without a decimal point in the right most portion of the field (i.e. 12.01 would be given as 1201 and 0.02 would be given as 2 in the fourth column of the field and the leading zeros would be optional).

Table 1.--*Standard format for DGMR/USGS version of GNAP*

Columns	Field width	Contents
1-3	3	NRM or STO
4	1	plotting symbol
5-10	6	sample identifier/lab. number
11-14	4	weight percent SiO_2
15-18	4	weight percent Al_2O_3
19-22	4	weight percent Fe_2O_3
23-26	4	weight percent FeO
27-30	4	weight percent MgO
31-34	4	weight percent CaO
35-38	4	weight percent Na_2O
39-42	4	weight percent K_2O
43-46	4	weight percent H_2O
47-49	3	weight percent TiO_2
50-52	3	weight percent P_2O_5
53-56	4	weight percent MnO
57-60	3	weight percent ZrO_2
61-63	4	weight percent CO_2
64-66	3	weight percent SO_3
67-69	3	weight percent Cl
70-72	3	weight percent F
73-75	3	weight percent S
76-78	3	weight percent Cr_2O_3
79-80	2	weight percent NiO

This standard format may be overridden by the user (see p. 12). However, columns 1-10 must remain unchanged as described above.

COMMANDS

Command cards are free-form as indicated above (p. 6). Each command given must end with a semicolon (";") and may occupy more than one card if necessary. The purpose of this second type of input is to instruct the computer as to which of the various functions it is to perform. The commands recognized in the program are summarized in table 2 (p. 23). In order to describe some of the commands that GNAP recognizes, it will be necessary to discuss the term "expression".

Expressions

Expressions, which are basic to any algorithmic process, are rules to obtain values of different kinds and types. An expression is composed of variable names, arithmetic operators, constants, and the grouping symbols ( ).

Variable names may be any of the basic names that have been associated with the values by calculation of the norm (see table 3, p. 24 for a list of these names). Additional variable names may have been determined by the "define command", which will be discussed later (see p. 16).

The arithmetic operators are the negative operator (e.g. SIO2), division (e.g. SIO2/FEO), multiplication (e.g. SIO2*FEO), subtraction (e.g. SIO2-FEO), and addition (e.g. SIO2+FEO). These operators appear above in the order of their priority.

In other words, the negation operation is performed before division, division before multiplication, multiplication before subtraction, and subtraction before addition. This order may be overridden by the use of parenthesis. Hence, A/B*C and A/(B*C) usually yield different values.

Constants are decimal numbers with or without a decimal point. Constants must be eight characters or less in length including the decimal point.

Expressions must be syntactically and semantically correct in the Fortran or PL/1 sense. This means that for each left parenthesis there must be a matching right parenthesis; no two arithmetic operators may appear immediately adjacent to one another and the variable names must be "known" to the computer when the expression is evaluated.

Modify-format command

The modify command is used to alter the standard analysis-card format described on page 10. This command takes the form MODIFY FORMAT = <format specifications> [,NCS=n]; where <format specifications> represents a Fortran IV object-time format and where the NCS parameter (number of cards per sample) is specified by n, an integer 1, 2, or 3. If n = 1, the NCS parameter is optional.

The format must specify 80 card columns, the first ten of which are specified by an X format code. This means that the format specification takes the restricted form FORMAT= (1ØX,...) where the user supplies format items describing card columns 11-80 denoted above by .... The order in which

the oxide values are given on the standard analysis card is not overridden when the user supplies his own format unless he follows the MODIFY FORMAT command with an OXIDES command (see below). As an example, suppose the user has five oxide values per card, starting in column 11 of each card, where the oxide values are located in fields of seven columns, and are given to two decimal places. Each of three cards contains in columns 1-3 either NRM or STO, the plotting symbol in column 4, and the sample identified in columns 5-10. The modify format command would be given as

```
MODIFY FORMAT (1ØX, 5F7.2), NCS = 3;
```

If the user wished to restore the standard-analysis format, he would command:

```
MODIFY FORMAT (1ØX, 9F4.2, 2F3.2, F4.2, F3.2, F4.2,
     5F3.2, F2.2), NCS = 1;
```

RASS and STATPAC modifications

Data from STATPAC/RASS files may be read into GNAP by specifying the pseudo-format arguement 'STATPAC';

```
MODIFY FORMAT (STATPAC);
```

The oxide command should be given before this command if the oxides differ from those in the standard format. GNAP input processing will now change phase; oxide values will be read in, 10 to an 80 column record. As many records as are needed will be input for each sample to satisfy the oxide list; so for 12 oxides, two records will be read. No NRM or STO commands will be necessary, and sample identifiers or laboratory numbers will be read from the first card for each sample

in columns 71-76. Plotting symbols, if they are required to be different for succeeding samples, must be specified on separate cards, which must precede the analysis cards to which they relate. The command that changes the plotting symbol is:

SYMBOL $;

where $, for example, represents the new plotting symbol. If a file of analysis records is generated by STATPAC/RASS, then in general the plotting symbol will be the same for each sample, and can be specified by the symbol command before the records are read in. If no symbol command is given, the default plotting symbol is "X".

After the last STATPAC/RASS record is read in, it is necessary to tell the program to change input phase back to the original form, a change accomplished by the last command:

LAST;

GNAP now reverts to the standard input format for subsequent analyses.

During STATPAC/RASS processing, norms are computed automatically for each sample. The FORTRAN format control under which data are read in is:

(1Ø (1X, G6.Ø), 1ØX)

Oxide command

The oxide command permits the user to specify which of the 21 possible oxides will be read when using the norm or store commands. The oxides for each sample are read in the order specified by the oxide command. If the oxide command

is not given, 20 oxides will be read in the order given in table 1, page 10. The oxide command must precede the analysis cards to which it refers. In most instances the oxide command will be used in conjunction with the modify command. The oxide command takes the form OXIDES <list of selected oxides>; where <list of selected oxides> contains the oxide names, separated by commas, which will be read by future norm or store commands (see "Analysis Card", p. 7). The oxide specifications for the Standard Format above would be as follows:

OXIDES SIO2, AL2O3, FE2O3, FEO, MGO, CAO, NA2O,
K2O, H2O, TIO2, P2O5, MNO, ZRO2, CO2,
SO3, CL, F, S, CR2O3, NIO;

An error in the oxide command specification yields a suitable message and the run is terminated.

Nonorms command

Whenever the norm command (see p. 7) or recalculate command (see p. 19) is issued, a page of printout per sample is generated by the program (see sample output p. 45-55). This printout can be suppressed by use of the nonorm command. The nonorm command takes the form NONORMS;. Once this command is issued, there is no way of reverting it for the remainder of that problem. It should be noted that the computation of norms is not effected by this command. It only suppresses printout.

Define command

The define command is used to define new variable names or redefine basic names (see p. 24). The define command takes the form <variable name> = <expression>; <variable name> is any string of letters and digits eight characters or less in length and whose first character is a letter (e.g., X, Y, X123, XYZ, etc.). <expression> is any valid arithmetic expression as discussed in the previous section. As an example:

CAO=CAO-(3*P2O5+CO2); redefines CaO as the original CaO minus the sum of CO_2 and $3P_2O_5$, and CI=AN+2.157003*DIEN+(FO+0.700837*HYEN)+MT+CM; defines the new variable CI to be equal to $An+2.157003En_{di}+(Fo+0.700837En_{hy})+mt+cm$.

Plot command

The plot command instructs the computer to create an x-y plot. The plot command takes the form PLOT <expression1>, <expression2>; or PLOT(R) <expression1>, <expression2> where <expression1> and <expression2> are valid arithmetic expressions as previously described and give respectively the abscissa and ordinate of the desired plot. As an example:

PLOT SIO2, (FEO+FE2O3)/(FEO+FE2O3+MGO); instructs the computer to plot SiO_2 versus $\frac{FeO+Fe_2O_3}{FeO+Fe_2O_3+MgO}$ with the abscissa (i.e. SiO_2) increasing from left to right. With the plot command, the abscissa values increase from left

to right; with the PLOT(R) command, abscissa values increase from right to left. The computer uses all values stored by calculation of the norm since the last (if there was one) clear command.

It is important to note that, unless otherwise specified by means of define commands, plots of expressions involving oxide values are based on the original oxide values. If plots of expressions of adjusted oxide values are desired, the user must so stipulate by means of appropriate commands. Plots of normative parameters of expressions, however, calculated or recalculated from adjusted oxide values, can be obtained by simply using the basic names of normative parameters (see table 3).

Ternary command

The ternary command instructs the computer to plot a ternary diagram. This command takes the form TERNARY <expression1>, <expression2>, <expression3>; where <expression1>, <expression2>, and <expression3> are valid arithmetic expressions as previously described and give respectively the topmost, lower left, and lower right apices of the ternary diagram to be plotted.

As an example: TERNARY OR, AB, AN; instructs the computer to plot the or:ab:an ternary diagram and TERNARY Q,OR, AB+AN; instructs the computer to plot the q:or:ab+an ternary diagram. The computer uses all values stored by the norm

calculation since the last (if there was one) clear card. As in the case of the PLOT command, any of the expressions may contain references to variables that have been defined.

Convert-values command

All values that have been stored are in weight percents, and it might become necessary to convert these values to molecular amounts. Conversion is accomplished by the convert command, which takes the form CONVERT VALUES;. Only the basic values stored by the norm calculation are converted. Hence, if the user wishes to define or redefine variable names in terms of molecular amounts, the convert command must precede these define commands. Norms will now be displayed in molecular amounts.

Print command

The print command instructs the computer to evaluate and print one or more expressions. The print command takes the form PRINT <expression> [,<expression>, <expression>,...]; where the brackets indicate that the user may optionally add more expressions to be operated upon. Again the expressions referred to above must be valid arithmetic expressions. As an example: PRINT (FEO+FE2O3)/(FEO+FE2O3+MGO), FEO+0.9*FE2O3; instructs the computer to evaluate and print the results of the two indicated expressions for each sample in storage. The print commands also utilize only the original oxide values, unless specified otherwise by means of define commands.

When in terminal mode, the output is always sent to the console, not the printer.

Recalculate-norms command

The recalculate command is used to calculate (or re-calculate) the norm for each sample in storage. This command takes the form RECALCULATE NORMS;. The recalculate command is used if the user has modified the original oxide values by the use of the define command and wishes to obtain the normative values based on this "adjusted" analysis. This command should not be used after a convert command because the norm calculation is based on oxides given in weight percents and not molecular amounts.

Graphs command

This command is used to specify whether plotting is to be done on the lineprinter or the Calcomp plotter. If none is given or if the graphs command is incorrectly given, the program uses the lineprinter for plotting. Both devices may be used within a run. All plot or ternary commands that precede the graphs command in the input stream specifying the Calcomp plotter will produce lineprinter plots. This command takes the form GRAPHS=CALCOMP; or GRAPHS=PRINTER;.

Scale command

All lineprinter plots are fixed in scale so that they occupy a full printed page.

For the x-y plots, the lineprinter divides the abscissa into 10 equal intervals, and the ordinate into five equal

intervals, based on the maximum and minimum values stored for each coordinate. In general this will produce a set of plots of widely different absolute scales; e.g., 1 inch on the abscissa or ordinate may be equivalent to 0.43 percent SiO_2, 0.15 percent Al_2O_3, 0.06 MgO/MgO+FeO, and so forth. It is often desirable that the scaling be consistent among plots of a single data set or for similar plots of different data sets so that direct comparison (say, by superposition over a light table) is possible. The user may achieve this by inserting two dummy data cards in the input deck. STO should be punched in columns 1-3, column 4 left blank (i.e. no plotting symbol), and the desired maximum and minimum values for each oxide punched in the appropriate fields. For example, if SiO_2 in the data cards ranges from 43.2 to 49.6, the lineprinter scale will be divided into intervals of $\frac{49.6-43.2}{10}$ or .64. Dummy cards punched with SiO_2=40.0 and SiO_2=50.0 would change the interval to 1.0.

When the Calcomp plotter has been specified, the abscissa is divided into 10 equal intervals and the ordinate into eight equal intervals. The numeric values of the intervals are determined by the Calcomp scaling subroutine that yields interval values of 1, 2, 4, 5, or 8 x 10^n. As for lineprinter plots, dummy data cards may be used to insure consistent scaling. In addition the absolute size of the plots may be varied using the scale command. If the scale command is not given, the ternary diagrams will have sides that are 9.08 inches in length. This figure allows direct overlay on

standard Keuffel and Esser triangular coordinate paper (K and E No. 464490). Two-dimensional x-y plots will have an abscissa of 10.0 inches and an ordinate of 8.0 inches. These dimensions are equated to a scale factor of 1.0. If the user wishes to modify this scale factor, he may do so with the scale command, and all plotter pen movement will be magnified by the specified scale factor. This command takes the form SCALE=n; where $0.0<n\leq 3.0$. Specifying a scale factor greater than 3.0 exceeds physical limits imposed by the 30-inch Calcomp plotter.

Summary command

The summary command is used to produce a summary printout of the samples in storage. Items in the summary printout are subdivided into nine groups. These groups are as follows:

1. Original oxides as punched on the analysis cards
2. Original oxides normalized to 100 percent

   Note: Normalization is automatically done before the calculation of normative parameters. If the user changes the original oxide values by means of define commands and then recalculates the norm using the recalculate command, then any subsequent summary command will print these new values
3. Normative minerals
4. Partitioning of normative di, hy, and ol
5. Barth's cations
6. Niggli values
7. Thornton and Tuttle's differentiation index

8. Ratios Al_2O_3/SiO_2 and FeO/Fe_2O_3

9. User-defined values

Any combination of these groups in any order may be specified in the summary command. These groups are denoted respectively by the keywords OXIDES, ADJUSTED, MINERALS, PARTITIONS, BARTH, NIGGLI, D.I.[1/] RATIOS, and USER. It should be noted that only those variables names that are unique (i.e. not in the list of basic names on page 24) can be printed in the user-defined values group. The summary command takes the form SUMMARY (<list of group keywords>); where each item in the <list of group keywords> is separated by a comma. As an example: SUMMARY OXIDES, MINERALS); will generate a summary printout of the oxides and normative minerals for each sample in storage. The summary output will also list the plotting symbol that will be used for each sample in ternary or x-y plots.

Clear command

The clear command is used to clear all previously stored values. This command would typically be used when plotting for one set of samples has been completed and the user wishes to start calculating and storing values for another set of plots. This command takes the form CLEAR STORAGE; (table 2).

[1/] D.I. may only be used in the summary command. To plot D.I. the user must specify the actual expression for the Thornton-Tuttle differentiation index, i.e., Q+OR+AB+NE+KP+LC.

Table 2.--*Summary of commands*

Command	Syntax
Clear	CLEAR STORAGE;
Convert	CONVERT VALUES;
Define	<variable name> = <expression>;
Graphs	GRAPHS=CALCOMP/PRINTER;
Modify	MODIFY FORMAT = <format specifications> [,NCS=n];
Nonorms	NONORMS;
Oxides	OXIDES <list of selected oxides>; where each oxide name is one of the standard oxides given on page 10 and the names separated by commas.
Plot	PLOT [(R)] <expression1>,<expression2>;
Print	PRINT <expression> [,<expression>, <expresion>...];
Scale	SCALE = <decimal number>;
Summary	SUMMARY <group keyword> [,<group keyword>,...]; where; <group keyword>: = OXIDES/RATIOS/MINERALS/ PARTITIONS/D.I./BARTHS/NIGGLI/ADJUSTED/USER
Ternary	TERNARY <expression1>, <expression2>, <expression3> ;
Commands for STATPAC/RASS processing	
Last	LAST;
Modify	MODIFY FORMAT (STATPAC);
Symbol	SYMBOL <print character>;

Note: The underlined portions of the commands indicate which characters the program will examine to recognize the command. Symbols used in the syntax have the following meanings:

A:=B	A is defined as B
n	an integer number
A/B	select A or B
[A]	A is optional

Table 3.--*Names in the GNAP dictionary. These names may be used in expressions, and for plotting or printing.*

Oxides		Minerals				Barth's cations	
SIO2	silica	Q	quartz	CM	chromite	#SI	silicon
AL2O3	alumina	C	corundum	HM	hematite	#AL	aluminium
FE2O3	ferric	Z	zircon	IL	ilmenite	#FE3	ferric
FEO	ferrous	OR	orthoclase	TN	titanite	#FE2	ferrous
MGO	magnesium	AB	albite	PF	perofskite	#MG	magnesium
CAO	calcium	AN	anorthite	RU	rutile	#CA	calcium
NA2O	sodium	LC	leucite	AP	apatite	#NA	sodium
K2O	potassium	NE	nepheline	FR	fluorite	#K	potassium
H2O	water	KP	kaliophilite	PR	pyrite	#H	hydrogen
TIO2	titanium	HL	halite	CC	calcite	#TI	titanium
P2O5	phosphorus	TH	thenardite	MG	MGO/FM[1/]	#P	phosphorus
MNO	manganese	NC	sodium carbonate	DI	diopside	#MN	manganese
ZRO2	zirconium	AC	acmite	DIWO	WO content of DI	#ZR	zirconium
CO2	carbon dioxide	NS	sodium metasilicate	DIEN	EN content of DI	#C	carbon
SO3	sulphate	KS	potassium metasilicate	DIFS	FG content of DI	#S1	sulphate
CL	chlorite	WO	wollastonite	HY	hypersthene	#S2	sulphur
F	fluorine	EN	enstatite	HYEN	EN content of HY	#CL	chlorine
S	sulphur	FS	ferrosilite	HYFS	HY content of HY	#F	fluorine
CR2O3	chromium	FO	forsterite	OL	olivine	#CR	chromium
NIO	nickel	FA	fayalite	OLFO	FO content of OL	#NI	nickel
BAO	barium	CS	calcium silicates	OLFA	FA content of OL	#BA	barium
		MT	magnetite	WOL	WO minus DIWO		

[1/] $FM = 2Fe_2O_3 + FeO + MgO + MnO + NiO$

OUTPUT

Each CIPW norm calculation creates a page of information. This page gives the weight percent of each oxide, their total, weight percents and molecular amounts of the normalized oxides and normative minerals, Thornton's differentiation index, the ratios Al_2O_3/SiO_2 and the Niggli values.

If the ternary command is issued, GNAP prints the ratio values along with the sample identification for each point on one page. The ternary diagram itself is either plotted on an additional page or on the Calcomp plotter if that was selected. The apices are annotated with the expressions used to define them.

For each print command issued, GNAP gives a page of output containing sample identifications and an expression value for each sample stored. If more than one expression was specified, a new page is printed for each additional expression.

Each define command is printed when it is executed. The first define command is printed at the beginning of a page. If subsequent define commands are given with no intervening commands of other types, they will be printed below the first define command.

For the summary output all values that have been stored by the norm calculation are subdivided into groups of 12 samples. Each of these groups is printed as a unit. The sample identifications are printed across the top of the page. Row identification is printed along the left-hand side of the page, so that each value in the summary table is identified.

Zero values in the summary table will be left blank. Any item other than ratios and the differentiation index that has zero values for each sample on a page will be omitted from the printout.

Input and output from a sample run of two problems is contained in Appendix B.

PROGRAM RESTRICTIONS

1. All names of variables must begin with a letter and be no more than eight characters in length.
2. A maximum of 14 new names of variables may be defined.
3. A maximum of 20 norm values may be stored without the use of the clear storage command.
4. Arithmetic expressions must be 40 characters or less in length, not including blanks.
5. Expressions that appear in the plot and ternary commands must be separated by commas.
6. All commands must end with a semicolon.
7. The define command has effect only on the values stored at the time the define command is executed.
8. Only analysis cards may contain NRM or STO in columns 1-3.
9. Only title cards may contain TITLE in columns 1-5.
10. All commands may contain a maximum of 120 non-blank characters.
11. The format specified in the modify command may be a maximum of 96 characters.

12. A maximum of three cards may be used to give the original weight-percent oxides for one analysis.

13. The optional specification of a scale factor for the Calcomp plotter may not exceed 3.0.

Note: The maximum number of norms that may be stored will be increased to an indefinite number in the next version of the program, but only 200 will be available for plotting.

TIMING AND STORAGE REQUIREMENTS

In interactive mode, timing depends on input from the console; when the PDP-11 is not busy, all commands will be executed, at most, in a few seconds. In batch mode, timing will depend on the number of commands; the example in Appendix B runs in 35 seconds elapse time with a quiescent system.

The program occupies a full page of the PDP-11's memory, 64 K bytes. It is more efficient of the machine's resources to run under batch mode.

ERROR MESSAGES

GNAP was designed to provide error recovery from those errors that were due to incorrect or inadequate input. When an error is detached, a message describing the error is printed and the program resumes scanning the input for commands to execute. A list of the error messages and their probable cause follows:

1. **NO MORE THAN 20 NORMS MAY BE STORED. THE LAST NORM WILL BE ERASED**

   More than 20 analysis cards have been processed without an intervening clear command.

2. NO MORE THAN 14 NAMES MAY BE DEFINED. DEFINITION IGNORED FOR NAME=<name>.

   More than 14 new variable names have been defined. Consider redefining existing names.

3. PLOT COMMAND ERROR ON <card>.

   The two expressions giving, respectively, the abscissa and ordinate of the desired plot were not separated by a comma.

4. TERNARY COMMAND ERROR ON <card>.

   The expressions giving the apices of the desired ternary diagram were not separated by commas.

5. SCALE MUST BE POSITIVE AND LESS THAN OR EQUAL TO 3.0.

   This message is self-explanatory.

6. "GRAPHS=CALCOMP" MUST BE SPECIFIED BEFORE SETTING SCALE.

   This restriction is imposed by the Calcomp software package.

7. STATEMENT LENGTH EXCEEDED ON <card>. DID YOU FORGET A SEMICOLON?

   This message might be caused by a missing semicolon or by an error in a user-supplied format.

8. FORMAT ERROR (MISSING PARENTHESIS) IN <card>.

   A user-supplied format contains a missing parenthesis.

9. ERROR IN MODIFY COMMAND ON <card>.

   The modify keyword was not followed by FORMAT or NCS.

10. UNRECOGNIZED COMMAND IN <card>.

    The computer identified an illegal command on the indicated card. This could be caused by a misspelling or an error in a user-supplied format.

11. NORM NOT COMPUTABLE. SEE ERROR CODE n OF PROGRAM WRITE-UP.

    This indicates that the norm was not computable due to one of the following causes:

n	Cause
2	Insufficient CaO to form AP
3	Insufficient Na_2O to form HL
5	Insufficient FeO to form PR
6	Cr_2O_3 in excess of FeO
9	Insufficient CaO to form FR
11	Insufficient MgO to form EN after CO_2 subtracted
12	Insufficient SiO_2 to form Z
31	Improper amount of SiO_2 to form KP or LC

12. ERROR IN EXPRESSION <expression>.

    This message is generally caused by an expression syntactically incorrect because of unmatched parenthesis, adjacent arithmetic operators, and so forth.

13. UNDEFINED NAME <name>.

    An undefined name appeared in an expression. The most probable cause would be misspelling.

14. FIRST WORD ILLEGAL IN <keyword list>.

    A group keyword is misspelled in the summary command.

15. THE FOLLOWING COMMAND CONTAINS EXCESSIVE CHARACTERS.«
    DID YOU FORGET A SEMICOLON? <command>.

    The length of the command is greater than the form given in the write-up.

16. OXIDE COMMAND CONTAINS A NAME WHICH IS NOT IN THE LIST OF ACCEPTABLE OXIDES. ACCEPTABLE OXIDES ARE: <list of acceptable oxide names>.

    An oxide name is misspelled in the oxide command.

17. PROGRAM GNAP NO TITLE SUPPLIED.

    This statement will be printed on each page of output if the input deck contains no title card(s).

RUNNING GNAP

Whether GNAP is run in batch mode or terminal mode, the program is always invoked at a console. If cards are punched for batch mode and handed in to the Computer Center, the user need not add any system-command cards to the deck. To run GNAP in batch mode the following procedure must be taken:

The user logs onto the on-line system by means of his user identifier, which will be provided by the center, and issues commands:

```
CRTL>∿C                    Control ∿ C
MCR>HEL [n,m]              where n,m is the user identifier
MCR>RUN $GNAP <escape>
GNAP---BATCH OR TERMINAL?------->  BATCH
```

NAME BATCH FILE:----------> CR:

BATCH PROCESSOR TAKES CONTROL

GNAP --- STOP

The underlined text is issued by the system or the program; the rest is user responses. If a deck has been filed on disc, the batch file would look something like this:

NAME BATCH FILE----------> WADI. DAT

where WADI. DAT is the file containing the GNAP commands and analysis cards.

Output will be sent to the printer when the job is finished. If any output is directed to the plotter, the following commands are necessary to generate the plot. The user must make sure that the graph plotter is attached to the system beforehand. After:

GNAP-----STOP

<CRTL∿C To return to command

MCR>FLX XY:/RS = GNAP.PLT/RS/FB:256.

MCR>BYE<escape>

The job is now complete and the user is logged off from the system.

To run GNAP interactively, the following sequence of commands is typed:

CRTL∿C to prompt the system

MCR>HEL[n,m] n,m is user id

MCR RUN $GNAP <escape>

GNAP---BATCH OR TERMINAL?-------> TERMINAL

OUTPUT TO TERMINAL OR PRINTER?-----> TERMINAL

**GNAP><commands>

GNAP issues the prompt "**GNAP" and is ready to receive commands as explained in the text. Output may be directed to the terminal or to the printer. If it is sent to the terminal, the user should use a teletype console, which prints the responses from the program, and should ensure that the terminal is set to a 132-character line width. If output is directed to the printer, all results are temporarily stored until the program exits and are then printed. Some output is always sent to the terminal, however--for example, print commands, error messages, and echoing of definitions. Output to the terminal is printed as it is generated by GNAP. Examples of interactive processing are shown in Appendix A, and an example of batch operation with output in Appendix B.

REFERENCES

Al Shanti, Ahmad M., 1974, Al Ji'lani layered basic intrusion Ad Dawadimi District, Kingdom of Saudi Arabia: Saudi Arabian Dir. Gen. Mineral Resources Bull. 12, 45 p.

_____ 1976, Geology of Ad Dawadimi District, Kingdom of Saudi Arabia: Saudi Arabian Dir. Gen. Mineral Resources, Bull. 13, 57 p.

Bowen, R.W., 1967, General rock norm program, Program M0016: U.S. Geological Survey Computer Program Documentation (unpublished).

Bowen, R.W., 1968, Molecular norms, Program M0015: U.S. Geol. Survey, Computer Program Documentation (unpublished).

_____ 1969, Subroutine EVAL, Program C130: U.S. Geol. Survey Computer Program Documentation (unpublished).

_____ 1970, Graphic normative analysis, Program C405: U.S. Geol. Survey Computer Program Documentation (unpublished).

_____ 1971, Graphic normative analysis program, Program C542: U.S. Geol. Survey Computer Program Documentation (unpublished).

DOherty, P.C., and Wright, T.L., 1971, Mineral distribution program--mod II, Program C129: U.S. Geol. Survey Computer Contribution Number 7.

Eicher, R.N., 1968, Amphibolite discriminant, Program W9091: U.S. Geological Survey Computer Program Documentation (unpublished).

Smidinger, P., 1966, PRPLØT, A 360 printer plotting program: Univ. Michigan Computer Program Documentation (unpublished).

Washington, H.S., 1917, Chemical analyses of igneous rocks: U.S. Geol. Survey Prof. Paper 99, 1201 p.

Wright, T.L., 1970, Mineral norms--mod I, Program C463: U.S. Geol. Survey Computer Program Documentation (unpublished).

page 35 follows

APPENDICES TO ACCOMPANY USGS PROJECT REPORT 254

GNAP

(GRAPHIC NORMATIVE ANALYSIS)

page 37 follows

APPENDIX A

Interactive operation

Listing 1.--*Demonstration of interactive operation. Machine prompts are underlined*

```
MCR>HEL [14,14]
MCR>RUN GNAP$
GNAP -- BATCH OR TERMINAL? -----> TERMINAL
OUTPUT TO TERMINAL OR PRINTER? -----> PRINTER
**GNAP>TITLE AD DAWADIMI DISTRICT GRANITES (AA).
**GNAP>OXIDES SIO2,TIO2,AL203,FE203,FEO,MNO,MGO,CAO,NA20,K20,H20,P205;
**GNAP>MODIFY FORMAT (10X,12F4.2);
**GNAP>NONORMS;
**GNAP>NRM BATCH [14,14]DAWA.DAT
**GNAP>NRMUISA16'7130003514930056017600030026000004440503005700005
**GNAP>NRMVISA39+ PROMPT
 SAMPLE ID ISA39;
 SIO2      .......> 7169
 TIO2      .......>    9
 AL203     .......> 1550
 FE203     .......>   53
 FEO       .......>  103
 MNO       .......>    2
 MGO       .......>   27
 CAO       .......>   51
 NA20      .......>  416
 K20       .......>  463
 H20       .-......>  158
 P205      .......>    2
**GNAP>SUMMARY (OXIDES,MINERALS,D.I.,RATIOS);
**GNAP>H20=0;FEO=FEO+FE203*0.9;FE203=0;FEMAG=FEO/(FEO+MGO);


            AD DAWADIMI DISTRICT GRANITES (AA).

H20       DEFINED AS 0

FEO       DEFINED AS FEO+FE203*0.9

FE203     DEFINED AS 0

FEMAG     DEFINED AS FEO/(FEO+MGO)
**GNAP>GRAPHS=CALCOMP;
**GNAP>SCALE=1.0,
**GNAP>TERNARY FEO+MNO,NA20+K20,MGO;
**GNAP>CLEAR STORAGE;
**GNAP>^Z

GNAP  --  STOP

MCR>FLX XY:/RS=GNAP.PLT/RS/FB:256.
MCR>BYE$
```

APPENDIX B

Batch operation

A listing is a series of commands and analysis cards for a batch job; the data for listings 2 and 3 are extracted from DGMR Bulletins 12 (Al Shanti, 1974) and 13 (Al Shanti, 1976). Two GNAP problems are separated by a CLEAR STORAGE command. Title cards are used throughout to label output, which is shown below.

The first problem computes norms for 12 analyses of granitic rocks from the Ad Dawadimi district. The first two norms are read in under the default Standard Format. These two analyses produce a page each for the computed norms and ancillary data. The MODIFY FORMAT and OXIDES commands that follow change the input format printing of norms. Ten further analysis cards follow in the new format. The following commands are then given:

PRINT gives a single printed page evaluating the expression. Note that in terminal mode this evaluation will always come to the terminal.

SUMMARY generates a page of tabulated data for the groups of variable specified; the norms of all analyses are printed.

TERNARY produces a triangular plot of Q, AB, OR on the printer (p. 50).

Two "STO" cards follow; these are dummies that set upper and lower limits for the following plot commands.

PLOT generates a graph on the lineprinter (p. 51).

GRAPHS changes the plotting device to the graph plotter; no response is produced by the program.

SCALE sets the plotting scale to 1.0 times the default scale, which is 10" on the abscissa and 8" on the ordinate. This command produces no response unless there is an error.

PLOT now generates a cartesian graph on the Calcomp plotter (fig. 1).

DEFINE H2O, FEO, and FE2O3 are redefined and a new variable is added to the dictionary FEMAG.

RECALCULATE now recomputes the norms with the new defined variable FEO, FE2O3, and H2O.

SUMMARY tabulates the result for the given group of variables.

PRINT demonstrates the use of Barth's Cations in the dictionary.

CONVERT all oxide values are now converted to molecular amounts; all subsequent displays will use these values.

TERNARY generates an alkali - Fe-Mg diagram on the graph plotter (fig. 2).

CLEAR erases all the stored values, resets the standard input format, and forces GRAPHS=PRINTER. The program is now ready for another set of analyses.

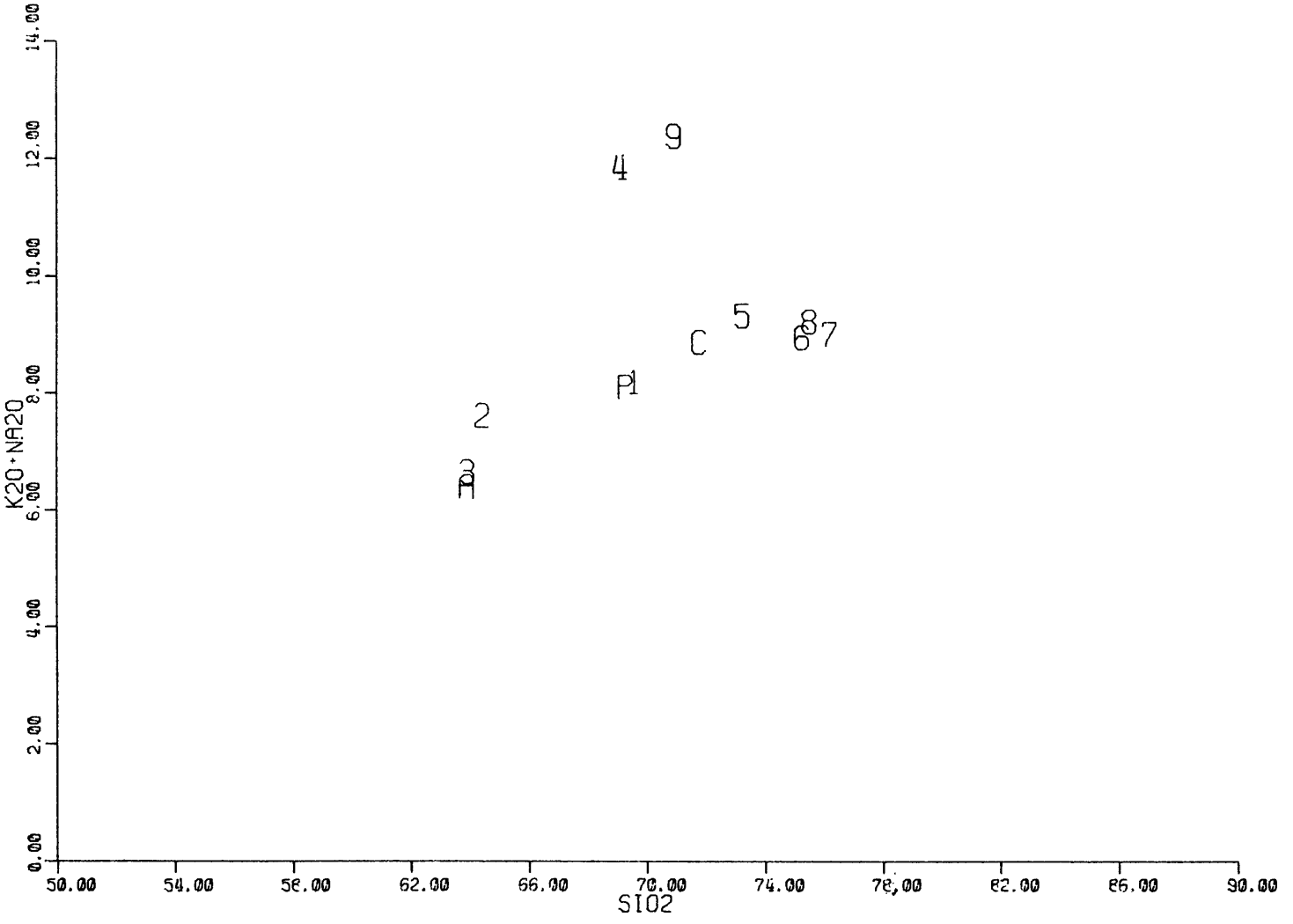


Figure 1. Cartesian graph plot.

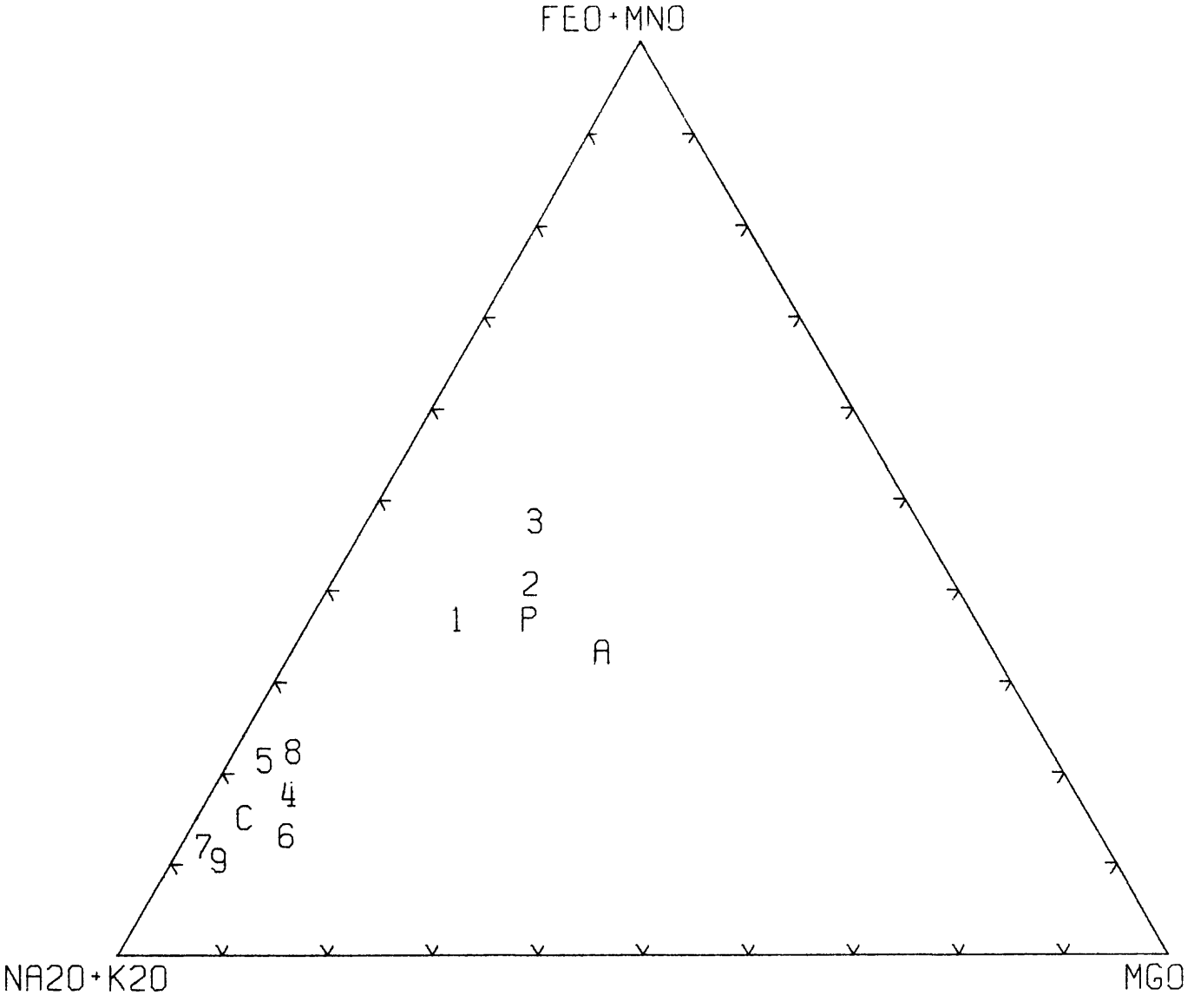


Figure 2. Triangular diagram plot.

The second problem shows how GNAP may deal with elements on other data. After a title card two oxides are selected as dummies. MODIFY FORMAT sets a new input format and 15 analyses cards are read in but no norms are computed because the "STO" option is ANORTH and STRONT, and a reversed graph is plotted on the Calcomp plotter (fig. 3).

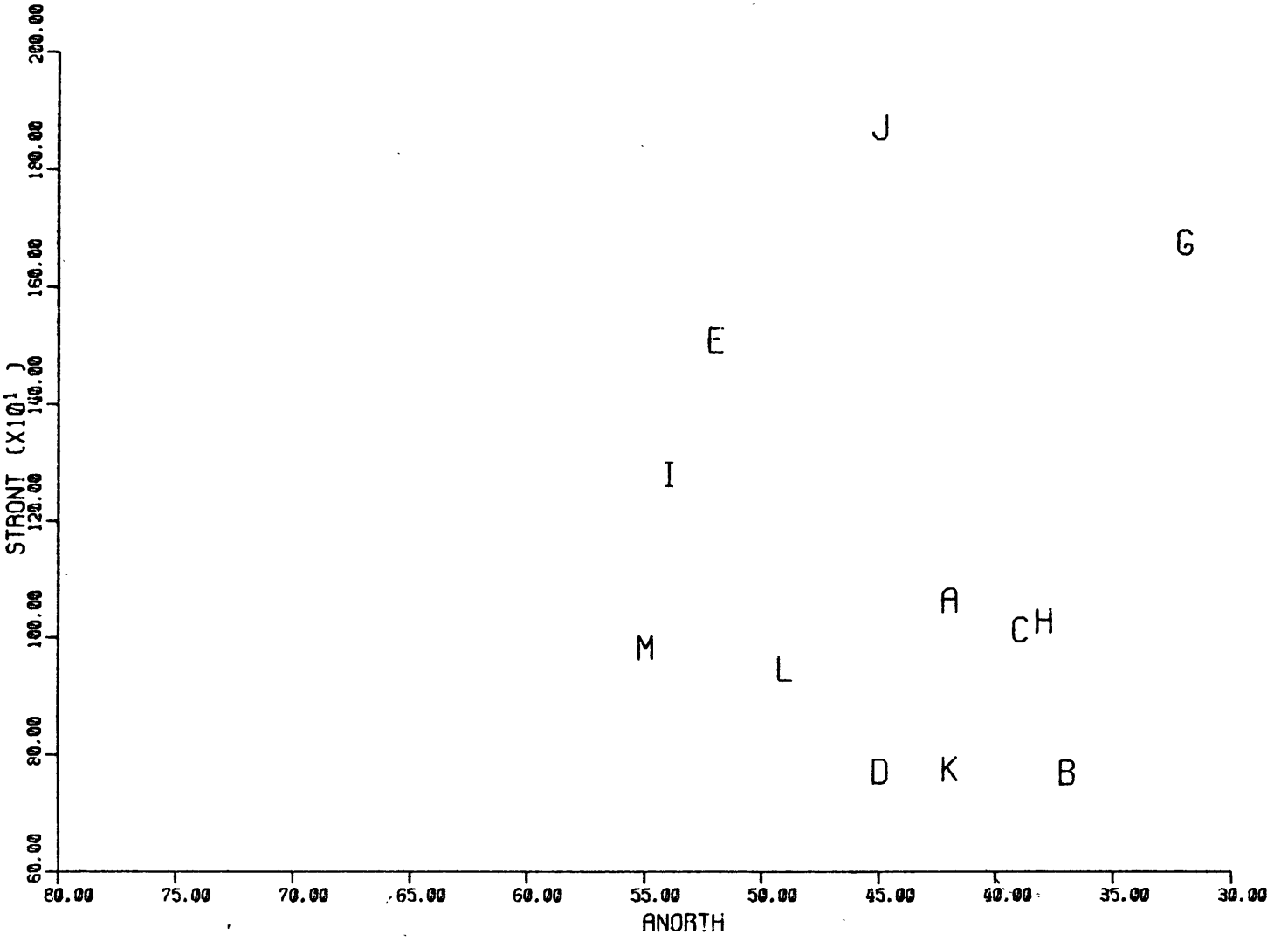


Figure 3. Cartesian graph plot of nonorm data.

Listing 2.-- *Batch mode input document*

```
TITLE GRANITIC ROCKS FROM THE AD DAWADIMI DISTRICT
NRMAISA39'6380160001840376028001790382025003500680140012
NRMCISA39+7169155000530103002700510416046301580090020002
OXIDES SIO2,TIO2,AL203,FE203,FEO,MNO,MGO,CAO,NA2O,K2O,H2O,P2O5;
MODIFY FORMAT (10X,12F4.2);
NONORMS;
NRM1AS-701695000381266024803460006012402750408040400750007
NRM2AS-704643200301382031904550008019903270453030100580009
NRM3AS-706638300671350034605300009016205150396026101080014
NRM4AS-206690000301306012201560004006602240687049000700004
NRM5AS-372731800261280006201950002002301260510041400200001
NRM6AS-395752000421110002801100009006101260512037500600007
NRM7AS726A7614001211450043007300030013008304410451003200002
NRM8AS-729754500671039011101640003003800950448046600670002
NRM9AS-822708300051340005600800001003301330450078000270001
NRMPAS-823692000171001029403920007021402700415038801010009
PRINT FEO+FE203/(MGO+FEO+FE203);
SUMMARY (OXIDES,MINERALS,BARTHS,RATIOS);
TERNARY Q,AB,OR;
STO DUMMY1850000101600040006000010025013000000130O
STO DUMMY2500000011000000100010001000100000000000
PLOT SIO2,CAO+NA2O;
GRAPHS=CALCOMP;
SCALE=1.0;
PLOT SIO2,K2O+NA2O;
H2O=0;FEO=FEO+0.9*FE203;FE203=0;FEMAG=FEO/(FEO+MGO);
RECALCULATE NORMS;
SUMMARY (ADJUSTED,MINERALS,PARTITIONS,USER);
TITLE PRINT OF SI BARTHS CATION:
PRINT #SI;
CONVERT VALUES;
TITLE THIS AFM PLOT IS IN MOL PERCENT.
TERNARY FEO+MNO,NA2O+K2O,MGO;
CLEAR STORAGE;
TITLE PLOTTING OF OTHER DATA - AL JI'LANI LAYERED INTRUSION.
OXIDES SIO2,AL203;
MODIFY FORMAT (10X,F2.0,2X,F4.0);
STOAG-114 42  1060
STOBG-116 37   765
STOCG-153 39  1010
STODG-163 45   766
STOEG-164A52  1505
STOFG-204 48   880
STOGG-207 32  1675
STOHG-212 38  1025
STOIG-214 54  1275
STOJG-217 45  1870
STOKG-263 42   770
STOLG-264 49   942
STOMG-269 55   978
STO DUMMY130  0600
STO DUMMY270  2000
ANORTH=SIO2;STRONT=AL203;
GRAPHS=CALCOMP;
PLOT(R) ANORTH,STRONT;
[*EOB*]
```

Listing 3.--*Output document*

GRANITIC ROCKS FROM THE AD DAWADIMI DISTRICT

ORIGINAL WT.PCT. OXIDES

SIO2	AL203	FE203	FEO	MGO	CAO	NA2O	K2O	H2O	TIO2	P2O5	MNO	ZRO2	CO2	SO3	CL	F	S	CR2O3	NIO	BAO
63.80	16.00	1.84	3.76	2.80	1.79	3.82	2.50	3.50	0.68	0.14	0.12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

SUM OF ORIGINAL OXIDES=100.75

CIPW NORM FOR SAMPLE NO. ISA39'

PLOTTING SYMBOL IS A

CONSTITUENTS	SIO2	AL203	FE203	FEO	MGO	CAO	NA2O	K2O	H2O	TIO2	P2O5	AL203/SIO2
PERCENTAGES	63.33	15.88	1.83	3.73	2.78	1.78	3.79	2.48	3.47	0.67	0.14	0.251
MOL. AMTS.	1.0539	0.1558	0.0114	0.0519	0.0689	0.0317	0.0612	0.0263	0.1928	0.0084	0.0010	

CONSTITUENTS	MNO	ZRO2	CO2	SO3	CL	F	S	CR2O3	NIO	BAO	FEO/FE2O3
PERCENTAGES	0.12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.043
MOL. AMTS.	0.0017	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	

MINERALS	Q	C	Z	OR	AB	AN	LC	NE	KP	HL	TH	NC
MOL. AMTS.	0.3693	0.0398	0.0000	0.0263	0.0612	0.0284	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
PERCENTAGES	22.190	4.060	0.000	14.663	32.083	7.906	0.000	0.000	0.000	0.000	0.000	0.000

MINERALS	AC	NS	KS	WO	EN	FS	FO	FA	CS	MT	CM	HM
MOL. AMTS.	0.0000	0.0000	0.0000	0.0000	0.0689	0.0337	0.0000	0.0000	0.0000	0.0114	0.0000	0.0000
PERCENTAGES	0.000	0.000	0.000	0.000	6.922	4.451	0.000	0.000	0.000	2.648	0.000	0.000

MINERALS	IL	TN	PF	RU	AP	FR	PR	CC	MG	TOTAL	SALIC	FEMIC
MOL. AMTS.	0.0084	0.0000	0.0000	0.0000	0.0010	0.0000	0.0000	0.0000	0.0000			
PERCENTAGES	1.282	0.000	0.000	0.000	0.329	0.000	0.000	0.000	0.000	96.535	80.903	15.632

MINERALS	DI	DI-WO	DI-EN	DI-FS	HY	HY-EN	HY-FS	OL	OL-FO	OL-FA	WOL
MOL. AMTS.	0.0000	0.0000	0.0000	0.0000	0.1027	0.0689	0.0337	0.0000	0.0000	0.0000	0.0000
PERCENTAGES	0.000	0.000	0.000	0.000	11.373	6.922	4.451	0.000	0.000	0.000	0.000

THORNTON + TUTTLE DIFFERENTIATION INDEX = 68.937

BARTHS CATIONS	SI	AL	FE+3	FE+2	MG	CA	NA	K	H	TI	P	MN
	60.99	18.03	1.32	3.01	3.99	1.83	7.08	3.05	22.32	0.49	0.11	0.10

	ZR	C	S1	CL	F	S2	CR	NI	BA
	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

NIGGLI VALUES	AL*	FM*	C*	ALK*	SI	RI	P	H	K	MG	SI'	QZ
	37.05	34.60	7.54	20.82	250.70	2.01	0.23	45.87	0.30	0.47	183.27	67.43

GRANITIC ROCKS FROM THE AD DAWADIMI DISTRICT

ORIGINAL WT.PCT. OXIDES

SIO2	AL203	FE203	FEO	MGO	CAO	NA20	K20	H20	TIO2	P205	MNO	ZRO2	CO2	SO3	CL	F	S	CR203	NIO	BAO
71.69	15.50	0.53	1.03	0.27	0.51	4.16	4.63	1.58	0.09	0.02	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

SUM OF ORIGINAL OXIDES=100.03

CIPW NORM FOR SAMPLE NO. ISA39+ PLOTTING SYMBOL IS C

CONSTITUENTS	SIO2	AL203	FE203	FEO	MGO	CAO	NA20	K20	H20	TIO2	P205	AL203/SIO2
PERCENTAGES	71.67	15.50	0.53	1.03	0.27	0.51	4.16	4.63	1.58	0.09	0.02	0.216
MOL. AMTS.	1.1928	0.1520	0.0033	0.0143	0.0067	0.0091	0.0671	0.0491	0.0877	0.0011	0.0001	

CONSTITUENTS	MNO	ZRO2	CO2	SO3	CL	F	S	CR203	NIO	BAO	FEO/FE203
PERCENTAGES	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.943
MOL. AMTS.	0.0003	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	

MINERALS	Q	C	Z	OR	AB	AN	LC	NE	KP	HL	TH	NC
MOL. AMTS.	0.4613	0.0271	0.0000	0.0491	0.0671	0.0086	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
PERCENTAGES	27.716	2.765	0.000	27.352	35.190	2.399	0.000	0.000	0.000	0.000	0.000	0.000

MINERALS	AC	NS	KS	WO	EN	FS	FO	FA	CS	MT	CM	HM
MOL. AMTS.	0.0000	0.0000	0.0000	0.0000	0.0067	0.0102	0.0000	0.0000	0.0000	0.0033	0.0000	0.0000
PERCENTAGES	0.000	0.000	0.000	0.000	0.672	1.342	0.000	0.000	0.000	0.768	0.000	0.000

MINERALS	IL	TN	PF	RU	AP	FR	PR	CC	MG	TOTAL	SALIC	FEMIC
MOL. AMTS.	0.0011	0.0000	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000	0.0000			
PERCENTAGES	0.171	0.000	0.000	0.000	0.047	0.000	0.000	0.000	0.000	98.422	95.421	3.000

MINERALS	DI	DI-WO	DI-EN	DI-FS	HY	HY-EN	HY-FS	OL	OL-FO	OL-FA	WOL
MOL. AMTS.	0.0000	0.0000	0.0000	0.0000	0.0169	0.0067	0.0102	0.0000	0.0000	0.0000	0.0000
PERCENTAGES	0.000	0.000	0.000	0.000	2.014	0.672	1.342	0.000	0.000	0.000	0.000

THORNTON + TUTTLE DIFFERENTIATION INDEX = 90.258

BARTHS CATIONS	SI	AL	FE+3	FE+2	MG	CA	NA	K	H	TI	P	MN
	67.48	17.19	0.38	0.81	0.38	0.51	7.59	5.56	9.92	0.06	0.02	0.02

	ZR	C	S1	CL	F	S2	CR	NI	BA
	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

NIGGLI VALUES	AL*	FM*	C*	ALK*	SI	RI	P	H	K	MG	SI'	QZ
	49.79	9.16	2.98	38.08	390.77	0.37	0.05	28.72	0.42	0.24	252.32	138.45

GRANITIC ROCKS FROM THE AD DAWADIMI DISTRICT

EVALUATION OF FEO+FE2O3/(MGO+FEO+FE2O3)

ID	VALUE	ID	VALUE	ID	VALUE	ID	VALUE	ID	VALUE	ID	VALUE
ISA39'	3.979	ISA39+	1.320	AS-701	3.805	AS-704	4.878	AS-706	5.633	AS-2?6	1.915
AS-372	2.171	AS-395	1.241	AS726A	1.063	AS-729	1.995	AS-822	1.131	AS-823	4.247

GRANITIC ROCKS FROM THE AD DAWADIMI DISTRICT

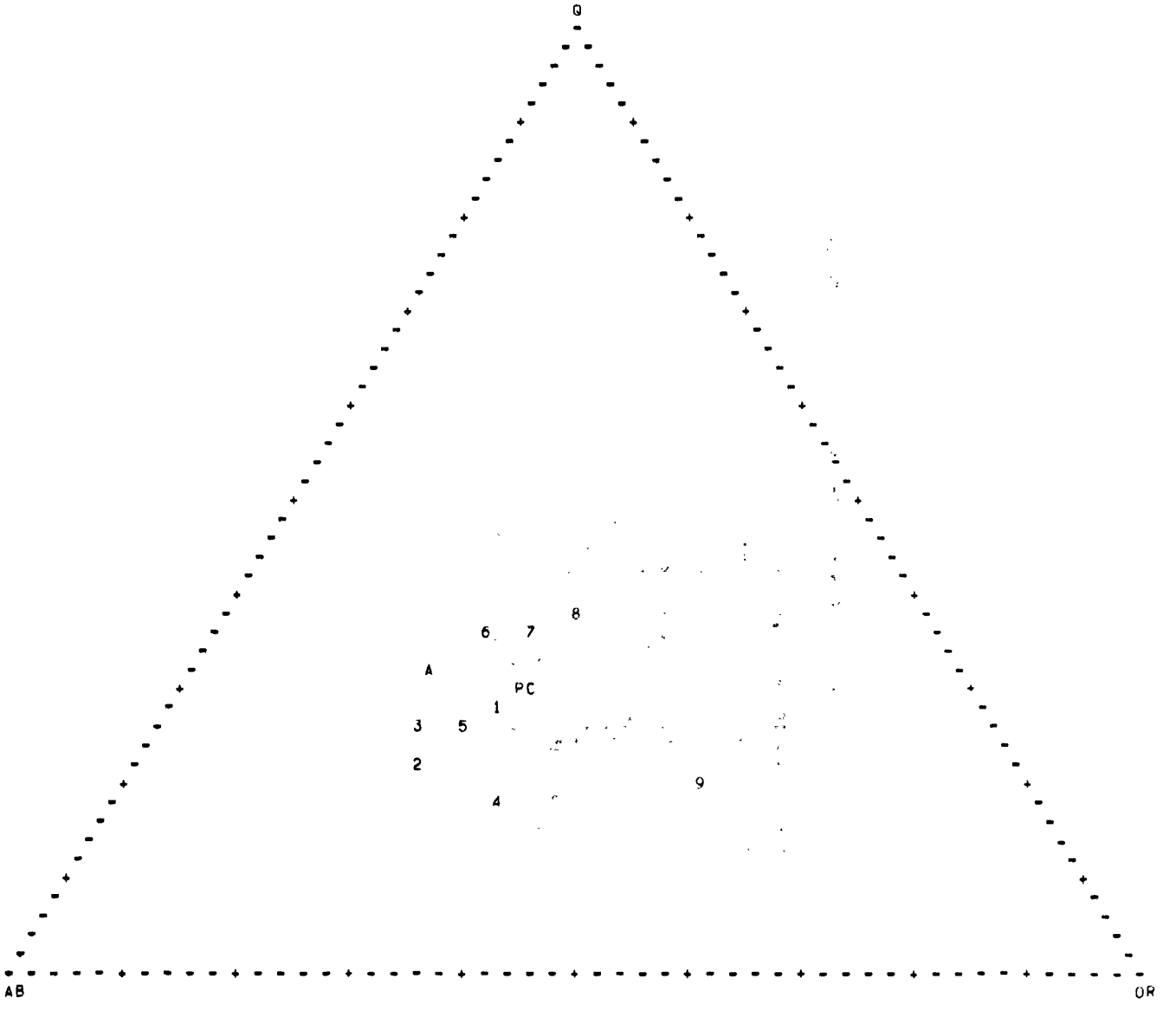
SYMBOL	ISA39' A	ISA39+ C	AS-701 1	AS-704 2	AS-706 3	AS-206 4	AS-372 5	AS-395 6	AS726A 7	AS-729 8	AS-822 9	AS-823 P
SIO2	63,80	71,69	69,50	64,32	63,83	69,00	73,18	75,20	76,14	75,45	70,83	69,20
TIO2	0,68	0,09	0,38	0,30	0,67	0,30	0,26	0,42	0,12	0,67	0,05	0,17
AL203	16,00	15,50	12,66	13,82	13,50	13,06	12,80	11,10	11,45	10,39	13,40	10,01
FE203	1,84	0,53	2,48	3,19	3,46	1,22	0,62	0,28	0,43	1,11	0,56	2,94
FEO	3,76	1,03	3,46	4,55	5,30	1,56	1,95	1,10	0,73	1,64	0,80	3,92
MNO	0,12	0,02	0,06	0,08	0,09	0,04	0,02	0,09	0,03	0,03	0,01	0,07
MGO	2,80	0,27	1,24	1,99	1,62	0,66	0,23	0,61	0,13	0,38	0,33	2,14
CAO	1,79	0,51	2,75	3,27	5,15	2,24	1,26	1,26	0,83	0,95	1,33	2,70
NA20	3,82	4,16	4,08	4,53	3,96	6,87	5,10	5,12	4,41	4,48	4,50	4,15
K20	2,50	4,63	4,04	3,01	2,61	4,90	4,14	3,75	4,51	4,66	7,80	3,88
H20	3,50	1,58	0,75	0,58	1,08	0,70	0,20	0,60	0,32	0,67	0,27	1,01
P205	0,14	0,02	0,07	0,09	0,14	0,04	0,01	0,07	0,02	0,02	0,01	0,09
TOTAL	100,75	100,03	101,47	99,73	101,41	100,59	99,77	99,60	99,12	100,45	99,89	100,28
Q	22,190	27,716	22,625	15,872	17,669	14,829	24,873	32,111	33,460	33,667	18,856	22,689
C	4,060	2,765										
OR	14,663	27,352	23,528	17,835	15,209	28,786	24,521	22,249	26,887	27,414	46,143	22,864
AB	32,083	35,190	34,024	38,435	33,043	39,662	42,889	36,363	34,086	27,376	25,528	29,803
AN	7,906	2,399	4,236	8,508	11,194							
AC						3,509	0,322	0,813	1,255	3,197	1,622	4,594
NS						3,292		1,446	0,497	1,567	2,502	
WO			3,657	2,993	5,469	4,504	2,589	2,429	1,680	1,905	2,731	5,332
EN	6,922	0,672	3,044	4,970	3,979	1,634	0,574	1,525	0,327	0,942	0,823	5,315
FS	4,451	1,342	3,734	5,388	5,852	2,429	2,775	1,500	1,209	1,952	1,407	5,918
MT	2,648	0,768	3,544	4,638	4,947		0,740					1,949
IL	1,282	0,171	0,711	0,571	1,255	0,566	0,495	0,801	0,230	1,267	0,095	0,322
AP	0,329	0,047	0,163	0,214	0,327	0,094	0,024	0,166	0,048	0,047	0,024	0,213
TOTAL	96,535	98,422	99,265	99,424	98,943	99,307	99,800	99,402	99,679	99,334	99,730	98,998
SALIC	80,903	95,421	84,412	80,651	77,115	83,277	92,283	90,722	94,434	88,457	90,527	75,356
FEMIC	15,632	3,000	14,853	18,773	21,828	16,030	7,518	8,680	5,245	10,877	9,203	23,642
BARTH'S CATIONS												
SI	60,99	67,48	64,69	60,52	59,98	62,81	68,11	70,39	71,77	70,74	65,37	65,39
AL	18,03	17,19	13,89	15,33	14,95	14,01	14,04	12,25	12,72	11,48	14,57	11,15
FE+3	1,32	0,38	1,74	2,26	2,45	0,84	0,43	0,20	0,31	0,78	0,39	2,09
FE+2	3,01	0,81	2,69	3,58	4,17	1,19	1,52	0,86	0,58	1,29	0,62	3,10
MG	3,99	0,38	1,72	2,79	2,27	0,90	0,32	0,85	0,18	0,53	0,45	3,01
CA	1,83	0,51	2,74	3,30	5,19	2,18	1,26	1,26	0,84	0,95	1,32	2,73
NA	7,08	7,59	7,36	8,26	7,22	12,12	9,20	9,29	8,06	8,14	8,05	7,60
K	3,05	5,56	4,80	3,61	3,13	5,69	4,92	4,48	5,42	5,57	9,18	4,68
H	22,32	9,92	4,66	3,64	6,77	4,25	1,24	3,75	2,01	4,19	1,66	6,37
TI	0,49	0,06	0,27	0,21	0,47	0,21	0,18	0,30	0,09	0,47	0,03	0,12
P	0,11	0,02	0,06	0,07	0,11	0,03	0,01	0,06	0,02	0,02	0,01	0,07
MN	0,10	0,02	0,05	0,06	0,07	0,03	0,02	0,07	0,02	0,02	0,01	0,06
AL203/SIO2	0,251	0,216	0,182	0,215	0,211	0,189	0,175	0,148	0,150	0,138	0,189	0,145
FEO/FE203	2,043	1,943	1,395	1,426	1,532	1,279	3,145	3,929	1,698	1,477	1,429	1,333

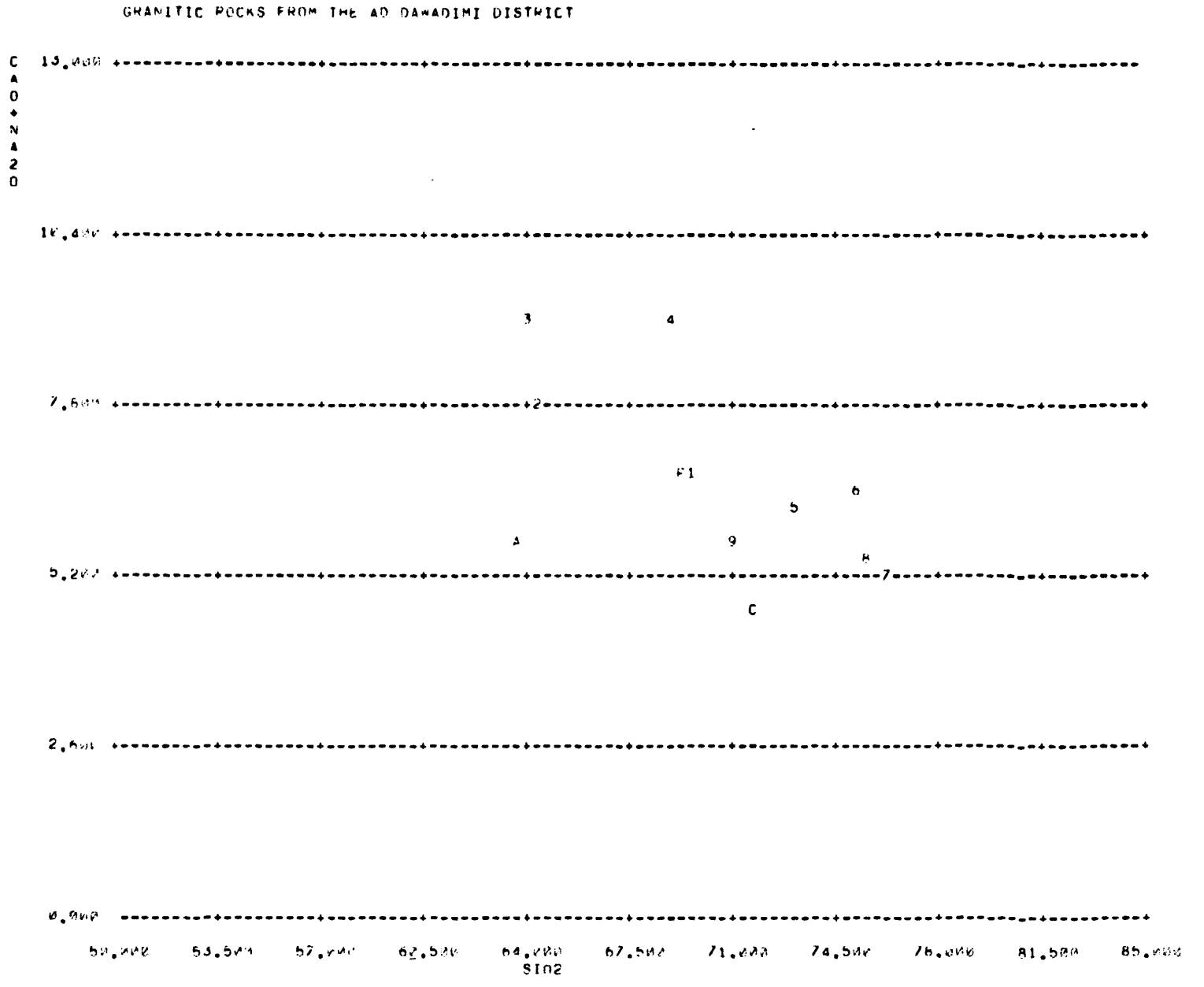
GRANITIC ROCKS FROM THE AD DAWADIMI DISTRICT

TERNARY RATIOS FOR Q,AB,OR

SAMPLE	R1	R2	R3	SYMBOL
ISA39'	32.19	46.54	21.27	A
ISA39+	30.71	38.99	30.30	C
AS-701	28.22	42.44	29.34	1
AS-704	22.00	53.28	24.72	2
AS-706	26.80	50.12	23.07	3
AS-206	17.81	47.63	34.57	4
AS-372	26.95	46.48	26.57	5
AS-395	35.39	40.08	24.52	6
AS726A	35.43	36.10	28.47	7
AS-729	38.06	30.95	30.99	8
AS-822	20.83	28.20	50.97	9
AS-823	30.11	39.55	30.34	P

GRANITIC ROCKS FROM THE AD DAWADIMI DISTRICT




GRANITIC ROCKS FROM THE AD DAWADIMI DISTRICT
CaO+Na2O
13.000
10.400
7.800
5.200
2.600
0.000
50.000
53.500
57.000
62.500
64.000
67.500
71.000
74.500
78.000
81.500
85.000
SiO2
3
4
2
F1
6
5
A
9
8
7
C

GRANITIC ROCKS FROM THE AD'DAWADIMI DISTRICT

H2O	DEFINED AS 0
FEO	DEFINED AS FEO+0.9*FE2O3
FE2O3	DEFINED AS 0
FEMAG	DEFINED AS FEO/(FEO+MGO)

GRANITIC ROCKS FROM THE AD DAWADIMI DISTRICT

SYMBOL	ISA39' A	ISA39+ C	AS-701 1	AS-704 2	AS-706 3	AS-206 4	AS-372 5	AS-395 6	AS726A 7	AS-729 8	AS-822 9	AS-823 P
ADJUSTED OXIDES												
SIO2	65.73	72.86	69.17	65.08	63.84	69.16	73.54	75.98	77.10	75.70	71.14	69.92
TIO2	0.70	0.09	0.38	0.30	0.67	0.30	0.26	0.42	0.12	0.67	0.05	0.17
AL2O3	16.48	15.75	12.60	13.98	13.50	13.09	12.86	11.22	11.59	10.42	13.46	10.11
FEO	5.58	1.53	5.67	7.51	8.42	2.66	2.52	1.37	1.13	2.65	1.31	6.63
MNO	0.12	0.02	0.06	0.08	0.09	0.04	0.02	0.09	0.03	0.03	0.01	0.07
MGO	2.88	0.27	1.23	2.01	1.62	0.66	0.23	0.62	0.13	0.38	0.33	2.16
CAO	1.84	0.52	2.74	3.31	5.15	2.25	1.27	1.27	0.84	0.95	1.34	2.73
NA2O	3.94	4.23	4.06	4.58	3.96	6.89	5.13	5.17	4.47	4.49	4.52	4.19
K2O	2.58	4.71	4.02	3.05	2.61	4.91	4.16	3.79	4.57	4.68	7.83	3.92
P2O5	0.14	0.02	0.07	0.09	0.14	0.04	0.01	0.07	0.02	0.02	0.01	0.09
Q	20.892	27.568	20.063	12.373	14.014	15.411	24.403	32.421	33.747	34.350	19.129	22.056
C	4.214	2.811										
OR	15.220	27.806	23.761	17.997	15.426	29.023	24.585	22.390	26.986	27.629	46.294	23.165
AB	33.301	35.774	34.362	38.785	33.514	39.989	43.002	36.593	34.212	27.590	25.612	30.196
AN	8.206	2.439	4.278	8.586	11.354							
NS						4.254	0.085	1.671	0.832	2.431	2.940	1.230
WO			3.693	3.020	5.547	4.541	2.595	2.444	1.686	1.920	2.740	5.403
EN	7.184	0.683	3.074	5.015	4.035	1.648	0.576	1.535	0.328	0.950	0.825	5.385
FS	9.319	2.699	9.890	13.438	14.514	4.470	4.234	1.977	1.933	3.808	2.341	12.030
IL	1.331	0.174	0.718	0.577	1.273	0.571	0.496	0.806	0.231	1.277	0.095	0.326
AP	0.342	0.048	0.165	0.216	0.332	0.095	0.024	0.168	0.048	0.048	0.024	0.215
TOTAL	100.009	100.001	100.004	100.006	100.008	100.003	100.001	100.005	100.001	100.001	100.001	100.006
SALIC	81.834	96.397	82.464	77.741	74.308	84.423	91.990	91.404	94.944	89.569	91.035	75.417
FEMIC	18.176	3.604	17.540	22.265	25.700	15.579	8.011	8.601	5.057	10.432	8.965	24.589
DI			7.597	6.181	11.443	9.297	5.436	4.885	3.517	3.971	5.616	10.995
DIWO			3.693	3.020	5.547	4.541	2.595	2.444	1.686	1.920	2.740	5.403
DIEN			0.926	0.859	1.283	1.281	0.340	1.067	0.266	0.409	0.750	1.729
DIFS			2.978	2.302	4.614	3.475	2.501	1.374	1.566	1.642	2.126	3.863
HY	16.503	3.383	9.060	15.292	12.653	1.362	1.969	1.071	0.430	2.706	0.290	11.822
HYEN	7.184	0.683	2.148	4.156	2.752	0.367	0.236	0.468	0.062	0.540	0.076	3.655
HYFS	9.319	2.699	6.911	11.136	9.900	0.995	1.733	0.603	0.367	2.166	0.215	8.166
WOL												
FEMAG	0.659	0.848	0.821	0.789	0.839	0.841	0.916	0.689	0.806	0.874	0.798	0.754

PRINT OF SI BARTHS CATION:

EVALUATION OF #SI

ID	VALUE	ID	VALUE	ID	VALUE	ID	VALUE	ID	VALUE	ID	VALUE
ISA39'	64.991	ISA39+	67.479	AS-701	64.690	AS-704	60.522	AS-706	59.982	AS-206	62.806
AS-372	68.138	AS-395	70.389	AS726A	71.771	AS-729	70.736	AS-822	65.366	AS-823	65.388
DUMMY1	56.106	DUMMY2	80.863								

THIS AFM PLOT IS IN MOL PERCENT.

TERNARY RATIOS FOR FEO+MNO,NA2O+K2O,MGO

SAMPLE	R1	R2	R3	SYMBOL
ISA39'	32.84	37.57	29.59	A
ISA39+	14.74	80.62	4.64	C
AS-701	36.47	49.52	14.01	1
AS-704	40.34	40.58	19.07	2
AS-706	47.32	36.62	16.06	3
AS-206	17.33	75.12	7.55	4
AS-372	21.06	75.53	3.41	5
AS-395	12.74	77.66	9.60	6
AS726A	11.55	86.11	2.33	7
AS-729	22.07	72.33	5.60	8
AS-822	10.06	85.44	4.50	9
AS-823	36.42	42.64	20.93	P
DUMMY1	40.30	41.19	18.51	
DUMMY2	62.04	0.00	37.96	

APPENDIX C

GNAP source code

Listing 4.--*GNAP source code*

```
GNAP OVERLAY DESCRIPTION.


  .ROOT GNAP-[1,1]XYLIBF4P/LB-R2-*(Z,A,B,C,D,E)
R2:   .FCTR UTILITY
 Z:   .FCTR PREPRO
 A:   .FCTR NORM
 B:   .FCTR FUNC-CONV
 C:   .FCTR SUMPNT-CLEAR
 D:   .FCTR PRNT-PRPLOT-[1,1]XYLIBF4P/LB
 E:   .FCTR TRIANG-SIDE-[1,1]XYLIBF4P/LB
  .END
```

```
C
C  A USER ORIENTED COMMAND LANGUAGE IS USED TO PROVIDE COMPLETE CONTROL
C  OVER THE COMPUTATION AND GRAPHIC OUTPUT OF THE STANDARD CIPW NORM
C  COMPUTATION. THE GRAPHIC OUTPUT FALLS INTO TWO CATEGORIES. THESE
C  CATEGORIES ARE X-Y PLOTS AND TERNARY DIAGRAMS. PLOTTING MAY BE DONE
C  ON THE LINEPRINTER OR ON AN OFFLINE PLOTTER (CALCOMP). PROVISION IS
C  ALSO MADE TO ALLOW THE USER TO EVALUATE AND PRINT ARITHMETIC
C  EXPRESSIONS GIVEN IN TERMS OF THE OXIDES AND NORMATIVE MINERALS
C  DETERMINED BY THE CIPW NORM CALCULATION.
C
C  THE ORIGINAL WEIGHT PERCENT OXIDES MAY BE PROVIDED IN A STANDARD
C  FORMAT OR THE USER MAY SPECIFY HIS OWN FORMAT FOR THESE VALUES. THESE
C  OXIDE VALUES MAY BE STORED AND MANIPULATED PRIOR TO THE NORM
C  DETERMINATION. PROVISION IS ALSO MADE FOR A COLUMNWISE SUMMARY
C  PRINTOUT OF THE STORED VALUES (BOTH ORIGINAL AND CALCULATED).
C
C  THE PROGRAM IS WRITTEN IN FORTRAN (IV) FOR THE IBM S360/65. REFERENCE
C  IS MADE TO A SYSTEM RESIDENT ROUTINE (CORE) WHICH PROVIDES IN-CORE
C  READ/WRITE CAPABILITIES. THE STANDARD CALCOMP PLOTTER SUBROUTINE
C  PACKAGE IS ALSO REFERENCED.
C                  (DATE OF LAST REVISION 10/26/72)
C                                          PROGRAMMED BY ROGER W. BOWEN
C
C  DEC 1977 - GNAP ADAPTED TO RUN ON THE D.G.M.R. PDP-11/45 COMPUTER BY
C  LESTER NORTH, U.S.G.S.
C
C  APR 1978 - MAJOR MODIFICATIONS MADE TO GNAP ON THE PDP-11/45 COMPUTER
C  FOR D.G.M.R. AND U.S.G.S. USE IN SAUDI ARABIA. LINES MODIFIED OR
C  ADDED HAVE, IN GENERAL, HAD THEIR SEQUENCE NUMBERS IN COLUMNS 73-80
C  REMOVED.  THIS VERSION RUNS UNDER RSX11-D USING THE F4P COMPILER,
C  AND OCCUPIES A FULL PAGE OF CORE.
C                                       JOHN ODELL, D.G.M.R.
C
C*****************************************************************************
C
      PARAMETER IV=20
C
      BYTE FIL(32),BATCH,BAT
C
      REAL NIGGLI(12,IV )
      DIMENSION X(IV+2),Y(IV+2),ADJUST(21,IV),AREA(1),VALUES(100,IV),
     1 TTDI( IV),RATIOS(2, IV),TOTALS(2, IV),TYPES(2, IV)
      LOGICAL*1 CARD(80),STMT(160),FORMAT(96),NEXTC(4),BLNK,NAME(8),
     1          HEADG(75),IFORM(16),SYM(IV),SPFMT(17),RASSYM,FLAG
     2 ,PLUS, ERR,STORE,NOTEQ,STAND,OFFINE,PRNTER,NONORM,SWCC,NORMAL
C
      INTEGER CNT,PT,BUFFER( 80 ),ORDER(21),UN,U
      DOUBLE PRECISION IDENT(IV),RNAME,NAMES(100),DUMMY(12),BLANK,
     1 FORMLD(2),DEFILE(10),DPFMT(3)
C
      COMMON HEADG,IFORM,NORMAL,RASSYM
C
      COMMON /COMM/ U,UN,FIL,BAT
```

```
C
C
C       THIS COMMON BLOCK ALIGNS VECTORS FOR ZEROING OUT
C
      COMMON /DSPACE/ TTDI,ADJUST,VALUES,NIGGLI,RATIOS,TOTALS,TYPES
C
      EQUIVALENCE (NAME(1),RNAME),(TTDI(1),AREA(1)),(DPFMT(1),SPFMT(1))
C
      DATA ISM/';'/,IBK/' '/
      DATA PT,BLNK/80,' '/,DEFTLE/'PROGRAM','GNAP, NO',' TITLE S',
     1 'UPPLIED.',6*'      '/,FORMLD/'(1H1,10X',',75A1//)'/
      DATA NAMES/
     2 'SIO2  ','AL203 ','FE203 ','FEO   ','MGO   ','CAO   ','NA20  ',
     3 'K20   ','H20   ','TIO2  ','P205  ','MNO   ','ZRO2  ','CO2   ',
     4 'SO3   ','CL    ','F     ','S     ','CR203 ','NIO   ','BAO   ',
     5 'Q     ','C     ','Z     ','OR    ','AB    ','AN    ','LC    ',
     6 'NE    ','KP    ','HL    ','TH    ','NC    ','AC    ','NS    ',
     7 'KS    ','WO    ','EN    ','FS    ','FO    ','FA    ','CS    ',
     8 'MT    ','CM    ','HM    ','IL    ','TN    ','PF    ','RU    ',
     9 'AP    ','FR    ','PR    ','CC    ','MG    ','DI    ','DIWO  ',
     A 'DIEN  ','DIFS  ','HY    ','HYEN  ','HYFS  ','OL    ','OLFO  ',
     B 'OLFA  ','WOL   ',' SI   ',' AL   ',' FE3  ',' FE2  ',' MG   ',
     C ' CA   ',' NA   ',' K    ',' H    ',' TI   ',' P    ',' MN   ',
     D ' ZR   ',' C    ',' S1   ',' CL   ',' F    ',' S2   ',' CR   ',
     E ' NI   ',' BA   ',14*'      '/
C
      DATA BLANK/'       '/,
     C DPFMT/'(10(1X,G','6,0),10X',')'/,NEXTC,DUMMY/4*"000,'(10X,9F4',
     D ',2,2F3.2',',F4.2,F3',',.2,F4.2,',5F3.2,F2',',2)     ',
     E 6*'      '/,PLUS/'X'/
C
C     INITIALIZE CONSTANTS TO DEFAULT VALUES
C
      BATCH=.FALSE.
      SWCC=.FALSE.
      FLAG=.FALSE.
  110 CNT=0
      NCS=1
      NUM=86
      NOX=20
      CALL MOVE (DUMMY,FORMAT,96)
      CALL MOVE (FORMLD,TFORM,16)
      CALL MOVE (DEFTLE,HEADG,75)
      PRNTER=.TRUE.
      NORMAL=.TRUE.
      STORE=.FALSE.
      NONORM=.FALSE.
      SCALE=1.0
      IF (FLAG) SCALE=2.54
      DO 120 I=1,21
  120 ORDER(I)=I
C     ZERO OUT ALL VALUES CALCULATED BY SUBROUTINE NORM
      KKK=IV*(1+21+100+12+2+2+2)
```

```
      DO 130 I=1,KKK
  130 AREA(I)=0.0
C
C
      BAT=.FALSE.
C
      IF (BATCH.OR.FLAG) GO TO 910
C
      CALL PREPRO(BATCH)
C
  910 DEFINE = .FALSE.
  140 LSTMT=0
  150 PT=PT+1
      IF (PT.LE.80) GO TO 210
C
      IF (.NOT.BATCH) WRITE(U,123)
123   FORMAT(' **GNAP>'$)
      PT=1
      READ (UN,740,END=690) CARD
      IF(NORMAL) GO TO 159
C
C     LAST CARD IN STATPAC FORMAT CHANGE INPUT PHASE * * *  * * * * * *
      IF(NOTEQ('LAST',CARD,4)) GO TO 151
      NORMAL=.TRUE.
      CALL MOVE(DUMMY,FORMAT,96)
      NCS=1
      PT=80
      GO TO 910
C
C     RASS SYMBOL PROCESSOR * * * * * * * * * * * * * * * * * * * * * *
C
  151 IF(NOTEQ('SYMBOL ',CARD,7)) GO TO 152
      RASSYM=CARD(8)
      PT=80
      GO TO 910
  152 CNT=CNT+1
      IF(CNT.LE.IV) GO TO 153
      CNT=IV
      WRITE(U,750) IV
  153 SYM(CNT)=RASSYM
      GO TO 201
C
C     TITLE PROCESSOR
C
  159 IF (NOTEQ('TITLE',CARD,5)) GO TO 170
      DO 160 J=1,75
  160 HEADG(J)=CARD(J+5)
      PT=80
      GO TO 910
  170 IF (NOTEQ('STO',CARD,3)) GO TO 180
      STORE=.TRUE.
      GO TO 190
  180 IF (NOTEQ('NRM',CARD,3)) GO TO 210
```

```
C
C       NORM AND STORE PROCESSOR * * * * * * * * * * * * * * * * * * * * * *
  190 CNT=CNT+1
      IF (CNT.LE.IV) GO TO 200
      CNT=IV
      WRITE(U,750) IV
200   CONTINUE
  201 CALL NORM (CARD,VALUES(1,CNT),STORE,FORMAT,NCS,TTDI(CNT),
     1 RATIOS(1,CNT),TOTALS(1,CNT),TYPES(1,CNT),
     2 NIGGLI(1,CNT),IDENT(CNT),NOX,ORDER,NONORM,ADJUST(1,CNT),
     3 SYM(CNT),NAMES,CNT)
C
 IF (BAT) GO TO 190
C
      STORE=.FALSE.
      PT=80
      GO TO 910
  210 NEXTC(4)=CARD(PT)
      NEXT=NEXTC(4)
C     TEST FOR SEMICOLON
      IF(NEXT.EQ.  59) GO TO 230
C     TEST FOR BLANK
      IF (NEXT.EQ. 32) GO TO 150
      LSTMT=LSTMT+1
      IF (LSTMT.GT.160) GO TO 220
      STMT(LSTMT)=NEXTC(4)
      GO TO 150
  220 WRITE(U,760) STMT
      GO TO 910
C
C       STATEMENT BUILT, DETERMINE PRINT AND PROCESS * * * * * * * * * * *
C
  230 IF (NOTEQ(STMT,'CLEAR',5)) GO TO 240
C
C       CLEAR STORAGE PROCESSOR * * * * * * * * * * * * * * * * * * * * * *
C       BY BRANCHING TO START, THEN OVERWRITE
      IF (LSTMT.GT.12) WRITE(U,900) (STMT(I),I=1,LSTMT)
      FLAG=.TRUE.
      GO TO 110
C
  240 IF (NOTEQ(STMT,'SCALE=',6)) GO TO 270
C
C       SET THE SCALE FOR THE CALCOMP PLOTTER * * * * * * * * * * * * * * *
      IF (LSTMT.GT.14) WRITE(U,900) (STMT(I),I=1,LSTMT)
      IF (PRNTER) GO TO 260
      SCALE=CONV(STMT,7,LSTMT,0,ERR)
      IF (ERR.OR.SCALE.GT.3.0) GO TO 250
      SCALE=SCALE*2.54
      CALL FACTOR (SCALE)
      GO TO 910
  250 WRITE(U,700)
      GO TO 910
  260 WRITE(U,710)
```

```
      GO TO 910
C
  270 IF (NOTEQ(STMT,'OXIDES',6)) GO TO 320
C
C     OXIDES AND ORDER STATEMENT PROCESSOR * * * * * * * * * * * * * * * *
      IF (LSTMT.GT.95) WRITE(U,900) (STMT(I),I=1,LSTMT)
      J=7
      LSTMT=LSTMT-6
      NOX=0
C     DETERMINE LENGTH (I) OF NEXT OXIDE NAME.
  280 I=INDEX(STMT(J),',',LSTMT)-1
      IF (I.LT.1) I=LSTMT
      RNAME=BLANK
      DO 290 L=1,I
  290 NAME(L)=STMT(J+L-1)
C     FIND NAME IN LIST OF OXIDES.
      DO 300 L=1,21
  300 IF (RNAME.EQ.NAMES(L)) GO TO 310
C     NAME NOT FOUND, PRINT ERROR MESSAGE AND TERMINATE RUN.
      WRITE(U,890) (NAMES(I),I=1,21),RNAME
      GO TO 690
C     INCREASE NUMBER OF OXIDES AND STORE ORDER.
  310 NOX=NOX+1
      ORDER(NOX)=L
      J=J+I+1
      LSTMT=LSTMT-I-1
      IF (LSTMT.GT.0) GO TO 280
      GO TO 910
C
  320 IF (NOTEQ(STMT,'NONORM',6)) GO TO 330
C
C     NONORM STATEMENT PROCESSOR * * * * * * * * * * * * * * * * * * * *
      IF (LSTMT.GT.7) WRITE(U,900) (STMT(I),I=1,LSTMT)
      NONORM=.TRUE.
      GO TO 910
C
  330 IF (NOTEQ(STMT,'NORM',4)) GO TO 340
C     NORM STATEMENT PROCESSOR * * * * * * * * * * * * * * * * * * * * *
      IF (LSTMT.GT.5) WRITE(U,900) (STMT(I),I=1,LSTMT)
      NONORM=.FALSE.
      GO TO 910
C
  340 IF (NOTEQ(STMT,'GRAPHS',6)) GO TO 370
C
C     PLOTTING DEVICE SELECTION COMMAND PROCESSOR * * * * * * * * * * *
      IF (LSTMT.GT.14) WRITE(U,900) (STMT(I),I=1,LSTMT)
      IF (NOTEQ(STMT(7),'CALCOMP',7).AND.NOTEQ(STMT(8),'CALCOMP',7)) GO
     1TO 360
      PRNTER=.FALSE.
      IF (SWCC) GO TO 350
      CALL NEWDEV(3,'SY:GNAP.PLT',11)
      CALL PLOTST(0.01,'CM')
      CALL FACTOR(2.54)
```

```
      SWCC=.TRUE.
  350 CALL PLOT (2.0,1.12,-3)
      GO TO 910
  360 PRNTER=.TRUE.
      GO TO 910
C
  370 IF (NOTEQ(STMT,'SUMMARY(',8)) GO TO 375
C
C     SUMMARY PRINT PROCESSOR * * * * * * * * * * * * * * * * * * * * *
      IF (LSTMT.GT.74) WRITE(U,900) (STMT(I),I=1,LSTMT)
      LSTMT=LSTMT-8
      CALL SUMPNT (STMT(9),LSTMT,VALUES,CNT,IDENT,NAMES,
     1 TOTALS,TYPES,NIGGLI,RATIOS,TTDI,NOX,ORDER,SYM,NUM,ADJUST)
      GO TO 910
C
  375 CONTINUE
      IF (NOTEQ(STMT,'MODIFY',6)) GO TO 410
C
C     MODIFY COMMAND PROCESSOR * * * * * * * * * * * * * * * * * * * *
      IF (NOTEQ(STMT(7),'FORMAT',6)) GO TO 390
      DO 376 J=1,96
  376 FORMAT(J)=BLNK
      IF(NOTEQ(STMT(14),'STATPAC',7)) GO TO 378
      DO 377 I=1,17
  377 FORMAT(I)=SPFMT(I)
      NORMAL=.FALSE.
      RASSYM=PLUS
      NCS=NOX/10
      IF(10*NCS.NE.NOX) NCS=NCS+1
      IF(LSTMT.EQ.20) GO TO 910
      I=14
      J=30
      CALL MOVE(STMT(22),STMT(7),LSTMT-21)
      LSTMT=LSTMT-15
      GO TO 379
  378 IF (INDEX(STMT,')',LSTMT).EQ.0) GO TO 380
      I=INDEX(STMT,'(',LSTMT)
      IF (I.EQ.0) GO TO 380
      J=INDEX(STMT,')',LSTMT)
      CALL MOVE (STMT(I),FORMAT,J-I+1)
      IF (J.EQ.LSTMT) GO TO 910
      CALL MOVE (STMT(J+2  ),STMT(7),LSTMT-J-1)
      LSTMT=LSTMT-J+5
  379 CALL MOVE (FORMAT,BUFFER,J-I+1)
      CALL MOVE (STMT(11),FORMAT(2),LSTMT-10)
      CALL MOVE (BUFFER,FORMAT(LSTMT-8),J-I+1)
      FORMAT(LSTMT-7+J-I)=FORMAT(LSTMT-8+J-I)
      IF(.NOT.NORMAL) GO TO 910
      GO TO 390
  380 WRITE(U,770) CARD
      GO TO 910
  390 IF (NOTEQ(STMT(7),'NCS=',4)) GO TO 400
      NCS=CONV(STMT,11,LSTMT,0,ERR)+0.1
```

```
      IF (ERR) GO TO 400
      GO TO 910
  400 WRITE(U,780) CARD
      NCS=1
      GO TO 910
C
  410 I=INDEX(STMT,'=',LSTMT)
      IF (I.EQ.0) GO TO 480
C
C     DEFINITION PROCESSOR * * * * * * * * * * * * * * * * * * * * * * *
      LSTM=LSTMT
      J=MIN0(8,I-1)
      RNAME=BLANK
      DO 420 L=1,J
  420 NAME(L)=STMT(L)
      LSTMT=LSTMT-I
      K=I+1
      L=NUM
      DO 430 J=1,NUM
  430 IF (RNAME.EQ.NAMES(J)) GO TO 450
      J=NUM+1
      NUM=NUM+1
      IF (J.LE.100) GO TO 440
      NUM=100
      WRITE(U,790) RNAME
      GO TO 910
  440 NAMES(J)=RNAME
  450 CALL PARSE (STMT(K),LSTMT,NAMES,L,ERR)
      IF (ERR) GO TO 910
      DO 460 I=1,CNT
  460 VALUES(J,I)=EVAL(VALUES(1,I))
      IF (DEFINE) GO TO 470
      WRITE(U,TFORM) HEADG
      DEFINE = .TRUE.
      WRITE(U,720) RNAME,(STMT(I),I=K,LSTM)
      GO TO 140
  470 WRITE(U,730) RNAME,(STMT(I),I=K,LSTM)
      GO TO 140
C
  480 IF (NOTEQ(STMT,'RECALC',6)) GO TO 500
C
C     NORM RECALCULATION PROCESSOR * * * * * * * * * * * * * * * * * * *
      IF (LSTMT.GT.16) WRITE(U,900) (STMT(I),I=1,LSTMT)
      DO 490 I=1,CNT
  490 CALL RECALC (VALUES(1,I),IDENT(I),SYM(I),TTDI(I),RATIOS(1,I),TOTAL
     1S(1,I),TYPES(1,I),NIGGLI(1,I),NONORM,ADJUST(1,I))
      GO TO 910
C
  500 IF (NOTEQ(STMT,'PRINT',5)) GO TO 550
C
C     PRINT PROCESSOR * * * * * * * * * * * * * * * * * * * * * * * * *
      J=1
      LSTMT=LSTMT-5
```

```
      K=6
  510 IF (J.EQ.0) GO TO 910
      J=INDEX(STMT(K),',',LSTMT)
      IF (J.NE.0) GO TO 520
      LE1=LSTMT
      GO TO 530
  520 LE1=J-1
      LSTMT=LSTMT-J
  530 CALL PARSE (STMT(K),LE1,NAMES,NUM,ERR)
      IF (ERR) GO TO 910
      DO 540 I=1,CNT
  540 X(I)=EVAL(VALUES(1,I))
      LE1=K+LE1-1
      WRITE(U,TFORM) HEADG
      WRITE(U,800) (STMT(I),I=K,LE1)
      WRITE(U,810)
      WRITE(U,820) (IDENT(I),X(I),I=1,CNT)
      K=K+J
      GO TO 510
C
  550 IF (NOTEQ(STMT,'CONVERT',7)) GO TO 560
C
C     VALUE CONVERT PROCESSOR * * * * * * * * * * * * * * * * * * * * *
      IF (LSTMT.GT.13) WRITE(U,900) (STMT(I),I=1,LSTMT)
      CALL CONVER (VALUES,CNT)
      GO TO 910
C
  560 IF (NOTEQ(STMT,'PLOT',4)) GO TO 610
C
C     X-Y PLOT PROCESSOR * * * * * * * * * * * * * * * * * * * * * * * *
      LSTMT=LSTMT-4
      STAND=NOTEQ(STMT(5),'(R)',3)
      K=5
      IF (STAND) GO TO 570
      LSTMT=LSTMT-3
      K=8
  570 I=INDEX(STMT(K),',',LSTMT)
      IF (I.NE.0) GO TO 580
      WRITE(U,830) CARD
      GO TO 910
  580 LE1=I-1
      LE2=LSTMT-I
      J=I+K
      CALL PARSE (STMT(K),LE1,NAMES,NUM,ERR)
      IF (ERR) GO TO 910
      DO 590 I=1,CNT
  590 X(I)=EVAL(VALUES(1,I))
      CALL PARSE (STMT(J),LE2,NAMES,NUM,ERR)
      IF (ERR) GO TO 910
      DO 600 I=1,CNT
  600 Y(I)=EVAL(VALUES(1,I))
      CALL PRNT (X,Y,SYM,CNT,STMT(K),STMT(J),LE1,LE2,STAND,     PRNTER,S
     1CALE)
```

```
      GO TO 910
C
  610 IF (NOTEQ(STMT,'TERNARY',7)) GO TO 680
C
C     TERNARY DIAGRAM PROCESSOR * * * * * * * * * * * * * * * * * * * * *
      LSTMT=LSTMT-7
      I=INDEX(STMT(8),',',LSTMT)
      IF (I.NE.0) GO TO 630
  620 WRITE(U,840) CARD
      GO TO 910
  630 LE1=I-1
      LSTMT=LSTMT-I
      I=INDEX(STMT(I+8),',',LSTMT)
      IF (I.EQ.0) GO TO 620
      LE2=I-1
      LE3=LSTMT-I
      J=LE1+LE2+10
      K=LE1+9
      CALL PARSE (STMT(8),LE1,NAMES,NUM,ERR)
      IF (ERR) GO TO 910
      DO 640 I=1,CNT
  640 Y(I)=AMAX1(0.,EVAL(VALUES(1,I)))
      CALL PARSE (STMT(J),LE3,NAMES,NUM,ERR)
      IF (ERR) GO TO 910
      DO 650 I=1,CNT
  650 X(I)=AMAX1(0.,EVAL(VALUES(1,I)))
      CALL PARSE (STMT(K),LE2,NAMES,NUM,ERR)
      IF (ERR) GO TO 910
      DO 660 I=1,CNT
      Y1=Y(I)
      X1=X(I)
      SUM=X1+Y1+AMAX1(0.,EVAL(VALUES(1,I)))
      X(I)=0.0
      Y(I)=0.0
      IF (SUM.EQ.0.0) GO TO 660
      Y(I)=100.0*Y1/SUM
      X(I)=100.0*X1/SUM
      IF (X(I).GE.0.0.AND.Y(I).GE.0.0) GO TO 660
      X(I)=0.0
      Y(I)=0.0
  660 CONTINUE
      LSTMT=LSTMT+LE1+8
      PRINT TFORM, HEADG
      PRINT 850, (STMT(I),I=8,LSTMT)
      PRINT 860
      DO 670 I=1,CNT
      Y1=Y(I)
      X1=X(I)
      SUM=SIGN(1.0,Y1)*(100.0-X1-Y1)
  670 PRINT 870, IDENT(I),Y1,SUM,X1,SYM(I)
      CALL TRIANG (IDENT,X,Y,CNT,LE1,LE2,LE3,SYM,STMT(8),STMT(K),STMT(J)
     1,     PRNTER,SCALE)
      GO TO 910
```

```
C
C     UNRECOGNIZED COMMAND * * * * * * * * * * * * * * * * * * * * * *
  680 WRITE(U,880) (STMT(I),I=1,LSTMT)
      GO TO 910
C
  690 IF (SWCC) CALL PLOTND
      STOP
C
  700 FORMAT ('1SCALE MUST BE POSITIVE AND LESS THAN OR EQUAL TO 3.0!)
  710 FORMAT ('1''GRAPHS=CALCOMP'' MUST BE SPECIFIED BEFORE SETTING SCA
     1E')
  720 FORMAT (' ',A8,' DEFINED AS ',80A1)
  730 FORMAT ('0',A8,' DEFINED AS ',80A1)
  740 FORMAT (80A1)
  750 FORMAT ('1NO MORE THAN ',I3,' NORMS MAY BE STORED. THE LAST NORM
     1WILL BE ERASED!)
  760 FORMAT ('1STATEMENT LENGTH(160) EXCEEDED ON THE FOLLOWING STATEMEN
     1T:'/(' ',80A1/),'0DID YOU FORGET A SEMICLON?')
  770 FORMAT ('1FORMAT COMMAND ERROR (MISSING PARENTHESIS) IN ',80A1)
  780 FORMAT ('1ERROR IN MODIFY COMMAND ON ',80A1)
  790 FORMAT ('1NO MORE THAN 14 NAMES MAY BE DEFINED. DEFINITION IGNORED
     1 FOR NAME=',A8)
  800 FORMAT (' EVALUATION OF ',40A1)
  810 FORMAT ('0   ID',7X,'VALUE',6X,'ID',7X,'VALUE',6X,'ID',7X,'VALUE
     16X,'ID',7X,'VALUE',6X,'ID',7X,'VALUE',6X,'ID',7X,'VALUE'/)
  820 FORMAT ('    ',A6,F11.3,3X,A6,F11.3,3X,A6,F11.3,3X,A6,F11.3,3X,A6,
     1F11.3,3X,A6,F11.3)
  830 FORMAT ('1PLOT COMMAND ERROR ON ',80A1)
  840 FORMAT ('1TERNARY COMMAND ERROR ON ',80A1)
  850 FORMAT (' TERNARY RATIOS FOR ',100A1)
  860 FORMAT ('0SAMPLE    R1',6X,'R2',6X,'R3   SYMBOL'/)
  870 FORMAT (' ',A6,3F8.2,4X,A1)
  880 FORMAT ('1UNRECOGNIZED COMMAND GIVEN AS:'/'0',120A1)
  890 FORMAT ('1OXIDE COMMAND CONTAINS A NAME WHICH IS NOT IN THE LIST O
     1F ACCEPTABLE OXIDES. ACCEPTABLE OXIDES ARE:'/' ',11A8/' ',10A8/'0T
     2HE UNRECOGNIZED NAME IS ',A8)
  900 FORMAT ('1THE FOLLOWING COMMAND CONTAINS EXCESSIVE CHARACTERS.'/
     1',120A1/'0DID YOU FORGET A SEMICOLON?')
      END
C
C THIS ROUTINE SETS UP THE I/O FOR GNAP OPENING FILES
C AND SWITCHING UNITS. JOHN ODELL, D.G.M.R.
C
 SUBROUTINE PREPRO(BATCH)
C
 BYTE FIL(32),BATCH
 INTEGER UN,U
 COMMON /COMM/ U,UN,FIL
C
 UN=5
 U=5
 TYPE 10
10 FORMAT(' GNAP -- BATCH OR TERMINAL? -----> ',$)
```

```
 ACCEPT 11,TCHAR
11 FORMAT(A4)
 IF (TCHAR.EQ.'BATC') GO TO 15
 TYPE 12
12 FORMAT(' OUTPUT TO TERMINAL OR PRINTER? ------> ',$)
 ACCEPT 11, TCHAR
 IF (TCHAR.EQ.'TERM') GO TO 25
 GO TO 28
25 CALL ASSIGN(6,'TI:')
 GO TO 28
15 TYPE 17
17 FORMAT(' NAME BATCH FILE: ------> ',$)
 ACCEPT 18, FIL
18 FORMAT(32A1)
 FIL(32)=0
 UN=2
 TYPE 26
26 FORMAT(' BATCH PROCESSOR TAKES CONTROL',//)
 IF (FIL(1).EQ.' ') GO TO 29
 OPEN (UNIT=UN,NAME=FIL,TYPE='OLD',READONLY)
 GO TO 27
29 OPEN (UNIT=UN,NAME='GNAP.BAT',TYPE='OLD',READONLY)
27 BATCH=.TRUE.
 U=6
28 RETURN
 END
C     ************************************************************************
C     NORM(CARD,OUTM,STORE,FORMAT,NCS,TTDI,RATIOS,TOTALS,TYPES,
C     NIGGLI,IDENT) IS USED TO CALCULATE THE STANDARD CIPW NORM. 'CARD'
C     IS AN 80 BYTE AREA WHICH CONTAINS ALL OR THE FIRST SEGMENT OF THE
C     OXIDE VALUES IN CHARACTER FORM. 'OUTM' IS AN ARRAY WHERE OXIDE
C     VALUES AND CALCULATED NORMATIVE MINERALS WILL BE STORED. 'STORE'
C     IS A LOGICAL VARIABLE WHICH INDICATES TO NORM WHEN JUST THE
C     STORAGE OF OXIDE VALUES AND SAMPLE IDS IS DESIRED. IT IS SET TO
C     .FALSE. UPON EXIT. 'FORMAT' GIVES THE OBJECT-TIME FORMAT UNDER
C     WHICH NORM WILL READ THE OXIDE VALUES FROM THE AREA CARD. 'NCS'
C     GIVES THE NUMBER OF INPUT CARDS REQUIRED FOR ONE SAMPLE ANALYSIS.
C     'TTDI' WILL CONTAIN THORNTON AND TUTTLE'S DIFFERENTIATION INDEX.
C     'RATIOS' IS AN ARRAY THAT WILL CONTAIN THE TWO RATIOS CALCULATED
C     BY NORM. 'TOTALS' IS AN ARRAY WHICH WILL CONTAIN THE TOTAL OXIDES
C     AND TOTAL NORMATIVE MINERALS. 'TYPES' IS AN ARRAY WHICH WILL
C     CONTAIN THE TOTALS OF THE NORMATIVE MINERALS FALLING IN THE SALIC
C     AND FEMIC GROUPS. 'OUTM(66+)' IS AN ARRAY WHICH WILL CONTAIN 21
C     BARTH CATIONS. 'NIGGLI' IS AN ARRAY WHICH WILL CONTAIN THE 12
C     NIGGLI VALUES. 'IDENT' WILL CONTAIN THE 5 CHARACTER SAMPLE ID.
C     'NOX' AND 'ORDER' GIVE THE NUMBER OF SELECTED OXIDES TO BE READ
C     AND THEIR ORDER. 'NONORM' IS A LOGICAL VARIABLE WHICH SUPPRESSES
C     PRINTOUT IF .TRUE.. 'ADJUST' CONTAINS THE NORMALIZED OXIDES.
C
C     THERE ARE TWO OTHER ENTRIES TO NORM (RECALC AND CONVER) WHICH ARE
C     COMMENTED UPON AT THEIR APPEARENCE.
C
C     DEC 1977 - ADAPTED BY LESTER NORTH, U.S.G.S. TO RUN ON THE D.G.M.R
```

```
C      PDP-11/45 COMPUTER UNDER RSX11-D WITH F4P COMPILER.
C
C      APR 1978 - MODIFICATIONS BY JOHN ODELL, D.G.M.R. TO D.G.M.R. AND
C      U.S.G.S. REQUIREMENTS IN SAUDI ARABIA.
C
C      ***********************************************************************
C
       SUBROUTINE NORM (CARD,OUTM,STORE,FORMAT,NCS,TTDI,RATIOS,TOTALS,
      1 TYPES,NIGGLI,IDENT,NOX,ORDER,NONORM,ADJUST,SYM,NAMES,CNT)
C
       IMPLICIT REAL(M,N)
C
       PARAMETER IV=20
C
       COMMON HEADG,TFORM,NORMAL,RASSYM
C
       COMMON /COMM/ U,UN,FIL,BAT
C
C      THESE COMMON BLOCKS ALIGN SCALARS AND VECTORS TO AVOID PARENTHESES
C
       COMMON /ALIGN1/ WC1,WC2,WC3,WC4,WC5,WC6,WC7,WC8,WC9,WC10,WC11,WC12
      1,WC13,WC14,WC15,WC16,WC17,WC18,WC19,WC20,WC21
       COMMON /ALIGN3/ MM1,MM2,MM3,MM4,MM5,MM6,MM7,MM8,MM9,MM10,MM11,MM12
      1,MM13,MM14,MM15,MM16,MM17,MM18,MM19,MM20,MM21,MM22,MM23,MM24,MM25,
      2MM26,MM27,MM28,MM29,MM30,MM31,MM32,MM33,MM34,MM35,MM36
       REAL K,MC(21),OUTM(100),WC(21),PM(33),NP(11),MM(36),PC(21),MG,VALU
      1ES(100,IV),RATIOS(2),TOTALS(2),TYPES(2),NIGGLI(12)
C
       REAL TTDI(1), RM(21),WM(33),ADJUST(21)
       REAL*8 NAMES(100)
C
       LOGICAL*1 CARD(80),SYM(1),IDENT(6),FORMAT(96),SAM,BUFFER(240),
      1 HEADG(75),TFORM(16),RASSYM,STORE,NONORM,NORMAL,BAT,FIL(32)
C
       INTEGER I,T,ORDER(21),NOX,NCS,UN,U,CNT
C
       EQUIVALENCE (WC1,WC(1)), (MM1,MM(1))
C
       DATA RM     /1.664314E-2,9.807652E-3,6.262047E-3,1.391858E-2,2.4806
      188E-2,1.783186E-2,1.61345E-2,1.061533E-2,5.550825E-2,1.251583E-2,7
      2.045002E-3,1.409694E-2,8.115645E-3,2.272213E-2,1.249029E-2,2.82063
      36E-1,5.263601E-2,3.118762E-2,6.579372E-3,1.33852E-2,6.521481E-3/
       DATA WM     /60.0848,101.9612,183.3036,556.6734,524.449,278.2102,43
      16.5038,284.1098,316.3342,116.8856,142.0412,105.989,462.0104,122.06
      238,154.2882,116.1642,100.3962,131.9312,140.7076,203.7776,172.2436,
      3231.5383,223.8363,159.6922,151.7449,196.063,135.9782,79.8988,336.2
      4084,78.0768,119.975,100.0894,84.32135/
C
 IF (BAT) GO TO 4006
C
            IF(NOTEQ(CARD( 5),'BATCH',5)) GO TO 1001
C
       IF (CARD(11).NE.' ') GO TO 4001
```

```
      OPEN(UNIT=2,NAME='GNAP.BAT',TYPE='OLD',READONLY)
      GO TO 4003
4001  DO 4009 I=11,42
4009  FIL(I-10)=CARD(I)
      FIL(32)=0
      OPEN(UNIT=2,NAME=FIL,TYPE='OLD',READONLY)
4003  BAT=.TRUE.
4006  READ(2,610,END=4010) CARD
C
        GO TO 1002
1001     IF(NOTEQ(CARD(12),'PROMPT',6)) GO TO 1002
          CALL MOVE(CARD(5),IDENT,6)
            SYM(1)=CARD(4)
          WRITE(U,2000) IDENT
2000      FORMAT('  SAMPLE ID ',8A1)
               DO 1003 I=1,NOX
            J=ORDER(I)
           WRITE(U,3000) NAMES(J)
            ACCEPT 3001,PM(I)
3000        FORMAT('  ',A8,'.......> 'S)
3001        FORMAT(F6.2)
1003        CONTINUE
          GO TO 1004
1002  IF(NORMAL) GO TO 4
      SYM(1)=RASSYM
      CALL MOVE(CARD(71),IDENT,6)
      GO TO 8
    4 CALL MOVE (CARD(5),IDENT,6)
      SYM(1)=CARD(4)
    8 CALL MOVE (CARD,BUFFER,80)
      IF (NCS.EQ.1) GO TO 20
      DO 10 I=2,NCS
C
      READ(2,610,END=4010) CARD
C
      J=80*I-79
   10 CALL MOVE (CARD,BUFFER(J),80)
 20    DECODF(240,FORMAT,BUFFER) (PM(I),I=1,NOX)
1004  DO 30 I=1,21
   30 PC(I)=0.0
      DO 40 I=1,NOX
      J=ORDER(I)
   40 PC(J)=PM(I)
      IF (STORE) GO TO 90
      DO 50 I=1,33
   50 PM(I)=0.0
      DO 60 I=1,11
   60 NP(I)=0.0
      GO TO 80
C     ***************************************************************
C     RECALC(OUTM,IDENT,SAM,ITDI,RATIOS,TOTALS,TYPES,NIGGLI) IS
C     USED TO RECALCULATE A NORM USING THE FIRST 21 VALUES IN THE ARRAY
C     'OUTM' AS THE INPUT OXIDE VALUES. 'IDENT' WILL ALREADY CONTAIN THE
```

```
C     5 CHARACTER SAMPLE ID. 'SAM' WILL ALREADY CONTAIN THE PLOTTING
C     SYMBOL WHICH WILL BE PRINTED. THE OTHER PARAMETERS ARE COMMENTED
C     UPON AT THE BEGINNING OF NORM (INCLUDING 'NONORM' AND 'ADJUST').
C     ****************************************************************
C
      ENTRY RECALC(OUTM,IDENT,SYM,TTDI,RATIOS,TOTALS,TYPES,NIGGLI,
     1 NONORM,ADJUST)
      DO 70 I=1,21
   70 PC(I)=OUTM(I)
   80 RATIO=0.0
      R1=0.0
      IF (PC(3).NE.0.0) RATIO=PC(4)/PC(3)
      IF (PC(1).NE.0.0) R1=PC(2)/PC(1)
      RATIOS(1)=R1
      RATIOS(2)=RATIO
   90 SUM=0.0
      DO 100 I=1,21
      OUTM(I)=PC(I)
  100 SUM=SUM+PC(I)
      TOTALS(1)=SUM
      IF (STORE) GO TO 570
      IF (NONORM) GO TO 110
      PRINT TFORM, HEADG
      PRINT 620, PC,SUM,IDENT,SYM
  110 SUM=100.0/SUM
      DO 120 I=1,21
      PC(I)=SUM*PC(I)
      ADJUST(I)=PC(I)
      MC(I)=PC(I)*RM(I)
  120 WC(I)=MC(I)
      IF (NONORM) GO TO 130
      PRINT 630, (PC(I),I=1,11),R1,(MC(I),I=1,11),(PC(I),I=12,21),RATIO,
     1(MC(I),I=12,21)
  130 DO 140 I=1,36
  140 MM(I)=0.0
      WC4=WC4+WC12+WC20
      WC6=WC6+WC21-3.33333*WC11
      IF (WC6.GE.0.0) GO TO 150
      T=2
      GO TO 470
  150 MM29=WC11
      WC7=WC7-0.5*WC16
      IF (WC7.GE.0.0) GO TO 160
      T=3
      GO TO 470
  160 MM10=0.5*WC16
      MM11=AMIN1(WC7,WC15)
      WC7=WC7-MM11
      WC15=WC15-MM11
      WC4=WC4-0.5*(WC18+WC15)
      IF (WC4.GE.0.0) GO TO 170
      T=5
      GO TO 470
```

```
170 MM31=0.5*(WC18+WC15)
    WC4=WC4-WC19
    IF (WC4.GE.0.0) GO TO 180
    T=6
    GO TO 470
180 MM23=WC19
    MM25=AMIN1(WC4,WC10)
    WC4=WC4-MM25
    WC10=WC10-MM25
    WC17=WC17-0.666667*MM29
    IF (WC17.LT.0.0) WC17=0.0
    WC6=WC6-0.5*WC17
    IF (WC6.GE.0.0) GO TO 190
    T=9
    GO TO 470
190 MM30=0.5*WC17
    MM32=AMIN1(WC6,WC14)
    WC6=WC6-MM32
    WC14=WC14-MM32
    WC5=WC5-WC14
    IF (WC5.GE.0.0) GO TO 200
    T=11
    GO TO 470
200 MM17=WC5
    MM33=WC14
    WC1=WC1-WC13
    IF (WC1.GE.0.0) GO TO 210
    T=12
    GO TO 470
210 MM3=WC13
    DIFF=WC2-WC8
    IF (DIFF.GE.0.0) GO TO 220
    MM4=WC2
    MM15=-DIFF
    WC2=0.0
    WC8=0.0
    GO TO 250
220 MM4=WC8
    WC2=DIFF
    DIFF=WC2-WC7
    WC8=0.0
    IF (DIFF.GE.0.0) GO TO 230
    MM5=WC2
    WC7=-DIFF
    WC2=0.0
    GO TO 250
230 MM5=WC7
    WC2=DIFF
    DIFF=WC2-WC6
    WC7=0.0
    IF (DIFF.GE.0.0) GO TO 240
    MM6=WC2
    WC6=-DIFF
```

```
      WC2=0.0
      GO TO 250
240   MM6=WC6
      MM2=DIFF
      WC2=0.0
      WC6=0.0
250   DIFF=WC10-WC6
      IF (DIFF.GE.0.0) GO TO 260
      MM26=WC10
      WC6=-DIFF
      GO TO 270
260   MM26=WC6
      MM28=DIFF
      WC6=0.0
270   DIFF=WC3-WC7
      IF (DIFF.GE.0.0) GO TO 280
      MM13=WC3
      MM14=-DIFF
      MM18=WC4
      GO TO 300
280   MM13=WC7
      WC3=DIFF
      WC7=0.0
      DIFF=WC3-WC4
      IF (DIFF.GE.0.0) GO TO 290
      MM22=WC3
      MM18=-DIFF
      GO TO 300
290   MM22=WC4
      MM24=DIFF
300   SUMMF=MM17+MM18
      R1=0.0
      R2=0.0
      IF (SUMMF.LE.0.0) GO TO 310
      R1=MM17/SUMMF
      R2=MM18/SUMMF
310   DIFF=SUMMF-WC6
      IF (DIFF.LT.0.0) GO TO 320
      MM34=WC6
      MM35=DIFF
      GO TO 330
320   MM34=SUMMF
      MM16=-DIFF
330   WC1=WC1-MM26-4.0*MM13-MM14-MM15-6.0*(MM4+MM5)-MM16-2.0*(MM6+MM34)-
     1MM35
      IF (WC1.LT.0.0) GO TO 340
      MM1=WC1
      GO TO 420
340   WC1=WC1+MM35
      DIFF=2.0*WC1-MM35
      IF (DIFF.LT.0.0) GO TO 350
      MM36=MM35-WC1
      MM35=DIFF
```

```
      GO TO 420
350   MM36=0.5*MM35
      MM35=0.0
      WC1=WC1-MM36+MM26
      IF (WC1.LT.0.0) GO TO 360
      MM27=MM26-WC1
      MM26=WC1
      GO TO 420
360   WC1=WC1+6.0*MM5
      MM27=MM26
      MM26=0.0
      DIFF=WC1-2.0*MM5
      IF (DIFF.LT.0.0.OR.WC1.GT.6.0*MM5) GO TO 370
      MM8=1.5*MM5-0.25*WC1
      MM5=0.25*DIFF
      GO TO 420
370   MM8=MM5
      MM5=0.0
      WC1=WC1-2.0*MM8+6.0*MM4
      DIFF=WC1-4.0*MM4
      IF (DIFF.LT.0.0.OR.WC1.GT.6.0*MM4) GO TO 380
      MM7=3.0*MM4-0.5*WC1
      MM4=0.5*DIFF
      GO TO 420
380   MM7=MM4
      MM4=0.0
      WC1=WC1-4.0*MM7+MM16
      DIFF1=MM16-WC1
      DIFF2=2.0*WC1-MM16
      IF (DIFF1.LT.0.0.OR.DIFF2.LT.0.0) GO TO 390
      MM21=DIFF1
      MM16=DIFF2
      GO TO 420
390   WC1=WC1+2.0*MM34
      FAC1=2.0*(WC1-MM34)-MM16
      FAC2=4.0*MM34+MM16-2.0*WC1
      FAC3=FAC2+2.0*MM16
      IF (FAC1.LT.0.0.OR.FAC2.LT.0.0.OR.FAC3.LT.0.0) GO TO 400
      MM21=0.25*FAC3
      MM36=MM36+0.25*FAC2
      MM34=0.5*FAC1
      MM16=0.0
      GO TO 420
400   MM36=MM36+0.5*MM34
      MM21=0.5*(MM16+MM34)
      WC1=WC1-MM34-0.5*MM16+4.0*MM7
      MM34=0.0
      MM16=0.0
      DIFF1=WC1-2.0*MM7
      DIFF2=WC1-2.0*DIFF1
      IF (DIFF1.GE.0.0.AND.DIFF2.GE.0.0) GO TO 410
      T=31
      GO TO 470
```

```
410 MM9=0.5*DIFF2
    MM7=0.5*DIFF1
420 DI=MM34
    DIWO=DI
    DIEN=R1*DI
    DIFS=R2*DI
    HY=MM35
    HYEN=R1*HY
    HYFS=R2*HY
    OL=MM36
    OLFO=R1*OL
    MM19=OLFO
    OLFA=R2*OL
    MM20=OLFA
    MM16=MM16+DIWO
    MM17=DIEN+HYEN
    MM18=DIFS+HYFS
    DO 430 I=1,33
430 PM(I)=WM(I)*MM(I)
    IF (NONORM) GO TO 440
    PRINT 640, (MM(I),I=1,12),(PM(I),I=1,12),(MM(I),I=13,24),(PM(I),I=
   113,24),(MM(I),I=25,33)
440 DIFF2=0.0
    DO 450 I=1,12
450 DIFF2=DIFF2+PM(I)
    DIFF1=DIFF2
    DO 460 I=13,33
460 DIFF1=DIFF1+PM(I)
    Y=DIFF1-DIFF2
    NP(2)=DIWO*WM(16)
    NP(3)=DIEN*WM(17)
    NP(4)=DIFS*WM(18)
    NP(1)=NP(2)+NP(3)+NP(4)
    NP(6)=HYEN*WM(17)
    NP(7)=HYFS*WM(18)
    NP(5)=NP(6)+NP(7)
    NP(9)=OLFO*WM(19)
    NP(10)=OLFA*WM(20)
    WOL=MM16-DIWO
    NP(11)=WOL*WM(16)
    NP(8)=NP(9)+NP(10)
    TOTALS(2)=DIFF1
    TYPES(1)=DIFF2
    TYPES(2)=Y
    TTDI(1)=PM(1)+PM(4)+PM(5)+PM(7)+PM(8)+PM(9)
    IF (NONORM) GO TO 480
    PRINT 650, (PM(I),I=25,33),DIFF1,DIFF2,Y,DI,DIWO,DIEN,DIFS,HY,HYEN
   1,HYFS,OL,OLFO,OLFA,WOL,(NP(I),I=1,11)
    PRINT 660, TTDI
    GO TO 480
470 IF (NONORM) PRINT 670
    PRINT 680, IDENT,T
480 DO 490 I=1,21
```

```
490 WC(I)=MC(I)
    WC2=2.0*WC2
    WC3=2.0*WC3
    WC7=2.0*WC7
    WC8=2.0*WC8
    WC9=2.0*WC9
    WC11=2.0*WC11
    WC19=2.0*WC19
    SUM=0.0
    DO 500 I=1,21
500 SUM=SUM+WC(I)
    SUM=SUM-WC9-WC16-WC17-WC18
    SUM=100.0/SUM
    DO 510 I=1,21
    WC(I)=SUM*WC(I)
    KK=I+65
510 OUTM(KK)=WC(I)
    IF (NONORM) GO TO 520
    PRINT 690, WC
520 DO 530 I=1,21
530 WC(I)=MC(I)
    AL=WC2+WC19
    FM=2.0*WC3+WC4+WC5+WC12+WC20
    C=WC6+WC21
    ALK=WC7+WC8
    SUM=100.0/(AL+FM+C+ALK)
    AL=SUM*AL
    NIGGLI(1)=AL
    FM=SUM*FM
    NIGGLI(2)=FM
    C=SUM*C
    NIGGLI(3)=C
    ALK=SUM*ALK
    NIGGLI(4)=ALK
    SI=SUM*WC1
    NIGGLI(5)=SI
    TI=SUM*WC10
    NIGGLI(6)=TI
    P=SUM*WC11
    NIGGLI(7)=P
    H=SUM*WC9
    NIGGLI(8)=H
    K=0.0
         DIFF1=WC7+WC8
    IF (DIFF1.NE.0.0) K=WC8/DIFF1
    MG=0.0
    SIP=100.0+4.0*ALK
    IF (FM.NE.0.0) MG=SUM*WC5/FM
    IF(ALK.GT.AL) SIP=100.+3*AL+ALK
    QZ=SI-SIP
    NIGGLI(9)=K
    NIGGLI(10)=MG
    NIGGLI(11)=SIP
```

```
      NIGGLI(18)=QZ
      IF (NONORM) GO TO 540
      PRINT 700, AL,FM,C,ALK,SI,TI,P,H,K,MG,QEP,QZ
  540 DO 550 I=1,33
  550 OUTM(I+21)=PM(I)
      DO 560 I=1,11
  560 OUTM(I+54)=NP(I)
570           CONTINUE
      RETURN
C
4010     CONTINUE
 CNT=CNT-1
        IF (BAT) CLOSE (UNIT=2)
        BAT=.FALSE.
 RETURN
C
C     ***************************************************************************
C     CONVER(VALUES,CNT) IS USED TO CONVERT VALUES OF OXIDES AND NORMATI
C     MINERALS FOR THE 'CNT' SAMPLES STORED IN 'VALUES' FROM PERCENTAGES
C     TO MOLECULAR AMOUNTS.
C     ***************************************************************************
      ENTRY CONVER(VALUES,CNT)
      DO 600 I=1,CNT
      DO 580 J=1,21
  580 VALUES(J,I)=VALUES(J,I)*RM(J)
      DO 590 J=22,54
  590 VALUES(J,I)=VALUES(J,I)/WM(J-21)
      X1=VALUES(56,I)/WM(16)
      VALUES(56,I)=X1
      Y1=VALUES(57,I)/WM(17)
      VALUES(57,I)=Y1
      SUM=VALUES(58,I)/WM(18)
      VALUES(58,I)=SUM
      VALUES(55,I)=X1+Y1+SUM
      X1=VALUES(60,I)/WM(17)
      VALUES(60,I)=X1
      Y1=VALUES(61,I)/WM(18)
      VALUES(61,I)=Y1
      VALUES(59,I)=X1+Y1
      X1=VALUES(63,I)/WM(19)
      VALUES(63,I)=X1
      Y1=VALUES(64,I)/WM(20)
      VALUES(64,I)=Y1
      VALUES(62,I)=X1+Y1
  600 VALUES(65,I)=VALUES(65,I)/WM(16)
      RETURN
C
  610 FORMAT (80A1)
  620 FORMAT (' ORIGINAL WT.PCT. OXIDES'/'0  SIO2 AL203 FE203   FEO   ',
     1'MGO   CAO   NA20   K20   H20  TIO2  P205   MNO  ZRO2   CO2   ','SO
     23    CL     F     S CR203   NIO   BAO'/' ',21F6.2/'0SUM OF ORIGINA
     3L OXIDES=',F6.2/'0CIPW NORM FOR SAMPLE NO. ',6A1,69X,'PLOTTING SYM
     480L IS ',A1)
```

```
  630 FORMAT ('0CONSTITUENTS    SIO2    AL2O3    FE2O3     FEO',6X,'MGO
     1    CAO',6X,'NA2O     K2O',6X,'H2O',6X,'TIO2',5X,'P2O5 AL2O3/SIO2'
     2/' PERCENTAGES',11F9.2,F10.3/' MOL. AMTS. ',11F9.4/'0CONSTITUENTS
     3   MNO',6X,'ZRO2     CO2',6X,'SO3',7X,'CL',6X,'F',8X,'S',6X,'CR2O3
     4     NIO     BAO',12X,'FEO/FE2O3'/' PERCENTAGES',10F9.2,F19.3/
     5' MOL. AMTS. ',10F9.4)
  640 FORMAT (/'0MINERALS',9X,'Q',8X,'C',8X,'Z',8X,'OR',7X,'AB',7X,'AN',
     17X,'LC',7X,'NE',7X,'KP',7X,'HL',7X,'TH',7X,'NC'/' MOL. AMTS. ',12F
     29.4/' PERCENTAGES',12F9.3/'0MINERALS',9X,'AC',7X,'NS',7X,'KS',7X,'
     3WO',7X,'EN',7X,'FS',7X,'FO',7X,'FA',7X,'CS',7X,'MT',7X,'CM',7X,'HM
     4'/' MOL. AMTS. ',12F9.4/' PERCENTAGES',12F9.3/'0MINERALS',9X,'IL',
     57X,'TN',7X,'PF',7X,'RU',7X,'AP',7X,'FR',7X,'PR',7X,'CC',7X,'MG
     6 TOTAL    SALIC    FEMIC'/' MOL. AMTS. ',9F9.4)
  650 FORMAT (' PERCENTAGES',12F9.3/'0MINERALS',9X,'DI     DI-WO',4X,'DI
     1-EN    DI-FS      HY     HY-EN    HY-FS      OL     OL-FO',4X,'OL-
     2FA     WOL'/' MOL. AMTS. ',11F9.4/' PERCENTAGES',11F9.3)
  660 FORMAT ('0THORNTON + TUTTLE DIFFERENTIATION INDEX = ',F7.3)
  670 FORMAT ('1')
  680 FORMAT ('0NORM NOT COMPUTABLE FOR SAMPLE NO. ',6A1/'0SEE ERROR COD
     1E ',I3,' OF PROGRAM WRITE-UP')
  690 FORMAT ('0BARTHS CATIONS   SI       AL     FE+3    FE+2     MG',6X,
     1'CA      NA',7X,'K',7X,'H',6X,'TI',7X,'P',6X,'MN'/' ',13X,12F8.2/'
     20',25X,'ZR',7X,'C',6X,'S1',6X,'CL',7X,'F',6X,'S2',6X,'CR',6X,'NI',
     36X,'BA'/' ',21X,9F8.2)
  700 FORMAT ('0NIGGLI VALUES    AL*     FM*     C*      ALK*    SI',6X,
     1'RI',7X,'P',7X,'H',7X,'K',6X,'MG',6X,'SI''     QZ'/' ',13X,12F8.2)
      END
C     ***************************************************************
C     TRIANG(IDENT,X,Y,CNT,LE1,LE2,LE3,SYM,E1,E2,E3,PAGE) IS USED TO
C     CONSTRUCT A TERNARY DIAGRAM WHOSE APEXES ARE GIVEN BY THE
C     EXPRESSIONS 'E1','E2' AND 'E3' WHOSE RESPECTIVE LENGTHS ARE 'LE1',
C     'LE2' AND 'LE3'. THE ARRAYS 'X' AND 'Y' WERE OBTAINED BY
C     EVALUATION OF EXPRESSIONS E3 AND E1 RESPECTIVELY. 'IDENT' IS AN
C     ARRAY GIVING THE SAMPLE IDS FOR EACH POINT IN THE TWO ARRAYS.
C     'SYM' IS AN ARRAY GIVING THE PLOTTING SYMBOL FOR EACH OF THE 'CNT'
C     POINTS IN X AND Y. 'PAGE' IS AN AREA USED FOR PLOTTING.
C     'PRNTER' AND 'SIZE' ARE COMMENTED UPON AT THE BEGINNING OF PRNT.
C     ***************************************************************
C
      SUBROUTINE TRIANG(IDENT,X,Y,CNT,LE1,LE2,LE3,SYM,E1,E2,E3,PRNTER,
     X SIZE)
C
      INTEGER CNT
      BYTE SYM(CNT),E1(LE1),E2(LE2),E3(LE3),PAGE(101,51),PRNTER
      BYTE BLNK,STAR,ZERO,MOD5,CHAR(4),HEADG(75),TFORM(16),LINK(8)
      DIMENSION X(1),Y(1)
      REAL*8 IDENT(CNT),BAD(45),MORE
C
      COMMON HEADG,TFORM
C
      DATA BLNK,ZERO,STAR,MORE/' ','+','-','+MORE'/
      DATA CHAR/4*"000/
C
```

```
      LINK(7)=1/'
C
      IF (PRNTER) GO TO 70
C     CONSTRUCT TRIANGLE ON CALCOMP PLOTTER
      X1=-0.12*LE2
      CALL SYMBOL (X1,-0.3,0.21,E2,0.0,LE2)
      CALL SIDE (0.0,0.0,0.0)
      X1=9.08-0.12*LE3
      CALL SYMBOL (X1,-0.3,0.21,E3,0.0,LE3)
      CALL SIDE (120.0,9.08,0.0)
      X1=4.54-0.12*LE1
      CALL SYMBOL (X1,7.96,0.21,E1,0.0,LE1)
      CALL SIDE (-120.0,4.54,7.86)
C     PLOT THE POINTS
      DO 60 I=1,CNT
      Y1=7.864E-2*Y(I)
      X1=9.08E-2*X(I)+Y1/1.7321
      IF (X1.NE.0.0.OR.Y1.NE.0.0) CALL SYMBOL (X1-0.05,Y1-0.07,
     X 0.356/SIZE,SYM(I),0.0,1)
   60 CONTINUE
      CALL PLOT (15.0,0.0,-3)
      RETURN
   70 CONTINUE
C     SUPERIMPOSE TRIANGLE ONTO THE PLOTTING SURFACE PAGE(,).
      DO 100 I=2,50
      LF=51-I
      DO 80 J=1,LF
   80 PAGE(J,I)=BLNK
      LF=LF+1
      PAGE(LF,I)=STAR
      MOD5=MOD(I-1,5).EQ.0
      IF (MOD5) PAGE(LF,I)=ZERO
      M=2*I-3
      DO 90 J=1,M
      LF=LF+1
   90 PAGE(LF,I)=BLNK
      LF=LF+1
      PAGE(LF,I)=STAR
      IF (MOD5) PAGE(LF,I)=ZERO
      LF=LF+1
      DO 100 J=LF,101
  100 PAGE(J,I)=BLNK
      DO 110 J=1,101
      PAGE(J,51)=BLNK
  110 PAGE(J,1)=BLNK
      PAGE(1,51)=STAR
      PAGE(51,1)=STAR
      DO 120 J=3,99,2
      PAGE(J,51)=STAR
  120 IF (MOD(J,10).EQ.1) PAGE(J,51)=ZERO
      PAGE(101,51)=STAR
      K=0
C     START PLOTTING THE POINTS.
```

```
      DO 150 I=1,CNT
      XT=X(I)
      YT=Y(I)
      IF (XT.EQ.0.0.AND.YT.EQ.0.0) GO TO 150
      J=0.5*YT+0.5
      J=51-J
      IX=XT+0.5*YT+1.5
      IF (XT.LT.1.0.OR.YT.LT.1.0.OR.XT+YT.GT.99.0) GO TO 140
      CHAR(4)=PAGE(IX,J)
            NPT=CHAR(4)
C     TEST FOR BLANK
      IF (NPT.EQ.32) GO TO 140
C     TEST FOR GRID SYMBOLS + OR -.
      IF(NPT .EQ. 43) GO TO 140
      IF(NPT .EQ. 45) GO TO 140
      K=K+1
      IF (K.LE.44) GO TO 130
      K=45
      BAD(36)=MORE
      GO TO 150
130   CALL MOVE (IDENT(I),LINK,6)
      LINK(8)=CHAR(4)
      CALL MOVE(LINK,BAD(K),8)
      GO TO 150
  140 PAGE(IX,J)=SYM(I)
  150 CONTINUE
      IF (K.EQ.0) GO TO 170
      CALL MOVE ('THE FOLLOWING SAMPLES WERE NOT PLOTTED',PAGE,38)
      CALL MOVE ('BECAUSE THEY WOULD HAVE FALLEN ON A',PAGE(1,2),35)
      CALL MOVE ('PREVIOUSLY PLOTTED POINT (/N):',PAGE(1,3),30)
      M=3
      DO 160 I=1,K
      J=MOD(I-1,3)
      IF (J.EQ.0) M=M+1
      LF=10*J+1
  160 CALL MOVE (BAD(I),PAGE(LF,M),8)
  170 PRINT TFORM,HEADG
      PRINT 180, (E1(I),I=1,LE1)
      DO 210 J=1,51
      PRINT 190,(PAGE(I,J),I=1,101)
  210 CONTINUE
      PRINT 190, (E2(I),I=1,LE2)
      PRINT 200, (E3(I),I=1,LE3)
      RETURN
C
  180 FORMAT (' ',55X,40A1)
  190 FORMAT (' ', 5X,101A1)
  200 FORMAT ('+',105X,27A1)
      END
C     ***************************************************************************
C     PRNT(X,Y,SYM,CNT,E1,E2,LE1,LE2,STAND,PAGE) IS USED TO PLOT THE
C     'CNT' POINTS IN THE ARRAYS 'X' AND 'Y', THESE ARRAYS WERE OBTAINED
C     BY EVALUATION OF THE EXPRESSIONS 'E1' AND 'E2' WHOSE RESPECTIVE
```

```
C     LENGTHS ARE 'LE1' AND 'LE2'. 'STAND' IS A LOGICAL VARIABLE WHICH
C     TELLS PRNT THE POSITIVE DIRECTION OF THE X-AXIS. IF TRUE, X WILL
C     INCREASE TO THE RIGHT. 'PAGE' IS AN AREA WHICH WILL BE USED IN THE
C     PLOTTING. 'SYM' IS AN ARRAY OF CNT PLOTTING SYMBOLS, ONE FOR EACH
C     POINT IN X AND Y. 'PRNTER' IS A LOGICAL VARIABLE GIVING THE
C     PLOTTING DEVICE (.TRUE. IS LINEPRINTER). 'SIZE' IS USED TO FIX THE
C     PLOTTING POINT SIZE AT 0.14".
C     ****************************************************************
C
      SUBROUTINE PRNT (X,Y,SYM,CNT,E1,E2,LE1,LE2,STAND,     PRNTER,
     X SIZE)
      COMMON HEADG,TFORM
      INTEGER CNT
      LOGICAL*1 SYM(CNT),E1(LE1),E2(LE2),STAND,PAGE(101,51),PRNTER,
     X HEADG(75),TFORM(16)
      LOGICAL *1 IMAGE(5151)
      DIMENSION X(  1),Y(  1)
      EQUIVALENCE (PAGE,IMAGE)
C
      IF(PRNTER) GO TO 5
      CALL SCALE(Y,8.0,CNT,1)
      YMIN=Y(CNT+1)
      YMAX=Y(CNT+2)
      CALL SCALE(X,10.,CNT,1)
      XMIN=X(CNT+1)
      XMAX=X(CNT+2)
      IF (STAND) GO TO 40
      XT=10.*XMAX+XMIN
      DO 8 I=1,CNT
8     X(I)=XMIN+XT-X(I)
      GO TO 40
    5 XMAX=X(1)
      XMIN=XMAX
      YMAX=Y(1)
      YMIN=YMAX
      DO 10 I=2,CNT
      XT=X(I)
      YT=Y(I)
      IF (XT.LT.XMIN) XMIN=XT
      IF (XT.GT.XMAX) XMAX=XT
      IF (YT.LT.YMIN) YMIN=YT
      IF (YT.GT.YMAX) YMAX=YT
   10 CONTINUE
   30 IF (STAND) GO TO 40
      XT=XMAX
      XMAX=XMIN
      XMIN=XT
   40 CALL PLOT2 (IMAGE,XMAX,XMIN,YMAX,YMIN,PRNTER,SIZE)
      DO 50 I=1,CNT
   50 CALL PLOT3 (SYM(I),X(I),Y(I))
      CALL PLOT4 (LE2,E2,LE1,E1,STAND)
      RETURN
      END
```

```
C     ************************************************************************
C     PRPLOT IS A MODIFIED VERSION OF THE ROUTINE BY THE SAME NAME
C     WRITTEN AT  UNIV. OF MICHIGAN. SEE PROGRAM DOCUMENTATION FOR
C     DETAILS.
C     ************************************************************************
C
C      THIS VERSION MODIFIED BY JOHN ODELL D.G.M.R. FOR GNAP
C
      SUBROUTINE PRPLOT
      IMPLICIT LOGICAL*1(W),LOGICAL*1(K)
      COMMON HEADG,TFORM
C
      COMMON /COMM/ U
C
      DIMENSION ABNOS(11)
       LOGICAL*1 NOS(10),IMAGE(5152),CH,LABEL(1),XLAB(1)
       LOGICAL*1 HC,NC,BL,HF,HF1,FOR1(19),FOR2(15),FOR3(19)
       LOGICAL*1 CHAR(4),HEADG(75),TFORM(16),STAND
C
 INTEGER U
C
      EQUIVALENCE (NPT,CHAR(1))
      DATA       NOS    /'0','1','2','3','4','5','6','7','8','9'/
     1                        ,HC/'-'/,NC/'+'/,BL/' '/,HF/'F'/,HF1/'.'/
     3                  ,FOR2    /'(','1','X','A','1',',',' ','9','X','1',
     4'2','1','A','1',')'/,FOR3    /'(','1','H','0','F',' ',' ','.',' ',
     5',',' ',' ','F',' ',' ',' ','.',' ',')'/
         DATA FOR1/'(','1','X','A','1',',','F','8','.','0',',',
     1'1','X','1','2','1','A','1',')'/
      DATA KPLOT1/.FALSE./,KPLOT2/.FALSE./
C
  110 NVP=NV+1
      NDH=5*NSH
      NDHP=NDH+1
      NDV=NV*NSV
      NDVP=NDV+1
      NIMG=(NDHP*NDVP)
      FOR1(10)=NOS(4)
      NA=MIN0(4,NSV)-1
      NB=11+MIN0(NA,120-NDV)
      I1=NB/10
      I2=NB-I1*10
      FOR3(6)=NOS(I1+1)
      FOR3(7)=NOS(I2+1)
      FOR3(9)=NOS(NA+1)
      I1=NV/10
      I2=NV-I1*10
      FOR3(11)=NOS(I1+1)
      FOR3(12)=NOS(I2+1)
      FOR3(13)=HF
      I1=NSV/100
      I3=NSV-I1*100
      I2=I3/10
```

```
      I3=I3-I2*10
      FOR3(14)=NOS(I1+1)
      FOR3(15)=NOS(I2+1)
      FOR3(16)=NOS(I3+1)
      FOR3(17)=HF1
      FOR3(18)=FOR3(9)
      KPLOT1=.TRUE.
      GO TO 120
C
      ENTRY PLOT2(IMAGE,XMAX,XMIN,YMAX,YMIN,WPRNT,SIZE)
      IF (WPRNT) GO TO 120
      WPLOT=.FALSE.
      SCALE=SIZE
      XMIN1=XMIN
      YMIN1=YMIN
      DV=XMAX
      DH=YMAX
      IF(ABS(DV-XMIN1).LT.1.E-6) DV=XMIN1+1.0
      IF(ABS(DH-YMIN1).LT.1.E-6) DH=YMIN1+1.0
      RETURN
  120 KPLOT2=.TRUE.
      WPLOT=.TRUE.
      IF (KPLOT1) GO TO 130
      NSH=10
      NV=10
      NSV=10
      GO TO 110
  130 CONTINUE
      YMX=YMAX
      DH=(YMAX-YMIN)/      (NDH)
      DV=(XMAX-XMIN)/      (NDV)
      DO 140 I=1,NVP
  140 ABNOS(I)=(XMIN+      ((I-1)*NSV)*DV)
      NIMG2=NIMG+NIMG
      DO 150 I=1,NIMG
  150 IMAGE(I)=BL
      DO 180 I=1,NDHP
      I2=I*NDVP
      I1=I2-NDV
      KNHOR=MOD(I-1,NSH).NE.0
      IF (KNHOR) GO TO 170
      DO 160 J=I1,I2
  160 IMAGE(J)=HC
  170 CONTINUE
      DO 185 J=I1,I2,NSV
      IF (KNHOR) GO TO 185
      IMAGE(J)=NC
185      CONTINUE
  180 CONTINUE
      XMIN1=XMIN-DV/2.
      YMIN1=YMIN-DH/2.
      RETURN
C
```

```
      ENTRY PLOT3(CH,X,Y)
      DUM1=(X-XMIN1)/DV
      DUM2=(Y-YMIN1)/DH
      IF (WPLOT) GO TO 190
      CALL SYMBOL (DUM1-0.05,DUM2-0.07,0.356/SCALE,CH,0.0,1)
      RETURN
  190 J=(NDHP-INT(DUM2)-1)*NDVP+INT(DUM1)+1
      CHAR(4)=IMAGE(J)
      IF(NPT .EQ. 64) GO TO 191
      IF(NPT .EQ. 78) GO TO 191
      IF(NPT .EQ. 96) GO TO 191
      I=(J-1)/NDVP+1
  191 IMAGE(J)=CH
      RETURN
C
      ENTRY PLOT4(NL,LABEL,NXL,XLAB,STAND)
      IF (WPLOT) GO TO 200
      CALL AXIS (0.0,0.0,LABEL,NL,7.0,90.0,YMIN1,DH)
C        THIS CODE IS NECESSARY TO REVERSE THE AXIS
C
      IF (STAND) GO TO 198
C
      YP=-0.1
      CALL PLOT(0.,YP,3)
      DO 300 I=1,10
      XP=I-1
      XP2=I
      CALL PLOT(XP,0.,2)
      CALL PLOT(XP2,0.,2)
      CALL PLOT(XP2,YP,2)
300   CONTINUE
      YP=-0.25
      FP=XMIN1
      DO 310 I=1,11
      XP=11-I-0.1
      CALL NUMBER(XP,YP,0.1,FP,0.,2)
      FP=FP+DV
310   CONTINUE
      YP=-0.5
      XP=5.
      CALL SYMBOL(XP,YP,0.14,XLAB,0.,NXL)
      GO TO 199
198   CALL AXIS (0.0,0.0,XLAB,-NXL,10.0,0.0,XMIN1,DV)
199   CALL PLOT (15.0,0.0,-3)
      RETURN
  200 CONTINUE
      PRINT TFORM,HEADG
      DO 220 I=1,NDHP
      NL=BL
      IF (I.LE.NL) NL=LABEL(I)
      I2=I*NDVP
      I1=I2-NDV
      IF (MOD(I-1,NSH).EQ.0) GO TO 210
```

```
      PRINT FOR2, WL,(IMAGE(J),J=I1,I2)
      GO TO 220
  210 CONTINUE
      ORDNO=(YMX-      (I-1)*DH)
      PRINT FOR1, WL,ORDNO,(IMAGE(J),J=I1,I2)
  220 CONTINUE
      PRINT FOR3, (ABNOS(J),J=1,NVP)
      PRINT 240, (XLAB(J),J=1,NXL)
      RETURN
C
  240 FORMAT (' ',50X,70A1//)
      END
C     ************************************************************************
C     CONV(A,N1,N2,E,ERR) RETURNS THE NUMERIC VALUE OF A DECIMAL NUMBER
C     IN CHARACTER FORM IN THE STRING A. A(N1) IS THE LEADING DIGIT AND
C     A(N2) IS THE TRAILING DIGIT. A IS ASSUMED TO BE AT LEAST N2 BYTES
C     LENGTH. E IS USED AS IT IS IN THE F(W,E) FORMAT CODE (IE, IF NO
C     DECIMAL POINT APPEARS IT WILL BE ASSUMED IN THE E POSITION. IF PRE
C     A DECIMAL POINT OVERRIDES E). ERR IS A LOGICAL VARIABLE INDICATING
C     WHETHER OR NOT THE CONVERSION WAS SUCCESSFUL.
C     ************************************************************************
C
      FUNCTION CONV (A,N1,N2,E,ERR)
      LOGICAL*1 A(1),CHAR(4)         ,ERR
      INTEGER D,E
            DATA CHAR/4*'000/
      VALUE=0.0
      D=E
      ERR=.FALSE.
      DO 20 I=N1,N2
      CHAR(4)=A(I)
           INDEX=CHAR(4)
C     IF A(I) IS BLANK, TREAT IT AS A ZERO.
      IF (INDEX.EQ.32) INDEX=48
C     IF A(I) IS A DECIMAL POINT, COMPUTE D (THE POWER OF TEN FACTOR).
      IF (INDEX.NE.46) GO TO 10
      D=I-N2
      GO TO 20
   10 DIGIT=INDEX- 48
      IF (DIGIT.LT.0.0.OR.DIGIT.GT.9.0) GO TO 40
      VALUE=10.0*VALUE+DIGIT
   20 CONTINUE
   30 CONV=VALUE*10.0**D
      RETURN
   40 ERR=.TRUE.
      VALUE=0.0
      GO TO 30
      END
      SUBROUTINE SIDE (T,X,Y)
      CT=COS(1.74533E-2*T)
      ST=SIN(1.74533E-2*T)
      DO 10 I=1,10
      AL=0.908*(I-1)
```

```
      BL=AL+0.908
      IH=2
      IF (I.EQ.1) IH=3
      CALL PLOT(X+AL*CT-0.086603*ST-0.05*CT,Y+AL*ST+0.086603*CT-0.05*ST,
     X 3)
      CALL PLOT(X+AL*CT,Y+AL*ST,IH)
      CALL PLOT(X+BL*CT,Y+BL*ST,2)
      IF(I .LE. 9) CALL PLOT(X+BL*CT-0.086603*ST+0.05*CT,Y+BL*ST+0.05*ST
     X +0.086603*CT,2)
   10 CONTINUE
      RETURN
      END
C     ************************************************************************
C     THIS SUBROUTINE IS A COLLECTION OF ENTRIES DESIGNED TO FACILITATE
C     PRINTING OF LISTS IN COLUMN-WISE RATHER THAN ROW-WISE FASHION.
C     'WIDTH' AND 'LENGTH' ARE INTEGERS WHICH GIVE THE DIMENSIONS OF
C     THE ARRAY 'PAGE' WHICH IS LENGTH*WIDTH BYTES IN SIZE. 'FORMAT' IS
C     AN ARRAY WHICH CONTAINS THE OBJECT-TIME FORMAT OF THE ITEM TO BE
C     WRITTEN ON LINE 'LINE' STARTING IN COLUMN 'COLUMN'. 'SIZE' IS
C     AN INTEGER WHICH GIVES THE NUMBER OF BYTES SPECIFIED IN FORMAT.
C     ************************************************************************
C
      SUBROUTINE CLEAR (WIDTH,LENGTH,PAGE)
      INTEGER LINE,COLUMN,LENGTH,WIDTH,SIZE,IS
      LOGICAL*1 PAGE(131,65),FORMAT(1),BLANK     ,STRING(1)
          DATA BLANK/' '/
C
C     BLANK OUT PAGE
      DO 10 J=1,LENGTH
      DO 10 I=1,WIDTH
   10 PAGE(I,J)=BLANK
      RETURN
C
C     R(EAL) S(CALER) W(RITER)
      ENTRY RSW(LINE,COLUMN,PS,FORMAT,SIZE)
C     CALL CORE (PAGE(COLUMN,LINE),SIZE)
C     WRITE (99,FORMAT) RS
           ENCODE(SIZE,FORMAT,PAGE(COLUMN,LINE))RS
      RETURN
C
C     I(NTEGER) S(CALER) W(RITER)
      ENTRY ISW(LINE,COLUMN,IS,FORMAT,SIZE)
C     CALL CORE (PAGE(COLUMN,LINE),SIZE)
C     WRITE (99,FORMAT) IS
           ENCODE(SIZE,FORMAT,PAGE(COLUMN,LINE)) IS
      RETURN
C
C     S(TRING) A(RRAY) W(RITER)
      ENTRY SAW(LINE,COLUMN,STRING,SIZE)
      DO 20 I=1,SIZE
   20 PAGE(COLUMN+I-1,LINE)=STRING(I)
      RETURN
C
```

```
C       PRINT CONSTRUCTED PAGE
        ENTRY OUTPUT(LEN,WIDTH)
        DO 30 J=1,LEN
   30   PRINT 40,(PAGE(I,J),I=1,WIDTH)
        RETURN
C
40      FORMAT (1H ,131A1)
        END
C       ***********************************************************************
C       THIS SUBPROGRAM IS USED TO EVALUATE ARITHMETIC EXPRESSIONS.
C       THESE EXPRESSIONS ARE COMPOSED OF VARIABLE NAMES STORED IN NAMES,
C       THE ARITHMETIC OPERATORS (UNARY MINUS, /, *, -, AND +), CONSTANTS,
C       AND THE GROUPING SYMBOLS (). EVALUATION IS PERFORMED IN THE ORDER
C       (GIVEN ABOVE) OF THE ARITHMETIC OPERATORS IN A LEFT TO RIGHT
C       FASHION. THIS ORDER CAN BE OVERRIDDEN BY THE USE OF ().
C
C       FIRST A CALL TO PARSE(EXPR,L,NAMES,N,ERR) MUST BE MADE. EXPR
C       CONTAINS THE CHARACTER REPRESENTATION OF THE EXPRESSION AND IS L
C       BYTES (CHARACTERS) IN LENGTH. NAMES IS A REAL*8 ARRAY CONTAINING
C       THE N NAMES WHICH MIGHT APPEAR IN EXPR. ERR IS LOGICAL*1 AND IS
C       SET TO .TRUE. IF THE EXPRESSION IS SYNTACTICALLY INCORRECT OR
C       CONTAINS A VARIABLE WHICH IS NOT IN NAMES.
C       PARSE GENERATES THE REVERSE POLISH FORM OF EXPR IN THE TWO ARRAYS
C       POLISH AND TYPE. POLISH CONTAINS THE NUMERIC VALUES OF ANY
C       CONSTANTS IN EXPR. TYPE CONTAINS THREE TYPES OF INFORMATION.
C       IF TYPE(I)=-5 THEN THE I'TH POLISH FORM ITEM IS UNARY MINUS.
C       IF TYPE(I)=-4 THEN THE I'TH POLISH FORM ITEM IS THE OPERATOR /.
C       IF TYPE(I)=-3 THEN THE I'TH POLISH FORM ITEM IS THE OPERATOR *.
C       IF TYPE(I)=-2 THEN THE I'TH POLISH FORM ITEM IS THE OPERATOR -.
C       IF TYPE(I)=-1 THEN THE I'TH POLISH FORM ITEM IS THE OPERATOR +.
C       IF TYPE(I)= 0 THEN THE I'TH POLISH FORM ITEM IS A CONSTANT FOUND
C          IN POLISH(I).
C       IF TYPE(I)>0 THEN THE I'TH POLISH FORM ITEM IS A VARIABLE WHOSE
C          VALUE IS IN VALUES(TYPE(I)).
C       A TRANSITION MATRIX TECHNIQUE IS USED FOR THE PARSE.
C
C       THE ENTRY EVAL(VALUES) MAY THEN BE USED TO RETURN THE NUMERIC
C       VALUE OF THE EXPRESSION BASED ON THE CURRENT VALUES IN VALUES
C       CORRESPONDING TO VARIABLES IN NAMES. A SIMPLE PUSH-DOWN STACK
C       TECHNIQUE IS USED.
C       ***********************************************************************
C
C        MODIFIED BY JOHN ODELL D.G.M.R. FOR GNAP FEBRUARY 1978
C
C
        FUNCTION EVAL(VALUES)
        LOGICAL*1 EXPR(1),NAME(8),CHAR(4)        ,POP,NUM,BLANK      ,ERR
             INTEGER TM(3,9),TOP,ROW,COLUMN,C,ELEMNT,SWITCH,SYMBOL(186),U
        INTEGER*2 INDEX,TYPE(41)
        REAL NAMES*8(1),VALUES(1),POLISH(41),STACK(41),VARBLE*8
        COMMON /COMM/ U
        EQUIVALENCE                    (NAME(1),VARBLE)
        DATA     TM      /12,32,0,81,2*41,0,2*51,0,2*61,0,2*71,101,3*0,92,93
```

```
     1,21,22,23,113,32,33/
        DATA CHAR/4*"000/,BLANK/' '/
      DATA SYMBOL/31*0,8,2*0,1,4*0,6,7,5,3,0,2,9,4,10*9,7*0,26*1,96*0/
      TOP=0
      DO 340 J=1,I
      INDEX=TYPE(J)
      IF(INDEX) 290,285,280
  280 TOP=TOP+1
      STACK(TOP)=VALUES(INDEX)
      GO TO 340
  285 TOP=TOP+1
      STACK(TOP)=POLISH(J)
      GO TO 340
  290 INDEX=5+INDEX
      IF(INDEX .EQ. 0) GO TO 295
      VT=STACK(TOP)
      TOP=TOP-1
      GO TO (310,300,330,320),INDEX
  295 STACK(TOP)=-STACK(TOP)
      GO TO 340
  300 STACK(TOP)=STACK(TOP)*VT
      GO TO 340
  310 IF(VT .NE. 0.0) GO TO 315
      WRITE(U,500)
      GO TO 360
  315 STACK(TOP)=STACK(TOP)/VT
      GO TO 340
  320 STACK(TOP)=STACK(TOP)+VT
      GO TO 340
  330 STACK(TOP)=STACK(TOP)-VT
  340 CONTINUE
      IF (TOP.NE.1) GO TO 360
      EVAL=STACK(1)
  350 RETURN
  360 WRITE(U,480) (EXPR(J),J=1,L)
      EVAL=0.0
      GO TO 350
C
      ENTRY PARSE(EXPR,L,NAMES,N,ERR)
C     INITIALIZATION
      ERR=.FALSE.
      ROW=1
      TOP=0
      I=0
      C=0
C     SET OPERATOR CODES
      UNARY=-5.
      DIV=-4.
      PROD=-3.
      DIFF=-2.
      PLUS=-1.
      PAREN=0.
C     CONSTRUCT POLISH FORM VIA TRANSITION MATRIX TM
```

```
    30 C=C+1
C       ARE THERE MORE CHARACTERS TO PROCESS IN EXPR ?
        IF (C.LE.L) GO TO 50
        IF(ROW.EQ.1) GO TO 460
        IF (POP) GO TO 250
        IF (NUM) GO TO 40
        SWITCH=1
C       GO FIND VALUE AND INSERT IN POLISH STRING
        GO TO 370
    40 SWITCH=2
C       GO EVALUATE CONSTANT AND INSERT IN POLISH STRING
        GO TO 420
C       GET THE CHARACTER, DETERMINE TASK (JOB) AND NEXT-STATE (ROW),
    50 CHAR(4)=EXPR(C)
                NEXT=CHAR(4)
        COLUMN=SYMBOL(NEXT    )
        IF (COLUMN.EQ.0) GO TO 360
        ELEMNT=TM(ROW,COLUMN)
        JOB=ELEMNT/10
        ROW=MOD(ELEMNT,10)
        GO TO (60,30,70,130,140,150,160,170,180,230,240), JOB
        GO TO 360
C       START AN ITEM
60          NAME(1)=CHAR(4)
        NCHAR=1
        POP=.FALSE.
        NUM=.FALSE.
        GO TO 30
C       ADD CURRENT CHARACTER TO PARTIALLY BUILT ITEM
    70 NCHAR=NCHAR+1
        NAME(NCHAR)=CHAR(4)
        GO TO 30
    80 IF (POP) GO TO 100
        POP=.TRUE.
        IF (NUM) GO TO 90
        SWITCH=3
C       GO FIND VALUE AND INSERT IN POLISH STRING
        GO TO 370
    90 SWITCH=4
C       GO EVALUATE CONSTANT AND INSERT IN POLISH STRING
        GO TO 420
   100 IF (TOP.EQ.0) GO TO 120
        IF (CODE.LT.STACK(TOP)) GO TO 120
        SWITCH=5
        VALUE=STACK(TOP)
        INDEX=VALUE
        GO TO 470
   110 TOP=TOP-1
        GO TO 100
   120 TOP=TOP+1
        STACK(TOP)=CODE
        GO TO 30
C       SET CODE FOR OPERATOR AND GO PROCESS ITEM
```

```
  130 CODE=DIFF
      GO TO 80
  140 CODE=PLUS
      GO TO 80
  150 CODE=DIV
      GO TO 80
  160 CODE=PROD
      GO TO 80
  170 CODE=UNARY
      GO TO 120
C     RIGHT PAREN SENSED, INSERT ITEM IN POLISH AND POP STACK
  180 IF (POP) GO TO 200
      POP=.TRUE.
      IF (NUM) GO TO 190
      SWITCH=6
C     GO FIND VALUE AND INSERT IN POLISH STRING
      GO TO 370
  190 SWITCH=7
C     GO EVALUATE CONSTANT AND INSERT IN POLISH STRING
      GO TO 420
  200 IF (TOP.EQ.0) GO TO 460
      IF (PAREN.EQ.STACK(TOP)) GO TO 220
      SWITCH=8
      VALUE=STACK(TOP)
      INDEX=VALUE
      GO TO 470
  210 TOP=TOP-1
      GO TO 200
  220 TOP=TOP-1
      GO TO 30
C     LEFT PAREN SENSED, INSERT IN STACK
  230 TOP=TOP+1
      STACK(TOP)=PAREN
      GO TO 30
  240 NAME(1)=CHAR(4)
      NCHAR=1
      NUM=.TRUE.
      POP=.FALSE.
      GO TO 30
  250 IF (TOP.EQ.0) GO TO 350
      SWITCH=9
      VALUE=STACK(TOP)
      IF(VALUE.EQ.PAREN) GO TO 460
      INDEX=VALUE
      GO TO 470
  260 TOP=TOP-1
      GO TO 250
C     GET INDEX ASSOCIATED WITH NAME
  370 NCHAR=NCHAR+1
      IF (NCHAR.GT.8) GO TO 390
C     PAD NAME WITH BLANKS
      DO 380 J=NCHAR,8
  380 NAME(J)=BLANK
```

```
  390 DO 400 J=1,N
      IF (VARBLE.EQ.NAMES(J)) GO TO 410
  400 CONTINUE
      WRITE(U,490) VARBLE
      GO TO 460
  410 INDEX=J
      GO TO 470
C     EVALUATE A CONSTANT
  420 VALUE=CONV(NAME,1,NCHAR,0,ERR)
      INDEX=0
      IF (ERR) GO TO 460
  470 I=I+1
      POLISH(I)=VALUE
      TYPE(I)=INDEX
      GO TO (250,250,100,100,110,200,200,210,260), SWITCH
  460 ERR=.TRUE.
      WRITE(U,480) (EXPR(J),J=1,L)
      PARSE=0.0
      RETURN
C
  480 FORMAT ('0ERROR IN EXPRESSION ',80A1)
  490 FORMAT ('0UNDEFINED NAME ',A8)
  500 FORMAT('0DIVIDE BY ZERO ATTEMPTED, HENCE!')
      END
C     ************************************************************************
C     MOVE(A,B,N) MOVES 'N' BYTES (IE CHARACTERS) FROM THE ARRAY 'A' TO
C     THE ARRAY 'B'.
C     ************************************************************************
C
      SUBROUTINE MOVE (A,B,N)
      LOGICAL*1 A(1),B(1)
      DO 10 I=1,N
   10 B(I)=A(I)
      RETURN
      END
C     ************************************************************************
C     INDEX(STRING,CHAR,N) RETURNS THE POSITION OF THE CHARACTER CHAR IN
C     THE STRING STRING. IF CHAR IS NOT PRESENT ZERO IS RETURNED
C     ************************************************************************
C
      FUNCTION INDEX (STRING,CHAR,N)
      LOGICAL*1 STRING(N),CHAR,TEMP(4)        ,TEM(4)
      EQUIVALENCE (TEMP(1),NS), (TEM(1),NC)
         DATA TEMP,TEM/8*Z00/
      TEM(4)=CHAR
      DO 10 I=1,N
      TEMP(4)=STRING(I)
      IF (NS.EQ.NC) GO TO 20
   10 CONTINUE
      I=0
   20 INDEX=I
      RETURN
      END
```

```
C     ************************************************************************
C     NOTEQ(A,B,N) RETURNS A VALUE (TRUE OR FALSE) INDICATING WHETHER OR
C     NOT THE TWO STRINGS A AND B ARE EQUAL IN THE FIRST N CHARACTERS
C     ************************************************************************
C
      LOGICAL FUNCTION NOTEQ(A,B,N)
      LOGICAL*1 A(1),B(1),TA(4)        ,TB(4)
      EQUIVALENCE (TA(1),NA), (TB(1),NB)
            DATA TA,TB/8*Z00/
      DO 10 I=1,N
      TA(4)=A(I)
      TB(4)=B(I)
      IF (NA.NE.NB) GO TO 30
   10 CONTINUE
      NOTEQ=.FALSE.
   20 RETURN
   30 NOTEQ=.TRUE.
      GO TO 20
      END
```