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

<table>
<thead>
<tr>
<th>CONTENTS</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>1</td>
</tr>
<tr>
<td>INTRODUCTION AND USE</td>
<td>1</td>
</tr>
<tr>
<td>PROGRAM DESCRIPTION</td>
<td>4</td>
</tr>
<tr>
<td>INPUT</td>
<td>6</td>
</tr>
<tr>
<td>Title card</td>
<td>6</td>
</tr>
<tr>
<td>Analysis card</td>
<td>7</td>
</tr>
<tr>
<td>MODES OF OPERATION</td>
<td>8</td>
</tr>
<tr>
<td>INTERACTIVE OPERATION</td>
<td>8</td>
</tr>
<tr>
<td>COMMANDS</td>
<td>11</td>
</tr>
<tr>
<td>Expressions</td>
<td>11</td>
</tr>
<tr>
<td>Modify-format command</td>
<td>12</td>
</tr>
<tr>
<td>RASS and STATPAC modifications</td>
<td>13</td>
</tr>
<tr>
<td>Oxide command</td>
<td>14</td>
</tr>
<tr>
<td>Nonorms command</td>
<td>15</td>
</tr>
<tr>
<td>Define command</td>
<td>16</td>
</tr>
<tr>
<td>Plot command</td>
<td>16</td>
</tr>
<tr>
<td>Ternary command</td>
<td>17</td>
</tr>
<tr>
<td>Convert-values command</td>
<td>18</td>
</tr>
<tr>
<td>Print command</td>
<td>18</td>
</tr>
<tr>
<td>Recalculate-norms command</td>
<td>19</td>
</tr>
<tr>
<td>Graphs command</td>
<td>19</td>
</tr>
<tr>
<td>Scale command</td>
<td>19</td>
</tr>
<tr>
<td>Summary command</td>
<td>21</td>
</tr>
<tr>
<td>Clear command</td>
<td>22</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>25</td>
</tr>
</tbody>
</table>
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
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 '0' (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-argument 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-argument 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-argument 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

<table>
<thead>
<tr>
<th>Columns</th>
<th>Field width</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>3</td>
<td>NRM or STO</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>plotting symbol</td>
</tr>
<tr>
<td>5-10</td>
<td>6</td>
<td>sample identifier/lab. number</td>
</tr>
<tr>
<td>11-14</td>
<td>4</td>
<td>weight percent SiO₂</td>
</tr>
<tr>
<td>15-18</td>
<td>4</td>
<td>weight percent Al₂O₃</td>
</tr>
<tr>
<td>19-22</td>
<td>4</td>
<td>weight percent Fe₂O₃</td>
</tr>
<tr>
<td>23-26</td>
<td>4</td>
<td>weight percent FeO</td>
</tr>
<tr>
<td>27-30</td>
<td>4</td>
<td>weight percent MgO</td>
</tr>
<tr>
<td>31-34</td>
<td>4</td>
<td>weight percent CaO</td>
</tr>
<tr>
<td>35-38</td>
<td>4</td>
<td>weight percent Na₂O</td>
</tr>
<tr>
<td>39-42</td>
<td>4</td>
<td>weight percent K₂O</td>
</tr>
<tr>
<td>43-46</td>
<td>4</td>
<td>weight percent H₂O</td>
</tr>
<tr>
<td>47-49</td>
<td>3</td>
<td>weight percent TiO₂</td>
</tr>
<tr>
<td>50-52</td>
<td>3</td>
<td>weight percent P₂O₅</td>
</tr>
<tr>
<td>53-56</td>
<td>4</td>
<td>weight percent MnO</td>
</tr>
<tr>
<td>57-60</td>
<td>3</td>
<td>weight percent ZrO₂</td>
</tr>
<tr>
<td>61-63</td>
<td>4</td>
<td>weight percent CO₂</td>
</tr>
<tr>
<td>64-66</td>
<td>3</td>
<td>weight percent SO₃</td>
</tr>
<tr>
<td>67-69</td>
<td>3</td>
<td>weight percent Cl</td>
</tr>
<tr>
<td>70-72</td>
<td>3</td>
<td>weight percent F</td>
</tr>
<tr>
<td>73-75</td>
<td>3</td>
<td>weight percent S</td>
</tr>
<tr>
<td>76-78</td>
<td>3</td>
<td>weight percent Cr₂O₃</td>
</tr>
<tr>
<td>79-80</td>
<td>2</td>
<td>weight percent NiO</td>
</tr>
</tbody>
</table>
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. SI02), division (e.g. SI02/FEO), multiplication (e.g. SI02*FEO), subtraction (e.g. SI02-FEO), and addition (e.g. SI02+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= (1$X,...)` 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 (10X, 5F7.2), NCS = 3;
```

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

```
MODIFY FORMAT (10X, 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 argument '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:

```
(10 (IX, G6.0), 10X)
```

**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 SI02, AL2O3, FE2O3, FEO, MGO, CAO, NA2O,
    K2O, H2O, TI02, 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*P205+CO2); redefines CaO as the original CaO minus the sum of CO₂ and 3P₂O₅,
and CI=AN+2.157003*DIEN+(FO+0.700837*HYEN)+MT+CM;
defines the new variable CI to be equal to An+2.157003Enₐi+(Fo+0.700837Enₜy)+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 SI02, (FEO+FE203)/(FEO+FE203+MGO);
instructs the computer to plot SiO₂ versus

\[
\frac{FeO+Fe₂O₃}{FeO+Fe₂O₃+MgO}
\]

with the abscissa (i.e. SiO₂) 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 (FE0+FE2O3)/(FE0+FE2O3+MGO), FE0+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$_2$O$_3$, 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 $8 \times 10^n$. 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<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 $\text{Al}_2\text{O}_3/\text{SiO}_2$ and $\text{FeO}/\text{Fe}_2\text{O}_3$

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.\textsuperscript{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).

\textsuperscript{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

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

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.

<table>
<thead>
<tr>
<th>Oxides</th>
<th>Minerals</th>
<th>Barth's cations</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIO2 silica</td>
<td>Q quartz</td>
<td>CM chromite</td>
</tr>
<tr>
<td>AL2O3 alumina</td>
<td>C corundum</td>
<td>HM hematite</td>
</tr>
<tr>
<td>FE2O3 ferric</td>
<td>Z zircon</td>
<td>IL ilmenite</td>
</tr>
<tr>
<td>FEO ferrous</td>
<td>OR orthoclase</td>
<td>TN titanite</td>
</tr>
<tr>
<td>MGO magnesium</td>
<td>AB albite</td>
<td>PF perofskite</td>
</tr>
<tr>
<td>CAO calcium</td>
<td>AN anorthite</td>
<td>RU rutile</td>
</tr>
<tr>
<td>NA2O sodium</td>
<td>LC leucite</td>
<td>AP apatite</td>
</tr>
<tr>
<td>K2O potassium</td>
<td>NE nepheline</td>
<td>FR fluorite</td>
</tr>
<tr>
<td>H2O water</td>
<td>KP kaliophilite</td>
<td>PR pyrite</td>
</tr>
<tr>
<td>TIO2 titanium</td>
<td>HL halite</td>
<td>CC calcite</td>
</tr>
<tr>
<td>P2O5 phosphorus</td>
<td>TH thenardite</td>
<td>MG MGO/FM^1/ P phosphorus</td>
</tr>
<tr>
<td>MNO manganese</td>
<td>NC sodium carbonate</td>
<td>DI diopside</td>
</tr>
<tr>
<td>ZRO2 zirconium</td>
<td>AC acmite</td>
<td>DIWO WO content of DI</td>
</tr>
<tr>
<td>CO2 carbon dioxide</td>
<td>NS sodium metasilicate</td>
<td>DIEN EN content of DI</td>
</tr>
<tr>
<td>S03 sulphate</td>
<td>KS potassium metasilicate</td>
<td>DIEN EN content of DI</td>
</tr>
<tr>
<td>CL chlorite</td>
<td>WO wollastonite</td>
<td>DIFS FG content of DI</td>
</tr>
<tr>
<td>F fluoride</td>
<td>EN enstatite</td>
<td>HY hypersthene</td>
</tr>
<tr>
<td>S sulphur</td>
<td>FS ferrosilite</td>
<td>HYEN EN content of HY</td>
</tr>
<tr>
<td>CR2O3 chromium</td>
<td>FO forsterite</td>
<td>HYEN EN content of HY</td>
</tr>
<tr>
<td>NIO nickel</td>
<td>FA fayalite</td>
<td>HYFS HY content of HY</td>
</tr>
<tr>
<td>BAO barium</td>
<td>CS calcium silicates</td>
<td>OL olivine</td>
</tr>
<tr>
<td></td>
<td>MT magnetite</td>
<td></td>
</tr>
</tbody>
</table>

^1/ FM = 2Fe₂O₃ + FeO + MgO + MnO + NiO

WOL WO minus DIWO
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 $\text{Al}_2\text{O}_3/\text{SiO}_2$ 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:

<table>
<thead>
<tr>
<th>n</th>
<th>Cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Insufficient CaO to form AP</td>
</tr>
<tr>
<td>3</td>
<td>Insufficient Na₂O to form HL</td>
</tr>
<tr>
<td>5</td>
<td>Insufficient FeO to form PR</td>
</tr>
<tr>
<td>6</td>
<td>Cr₂O₃ in excess of FeO</td>
</tr>
<tr>
<td>9</td>
<td>Insufficient CaO to form FR</td>
</tr>
<tr>
<td>11</td>
<td>Insufficient MgO to form EN after CO₂ subtracted</td>
</tr>
<tr>
<td>12</td>
<td>Insufficient SiO₂ to form Z</td>
</tr>
<tr>
<td>31</td>
<td>Improper amount of SiO₂ to form KP or LC</td>
</tr>
</tbody>
</table>

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**


APPENDICES TO ACCOMPANY USGS PROJECT REPORT 254

GNAP

(GRAPHIC NORMATIVE ANALYSIS)

Page

APPENDIX A - Interactive operation.......................... 37
APPENDIX B - Batch operation................................. 38
APPENDIX C - GNAP source code.............................. 56

35

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 SI02,TI02,AL203,FE203,FE0,MNO,MGO,CA0,NA20,K20,H20,P205;
**GNAP>MODIFY FORMAT (10X,12F4.2);
**GNAP>NONORMS;
**GNAP>NRM BATCH [14,14]DAWA.DAT
**GNAP>NRMUISA16 '713000351493005601760003002600000444050300570005
**GNAP>NRMVISA39+ PROMPT
SAMPLE ID ISA391
S102 .......... 7169
T102 .......... 9
AL203 .......... 1550
FE203 .......... 53
FE0 .......... 103
MNO .......... 2
MGO .......... 27
CA0 .......... 51
NA20 .......... 416
K20 .......... 463
H20 .......... 158
P205 .......... 2
**GNAP>SUMMARY (OXIDES,MINERALS,D,I.,RATIOS);
**GNAP>H20=0;FE0=FE0+FE203*0.9;FE203=0;FEMAG=FE0/(FE0+MGO);

AD DAWADIMI DISTRICT GRANITES (AA).

H20 DEFINED AS 0
FE0 DEFINED AS FE0+FE203*0.9
FE203 DEFINED AS 0
FEMAG DEFINED AS FE0/(FE0+MGO)
**GNAP>GRAPHS=CALCOMP;
**GNAP>SCALE=1.0;
**GNAP>TERNARY FE0+MNO,NA20+K20,MGO;
**GNAP>CLEAR STORAGE;
**GNAP>~Z

GNAP -- STOP

MCR>FLX X Y:/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 DAMADIMI DISTRICT
NRMAISA39'6380160001840376028001790382025003500680140012
NRMCIA39'716915500053010302700510416046301580090020002
OXIDES $SI02$,$TI02$,$AL203$,$FE0$,$MNO$,$MGO$,$CA0$,$NA20$,$K20$,$H20$,$P205$;
MODIFY FORMAT (10X,12F4.2);

MODIFY FORMAT (10X,12F4.2);

NRMAISA-70169500038126602480346006012402750408040400075000
NRMAAS-7046432003013820319045500801990327045303010580009
NRMAAS-70663830067135003460530009016205150396026101080014
NRMAAS-2066900003013060122015600406062240687049000700004
NRMAAS-372731800261280006201950020230126051004100200001
NRMAAS-3957520004211100280110000906101260512037500600007
NRMAAS-726A7640121450043007300301300830441045100320002
NRMAAS-729754500671039011101640003000003800950448046606700002
NRMAAS-8227030005134000056000010013031300450078000270001
NRMAAS-82369200017100102940392000702142070415038801010009
PRINT FEO+FE203/(MGO+FEO+FE203);

SUMMARY (OXIDES, MINERALS, BARTHS, RATIOS);
TERNARY (AB, OR);
STO DUMMY1850000101600040006000100250130000001300
STO DUMMY25000000110000001000100010000000000000

PLOT SI02, CA0+NA20;
GRAPH=CALCOMP;
SCALE=1.0;
PLOT SI02, K20+NA20;
H20=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+NA20+K20+MGO;
CLEAR STORAGE;
TITLE PLOTTING OF OTHER DATA -- AL JI'LANI LAYERED INTRUSION.
OXIDES SI02, 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-164 52 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=SI02; STRONT=AL203;
GRAPH=CALCOMP;
PLOT(R) ANORTH, STRONT;
/* EOF */
Granitic Rocks from the Ad Dawadi District

Original wt. pct. oxides:

| Constituents | SiO₂ | Al₂O₃ | Fe₂O₃ | FeO | MgO | CaO | Na₂O | K₂O | H₂O | TiO₂ | P₂O₅ | MnO | ZrO₂ | CO₂ | SO₃ | Cl | F | S | Cr₂O₃ | NiO | MaO | Granitic Rocks from the Ad Dawadi District |
|--------------|------|-------|-------|-----|-----|-----|------|-----|-----|------|------|-----|-----|-----|-----|---|---|---|---|-----|-----|-----|-------|
| Percentages  | 63.33| 15.88 | 1.83  | 3.78| 2.78| 1.78| 3.79 | 2.48| 3.47| 0.67 | 0.14 | 0.26| 0.01 | 0.01 | 0.00| 0.00| 0.00| 0.00| 0.00| 0.00| 0.00| 0.00|
| MOL. AMTS.   | 1.0339| 0.2158| 0.0219| 0.0619| 0.0689| 0.0317| 0.0512| 0.0265| 0.1928| 0.0264| 0.0018| 0.0102|

Plotting symbol is A

CIPW norm for sample no. 18439:

<table>
<thead>
<tr>
<th>Minerals</th>
<th>Q</th>
<th>C</th>
<th>Z</th>
<th>OR</th>
<th>AB</th>
<th>AN</th>
<th>LC</th>
<th>NE</th>
<th>KP</th>
<th>HL</th>
<th>TH</th>
<th>NC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentages</td>
<td>22.19</td>
<td>4.969</td>
<td>0.099</td>
<td>14.623</td>
<td>32.093</td>
<td>7.966</td>
<td>0.099</td>
<td>0.099</td>
<td>0.099</td>
<td>0.099</td>
<td>2.640</td>
<td>0.099</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Minerals</th>
<th>AC</th>
<th>NS</th>
<th>KS</th>
<th>WO</th>
<th>EN</th>
<th>FS</th>
<th>FO</th>
<th>FA</th>
<th>CS</th>
<th>MT</th>
<th>CM</th>
<th>HP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentages</td>
<td>0.0004</td>
<td>0.0007</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Minerals</th>
<th>IL</th>
<th>TN</th>
<th>PF</th>
<th>RU</th>
<th>AP</th>
<th>FR</th>
<th>PR</th>
<th>CC</th>
<th>MG</th>
<th>TOTAL</th>
<th>SALIC</th>
<th>FEMIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentages</td>
<td>1.282</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>96.535</td>
<td>80.983</td>
<td>15.532</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Minerals</th>
<th>DI</th>
<th>DI=WN</th>
<th>DI=EN</th>
<th>DI=FS</th>
<th>HY</th>
<th>HY=EN</th>
<th>HY=FS</th>
<th>OL</th>
<th>OL=FO</th>
<th>OL=FA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentages</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.1027</td>
<td>0.0689</td>
<td>0.0337</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

Thornton + Tuttle differentiation index = 68.857

Barth's cations:

<table>
<thead>
<tr>
<th></th>
<th>Si</th>
<th>Al</th>
<th>Fe³⁺</th>
<th>Fe²⁺</th>
<th>Mg</th>
<th>Ca</th>
<th>Na</th>
<th>K</th>
<th>H²⁺</th>
<th>Tl</th>
<th>P</th>
<th>Mn</th>
</tr>
</thead>
<tbody>
<tr>
<td>SI</td>
<td>60.99</td>
<td>15.63</td>
<td>3.12</td>
<td>3.99</td>
<td>1.83</td>
<td>7.08</td>
<td>3.05</td>
<td>2.32</td>
<td>0.49</td>
<td>0.11</td>
<td>0.10</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Zr</th>
<th>C</th>
<th>Si</th>
<th>Cl</th>
<th>F</th>
<th>S</th>
<th>Fe</th>
<th>Cr</th>
<th>Ni</th>
<th>Ba</th>
</tr>
</thead>
<tbody>
<tr>
<td>SI</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Nigglí values:

<table>
<thead>
<tr>
<th></th>
<th>Al⁺⁺</th>
<th>Fe⁺⁺</th>
<th>Ca²⁺</th>
<th>Na⁺⁺</th>
<th>K⁺⁺</th>
<th>H⁺⁺</th>
<th>Tl⁺⁺</th>
<th>P</th>
<th>Mn⁺⁺</th>
</tr>
</thead>
<tbody>
<tr>
<td>SI</td>
<td>37.05</td>
<td>34.60</td>
<td>7.54</td>
<td>20.82</td>
<td>250.70</td>
<td>2.01</td>
<td>0.23</td>
<td>45.87</td>
<td>0.30</td>
</tr>
</tbody>
</table>

Listing 3: Output document
### Granitic Rocks from the Ad Dawadimi District

**Original wt. pct. oxides**

<table>
<thead>
<tr>
<th>Oxides</th>
<th>SiO2</th>
<th>Al2O3</th>
<th>Fe2O3</th>
<th>FeO</th>
<th>MgO</th>
<th>CaO</th>
<th>Na2O</th>
<th>K2O</th>
<th>H2O</th>
<th>TiO2</th>
<th>P2O5</th>
<th>MnO</th>
<th>ZrO2</th>
<th>CO2</th>
<th>SO3</th>
<th>Cl</th>
<th>F</th>
<th>S</th>
<th>Cr2O3</th>
<th>NiO</th>
<th>BaO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values</td>
<td>71.09</td>
<td>15.58</td>
<td>0.53</td>
<td>1.03</td>
<td>0.27</td>
<td>0.51</td>
<td>4.16</td>
<td>4.63</td>
<td>1.58</td>
<td>0.09</td>
<td>0.92</td>
<td>0.02</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Percentages</td>
<td>27.716</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.0171</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Thornton + Tuttle Differentiation Index = 90.258**

**Barth's Cations**

<table>
<thead>
<tr>
<th>Cations</th>
<th>Si</th>
<th>Al</th>
<th>Fe+3</th>
<th>Fe+2</th>
<th>Mg</th>
<th>Ca</th>
<th>Na</th>
<th>K</th>
<th>H</th>
<th>Ti</th>
<th>P</th>
<th>Mn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values</td>
<td>67.48</td>
<td>17.19</td>
<td>0.38</td>
<td>0.38</td>
<td>0.51</td>
<td>7.59</td>
<td>5.56</td>
<td>9.92</td>
<td>0.06</td>
<td>0.02</td>
<td>0.02</td>
<td></td>
</tr>
</tbody>
</table>

**Nigglı Values**

<table>
<thead>
<tr>
<th>Values</th>
<th>Al+</th>
<th>FM+</th>
<th>C+</th>
<th>AlK+</th>
<th>Si</th>
<th>RI</th>
<th>P</th>
<th>H</th>
<th>K</th>
<th>Mg</th>
<th>St</th>
<th>Qz</th>
</tr>
</thead>
<tbody>
<tr>
<td>49.79</td>
<td>9.16</td>
<td>2.98</td>
<td>38.08</td>
<td>300.77</td>
<td>0.37</td>
<td>0.05</td>
<td>28.72</td>
<td>0.42</td>
<td>0.24</td>
<td>252.32</td>
<td>138.45</td>
<td></td>
</tr>
</tbody>
</table>
### GRANITIC ROCKS FROM THE AD DAWANIM DISTRICT

**EVALUATION OF FeO+Fe₂O₃/(MgO+FeO+Fe₂O₃)**

<table>
<thead>
<tr>
<th>ID</th>
<th>VALUE</th>
<th>ID</th>
<th>VALUE</th>
<th>ID</th>
<th>VALUE</th>
<th>ID</th>
<th>VALUE</th>
<th>ID</th>
<th>VALUE</th>
<th>ID</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYMBOL</td>
<td>GRANITIC ROCKS FROM THE ADDAWADIMI DISTRICT</td>
<td>GRANITIC ROCKS FROM THE ADDAWADIMI DISTRICT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>---------------------------------</td>
<td>---------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS</td>
<td>5.74</td>
<td>5.74</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SI</td>
<td>0.61</td>
<td>0.61</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>6.92</td>
<td>6.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fe</td>
<td>1.83</td>
<td>1.83</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O</td>
<td>3.99</td>
<td>3.99</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fe2O3</td>
<td>1.97</td>
<td>1.97</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TiO2</td>
<td>1.1</td>
<td>1.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MgO</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MnO</td>
<td>0.05</td>
<td>0.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P2O5</td>
<td>0.01</td>
<td>0.01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K2O</td>
<td>0.03</td>
<td>0.03</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Na2O</td>
<td>0.02</td>
<td>0.02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CaO</td>
<td>0.01</td>
<td>0.01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ti</td>
<td>0.01</td>
<td>0.01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FeO</td>
<td>0.01</td>
<td>0.01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mg</td>
<td>0.01</td>
<td>0.01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Al2O3</td>
<td>0.01</td>
<td>0.01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SiO2</td>
<td>2.04</td>
<td>2.04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H2O</td>
<td>0.01</td>
<td>0.01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>100.00</td>
<td>100.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Analysis of the data shows a high concentration of Fe2O3 and TiO2, indicating the presence of iron and titanium minerals in the granitic rocks. The insoluble residue (8.98%) suggests the presence of refractory minerals such as feldspar and quartz.
GRANITIC ROCKS FROM THE AD DAWADIM DISTRICT

TERNARY RATIOS FOR Q, A, OR

<table>
<thead>
<tr>
<th>SAMPLE</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>SYMBOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISAJ91</td>
<td>32.19</td>
<td>46.54</td>
<td>21.27</td>
<td>A</td>
</tr>
<tr>
<td>ISAJ96</td>
<td>38.71</td>
<td>38.99</td>
<td>30.30</td>
<td>C</td>
</tr>
<tr>
<td>AS-784</td>
<td>29.22</td>
<td>42.44</td>
<td>29.34</td>
<td>1</td>
</tr>
<tr>
<td>AS-786</td>
<td>22.00</td>
<td>53.25</td>
<td>24.72</td>
<td>2</td>
</tr>
<tr>
<td>AS-785</td>
<td>25.00</td>
<td>59.12</td>
<td>23.97</td>
<td>3</td>
</tr>
<tr>
<td>AS-782</td>
<td>17.01</td>
<td>47.63</td>
<td>34.57</td>
<td>4</td>
</tr>
<tr>
<td>AS-787</td>
<td>26.95</td>
<td>46.48</td>
<td>26.57</td>
<td>5</td>
</tr>
<tr>
<td>AS-795</td>
<td>35.39</td>
<td>40.98</td>
<td>24.52</td>
<td>6</td>
</tr>
<tr>
<td>AS7284A</td>
<td>35.43</td>
<td>36.10</td>
<td>28.47</td>
<td>7</td>
</tr>
<tr>
<td>AS-729</td>
<td>38.96</td>
<td>39.95</td>
<td>30.99</td>
<td>8</td>
</tr>
<tr>
<td>AS-822</td>
<td>20.41</td>
<td>34.22</td>
<td>45.07</td>
<td>9</td>
</tr>
<tr>
<td>AS-823</td>
<td>34.11</td>
<td>59.55</td>
<td>34.34</td>
<td>P</td>
</tr>
</tbody>
</table>
GRANITIC ROCKS FROM THE ADÁDÁNÍMI DISTRICT

<table>
<thead>
<tr>
<th></th>
<th>Defined as</th>
</tr>
</thead>
<tbody>
<tr>
<td>H2O</td>
<td></td>
</tr>
<tr>
<td>FeO</td>
<td>FeO * H2O + Fe2O3</td>
</tr>
<tr>
<td>Fe2O3</td>
<td>Defined as H</td>
</tr>
<tr>
<td>FeMg</td>
<td>Defined as FeO / (FeO + MgO)</td>
</tr>
</tbody>
</table>
# Granitic Rocks from the Ad Dawadimi District

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ADJUSTED OXIDES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIO2</td>
<td>65.73</td>
<td>72.86</td>
<td>69.17</td>
<td>65.88</td>
<td>63.84</td>
<td>69.16</td>
<td>73.54</td>
<td>75.98</td>
<td>77.18</td>
<td>75.78</td>
<td>71.14</td>
<td>69.42</td>
<td></td>
</tr>
<tr>
<td>TIO2</td>
<td>4.70</td>
<td>8.09</td>
<td>6.38</td>
<td>7.30</td>
<td>8.07</td>
<td>8.30</td>
<td>8.26</td>
<td>8.42</td>
<td>8.12</td>
<td>8.67</td>
<td>8.95</td>
<td>8.17</td>
<td></td>
</tr>
<tr>
<td>P2O5</td>
<td>5.58</td>
<td>1.53</td>
<td>5.67</td>
<td>7.51</td>
<td>8.42</td>
<td>2.66</td>
<td>2.52</td>
<td>1.37</td>
<td>1.13</td>
<td>2.65</td>
<td>1.31</td>
<td>6.65</td>
<td></td>
</tr>
<tr>
<td>MMO</td>
<td>0.12</td>
<td>0.02</td>
<td>0.06</td>
<td>0.08</td>
<td>0.09</td>
<td>0.04</td>
<td>0.02</td>
<td>0.09</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
<td>0.87</td>
</tr>
<tr>
<td>MO</td>
<td>2.88</td>
<td>0.27</td>
<td>0.23</td>
<td>0.81</td>
<td>1.62</td>
<td>0.66</td>
<td>0.23</td>
<td>0.62</td>
<td>0.13</td>
<td>0.36</td>
<td>0.33</td>
<td>2.16</td>
<td></td>
</tr>
<tr>
<td>CAO</td>
<td>1.84</td>
<td>0.52</td>
<td>2.74</td>
<td>3.31</td>
<td>5.15</td>
<td>2.25</td>
<td>1.27</td>
<td>0.84</td>
<td>0.95</td>
<td>1.34</td>
<td>2.73</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NAO</td>
<td>3.94</td>
<td>4.23</td>
<td>4.00</td>
<td>4.58</td>
<td>3.90</td>
<td>6.89</td>
<td>5.13</td>
<td>5.17</td>
<td>4.47</td>
<td>4.47</td>
<td>4.47</td>
<td>4.47</td>
<td>4.47</td>
</tr>
<tr>
<td>KO</td>
<td>7.58</td>
<td>4.71</td>
<td>4.02</td>
<td>3.95</td>
<td>2.61</td>
<td>4.91</td>
<td>4.16</td>
<td>3.79</td>
<td>4.57</td>
<td>4.57</td>
<td>4.57</td>
<td>4.57</td>
<td>4.57</td>
</tr>
<tr>
<td>P2O5</td>
<td>8.14</td>
<td>8.52</td>
<td>8.09</td>
<td>8.80</td>
<td>8.14</td>
<td>8.04</td>
<td>8.01</td>
<td>8.07</td>
<td>8.02</td>
<td>8.02</td>
<td>8.02</td>
<td>8.02</td>
<td>8.02</td>
</tr>
<tr>
<td>C</td>
<td>4.316</td>
<td>2.811</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AB</td>
<td>33.301</td>
<td>35.774</td>
<td>34.362</td>
<td>38.785</td>
<td>35.514</td>
<td>39.099</td>
<td>43.062</td>
<td>35.993</td>
<td>34.212</td>
<td>27.590</td>
<td>25.012</td>
<td>36.196</td>
<td></td>
</tr>
<tr>
<td>AN</td>
<td>8.286</td>
<td>2.439</td>
<td>4.278</td>
<td>8.586</td>
<td>11.354</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WO</td>
<td>5.953</td>
<td>3.822</td>
<td>5.547</td>
<td>4.041</td>
<td>2.555</td>
<td>2.444</td>
<td>1.986</td>
<td>1.929</td>
<td>2.744</td>
<td>5.463</td>
<td>2.944</td>
<td>1.234</td>
<td></td>
</tr>
<tr>
<td>EN</td>
<td>7.184</td>
<td>5.683</td>
<td>3.974</td>
<td>5.915</td>
<td>4.055</td>
<td>1.448</td>
<td>0.576</td>
<td>1.535</td>
<td>0.928</td>
<td>0.956</td>
<td>0.925</td>
<td>5.365</td>
<td></td>
</tr>
<tr>
<td>IL</td>
<td>1.331</td>
<td>0.749</td>
<td>0.718</td>
<td>0.577</td>
<td>1.273</td>
<td>0.971</td>
<td>0.496</td>
<td>0.866</td>
<td>0.231</td>
<td>1.777</td>
<td>0.365</td>
<td>4.326</td>
<td></td>
</tr>
<tr>
<td>AP</td>
<td>0.342</td>
<td>0.284</td>
<td>0.322</td>
<td>0.216</td>
<td>0.824</td>
<td>0.468</td>
<td>0.010</td>
<td>0.090</td>
<td>0.055</td>
<td>0.215</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>100.000</td>
<td>100.000</td>
<td>100.000</td>
<td>100.000</td>
<td>100.000</td>
<td>100.000</td>
<td>100.000</td>
<td>100.000</td>
<td>100.000</td>
<td>100.000</td>
<td>100.000</td>
<td>100.000</td>
<td></td>
</tr>
<tr>
<td>SALIC</td>
<td>81.344</td>
<td>90.397</td>
<td>82.464</td>
<td>77.741</td>
<td>74.308</td>
<td>84.423</td>
<td>91.999</td>
<td>91.404</td>
<td>94.494</td>
<td>89.569</td>
<td>91.056</td>
<td>75.417</td>
<td></td>
</tr>
</tbody>
</table>

**DI**

- 7,597 6,181 11,443 9,297 5,436 4,885 3,517 3,971 5,616 10,995
- 3,693 3,822 5,547 4,541 2,595 2,444 1,866 1,926 2,746 5,463
- 8,928 8,859 1,263 1,284 1,300 1,281 1,280 1,280 1,280 1,280
- 2,078 2,302 4,614 3,475 2,501 1,574 1,566 1,566 1,566 1,566
- 19,363 3,383 9,060 15,292 12,653 1,362 1,699 1,699 1,699 1,699
- 7,184 0,683 2,148 4,156 2,752 8,167 9,236 9,468 9,662 9,662
- 9,319 2,699 6,911 11,136 9,908 6,959 1,733 6,063 8,367 2,166

**FEMAG**

- 0.659 0.848 0.821 0.789 0.839 0.839 0.916 0.589 0.806 0.874 0.798 0.754
### Evaluation of #51

<table>
<thead>
<tr>
<th>ID</th>
<th>VALUE</th>
<th>ID</th>
<th>VALUE</th>
<th>ID</th>
<th>VALUE</th>
<th>ID</th>
<th>VALUE</th>
<th>ID</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISA39</td>
<td>60.991</td>
<td>ISA39</td>
<td>67.479</td>
<td>AS-701</td>
<td>64.640</td>
<td>AS-704</td>
<td>60.522</td>
<td>AS-729</td>
<td>59.982</td>
</tr>
<tr>
<td>AS-372</td>
<td>68.141</td>
<td>AS-055</td>
<td>71.389</td>
<td>AS726A</td>
<td>71.771</td>
<td>AS-729</td>
<td>70.736</td>
<td>AS-822</td>
<td>65.368</td>
</tr>
<tr>
<td>DUMMY1</td>
<td>50.100</td>
<td>DUMMY2</td>
<td>87.383</td>
<td>DUMMY3</td>
<td>87.806</td>
<td>DUMMY4</td>
<td>62.806</td>
<td>DUMMY5</td>
<td>62.806</td>
</tr>
</tbody>
</table>
This AFM plot is in mol percent.

Ternary ratios for FeO+MgO, Na2O+K2O, MGO

<table>
<thead>
<tr>
<th>Sample</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISA39+</td>
<td>32.84</td>
<td>37.57</td>
<td>24.59</td>
<td>4</td>
</tr>
<tr>
<td>ISA39+</td>
<td>14.74</td>
<td>50.62</td>
<td>4.64</td>
<td>C</td>
</tr>
<tr>
<td>AS-701</td>
<td>35.47</td>
<td>49.52</td>
<td>14.01</td>
<td>1</td>
</tr>
<tr>
<td>AS-704</td>
<td>40.34</td>
<td>49.56</td>
<td>19.07</td>
<td>2</td>
</tr>
<tr>
<td>AS-706</td>
<td>47.52</td>
<td>36.62</td>
<td>15.86</td>
<td>3</td>
</tr>
<tr>
<td>AS-206</td>
<td>17.33</td>
<td>75.12</td>
<td>7.55</td>
<td>4</td>
</tr>
<tr>
<td>AS-372</td>
<td>21.06</td>
<td>75.53</td>
<td>3.41</td>
<td>5</td>
</tr>
<tr>
<td>AS-J95</td>
<td>12.74</td>
<td>77.66</td>
<td>9.60</td>
<td>6</td>
</tr>
<tr>
<td>AS-729</td>
<td>11.55</td>
<td>86.11</td>
<td>2.33</td>
<td>7</td>
</tr>
<tr>
<td>AS-822</td>
<td>19.06</td>
<td>65.44</td>
<td>4.50</td>
<td>9</td>
</tr>
<tr>
<td>AS-823</td>
<td>36.42</td>
<td>42.66</td>
<td>20.93</td>
<td>P</td>
</tr>
<tr>
<td>DUMMY1</td>
<td>40.38</td>
<td>41.19</td>
<td>18.33</td>
<td></td>
</tr>
<tr>
<td>DUMMY2</td>
<td>62.04</td>
<td>0.00</td>
<td>37.96</td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX C

GNAP source code

Listing 4. -- GNAP source code

GNAP OVERLAY DESCRIPTION.

\texttt{.ROOT GNAP=\{1,1\}XYLIBF4P/LB=R2**(Z,A,B,C,D,E)}
\texttt{.FCTR UTILITY}
\texttt{Z1 .FCTR PREPRO}
\texttt{A1 .FCTR NORM}
\texttt{B1 .FCTR FUNC-CONV}
\texttt{C1 .FCTR SMPNT=CLEAR}
\texttt{D1 .FCTR PRNT-PRPLOT=\{1,1\}XYLIBF4P/LB}
\texttt{E1 .FCTR TRIANG-SIDE=\{1,1\}XYLIBF4P/LB}
\texttt{.END}
A USER ORIENTED COMMAND LANGUAGE IS USED TO PROVIDE COMPLETE CONTROL
OVER THE COMPUTATION AND GRAPHIC OUTPUT OF THE STANDARD CIPW NORM
COMPUTATION. THE GRAPHIC OUTPUT FALLS INTO TWO CATEGORIES, THESE
CATEGORIES ARE X-Y PLOTS AND TERNARY DIAGRAMS. PLOTTING MAY BE DONE
ON THE LINEPRINTER OR ON AN OFFLINE PLOTTER (CALCOMP). PROVISION IS
ALSO MADE TO ALLOW THE USER TO EVALUATE AND PRINT ARITHMETIC
EXPRESSIONS GIVEN IN TERMS OF THE OXIDES AND NORMATIVE MINERALS
DETERMINED BY THE CIPW NORM CALCULATION.

THE ORIGINAL WEIGHT PERCENT OXIDES MAY BE PROVIDED IN A STANDARD
FORMAT OR THE USER MAY SPECIFY HIS OWN FORMAT FOR THESE VALUES, THESE
OXIDE VALUES MAY BE STORED AND MANIPULATED PRIOR TO THE NORM
DETERMINATION. PROVISION IS ALSO MADE FOR A COLUMNWISE SUMMARY
PRINTOUT OF THE STORED VALUES (BOTH ORIGINAL AND CALCULATED).

THE PROGRAM IS WRITTEN IN FORTRAN (IV) FOR THE IBM 360/65, REFERENCE
IS MADE TO A SYSTEM RESIDENT ROUTINE (CORE) WHICH PROVIDES IN-CORE
READ/WRITE CAPABILITIES. THE STANDARD CALCOMP PLOTTER SUBROUTINE
PACKAGE IS ALSO REFERENCED.

DATE OF LAST REVISION 1V/26/72

PROGRAMMED BY ROGER W. BOWEN

DEC 1977 - GNAP ADAPTED TO RUN ON THE D.G.M.R. PDP-11/45 COMPUTER BY
LESTER NORTH, U.S.G.S.

APR 1978 - MAJOR MODIFICATIONS MADE TO GNAP ON THE PDP-11/45 COMPUTER
FOR D.G.M.R. AND U.S.G.S. USE IN SAUDI ARABIA. LINES MODIFIED OR
ADDED HAVE, IN GENERAL, HAD THEIR SEQUENCE NUMBERS IN COLUMNS 73-80
REMOVED. THIS VERSION RUNS UNDER RSX11-D USING THE F4P COMPILER,
AND OCCUPIES A FULL PAGE OF CORE.

JOHN ODELL, D.G.M.R.

PARAMETER IV=20

BYTE FIL(32),BATCH,BAT

REAL NIG(12,IV )
DIMENSION X(IV+2),Y(IV+2),ADJUST(21,IV),AREA(1),VALUES(100,IV),
1 TTD( IV), RATIOS(2, IV), TOTALS(2, IV), TYPES(2, IV)
LOGICAL*1 CARD(80),STMT(160),FORMAT(96),NEXTC(4),BLNK,NAMÈ(8),
1 HEAD(75),TFORM(16),SYM(IV),SPFMT(17),RASSYM,FLAG
2 ,PLUS, ERH,STORE,NOTEQ,STAND,OFFINE,PRINTER,NONORM,SWCC,normal

INTEGER CNT,PT,BUFFER( 80 ),ORDER(21),UN,U
DOUBLE PRECISION IDENT(IV),RNAME,NAMES(100),DUMMY(12),BLANK,
1 FORMLD(2),DFILE(10),DFMT(3)

COMMON HEADG,TFORM,NORMAL,RASSYM

COMMON /COMM/ U,UN,FIL,BAT
THIS COMMON BLOCK ALIGNS VECTORS FOR ZEROING OUT

COMMON /DSPACE/ TTDI, ADJUST, VALUES, NIGGLI, RATIOS, TOTALS, TYPES

EQUIVALENCE (NAME(1), RNAME), (TTDI(1), AREA(1)), (DPFMT(1), SPFMT(1))

DATA ISM'/''/,, IBK'/''/
DATA PT, BLNK/80, '/,, DEFTLE/'PROGRAM', 'GNAP, NO', 'TITLE S', 1 'APPLIED', 6''/,, FORMLD'/''1NH1, 10X1, ',75A1/''/'
DATA NAMES/
  2 'S102', 'AL203', 'FE203', 'FEO', 'MGO', 'CAO', 'NA20',
  3 'K20', 'H20', 'TI02', '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', 'S', 'CL', 'F', 'S2', 'CR',
E 'NI', 'BA', '14''/''/

DATA BLANK'/''/
C DPFMT'/''10(1X,G1,'6.0'), 10X1, '', NEXTC,DUMMY/4''000', '10X9F4.4',
D '/''2.2F3.2', ',', 'F4.2', 'F3', ';', 'F4.2', ';', 'F5F3.2', 'F2', ';', '2',
E 6''/''/', PLUS/1X''/

INITIALIZE CONSTANTS TO DEFAULT VALUES

BATCH=.FALSE.,
S*CC=.FALSE.,
FLAG=.FALSE.,
110 CNT=0
  NCS=1
  NIM=86
  NOX=20
CALL MOVE (DUMMY, FORMAT, 96)
CALL MOVE (FORMLD, TFORM, 16)
CALL MOVE (DEFTLE, HEADG, 75)
PRINTER=.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

BAT=.FALSE.
IF (BATCH .OR. .FLAG) GO TO 910

CALL PREPRO(BATCH)

910 DEFINE = .FALSE.
140 LSTMT=0
150 PT=PT+1
IF (PT.LE.80) GO TO 210

IF (.NOT.BATCH) WRITE(U,123)
123 FORMAT( 240 GNAP>1$)
PT=1
READ (UN,74H,END=690) CARD
IF(NORMAL) GO TO 159

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

RASS SYMBOL PROCESSOR

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

TITLE PROCESSOR

159 IF (NOTEQ('TITLE',CARD,5)) GO TO 170
DO 160 J=1,75
160 HEAD(J)=CARD(J+5)
PT=80
GO TO 910
170 IF (NOTEQ('STOR',CARD,3)) GO TO 180
STORE=.TRUE.
GO TO 190
180 IF (NOTEQ('NRM',CARD,3)) GO TO 210

59
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 NORMCARD,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
   CLEAR STORAGE PROCESSOR * * * * * * * * * * * * *
   BY BRANCHING TO START, THEN OVERWRITE
   IF (LSTMT.GT.16) WRITE(U,900) (STMT(I),I=1,LSTMT)
   FLAG=.TRUE.
   GO TO 110
C 240 IF (NOTEQ(STMT,'SCALE',6)) GO TO 270
C
   SET THE SCALE FOR THE CALCOMP PLOTTER * * * * * * * * * * * *
   IF (LSTMT.GT.14) WRITE(U,900) (STMT(I),I=1,LSTMT)
   IF (PRINTER) 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 ****************************
C 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,NOX
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,NOX),RNAME
GO TO 690
C INCREASE NUMBER OF OXIDES AND STORE ORDER.
310 NOX=NOX+1
ORDER(NOX)=L
J=J+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 ****************************
C 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
C NORM STATEMENT PROCESSOR ****************************
C 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 ****************************
C 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 TO 360
PRINTER=.FALSE.
IF (SWCC) GO TO 350
CALL NEWDEV(3,'SIGNAP',PLT',11)
CALL PLOTST(U,01,'CM')
CALL FACTOR(2,54)
SNC0.TRVC,
J50 CALL PLOT (2,0,1,12,-3)
GO TO 910

PRINTER=True,
GO TO 910

C
370 IF (NOTEQ(STMT,'SUMMARY(1,8)) GO TO 375

C
370 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,TDI,NOX,ORDER,SYM,NUM,ADJUST)
GO TO 910

C
375 CONTINUE

IF (NOTEQ(STMT,'MODIFY',6)) GO TO 410

C
376 MODIFY COMMAND PROCESSOR **************

IF (NOTEQ(STMT(7),'FORMAT',6)) GO TO 390
DO 376 J=1,96

J*1,96

IF (LSTMT,GE,20) GO TO 910
CALL MOVE(STMT(22),STMT(7),LSTMT-21)
LSTMT=LSTMT-15

378 IF (INDEX(STMT,')',LSTMT),EQ,0) GO TO 380
I=INDEX(STMT,')',LSTMT)

I=INDEX(STMT,')',LSTMT)

IF (I,EQ,0) GO TO 380
J=INDEX(STMT,')',LSTMT)
CALL MOVE (STMT(I),FORMAT,J+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+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.NORM) 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,ERP)+2,1

62
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=MIN(B,I-1)
RNAME=BLANK
DO 420 L=1,J

420 NAME(L)=STMT(L)
LSTM=LSTM-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),LSTM,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,TFORM1) 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 (LSTM.GT.16) WRITE(U,900) (STMT(I),I=1,LSTM)
DO 490 I=1,CNT

490 CALL RECALC(VALUES(1,I),IDENT(I),SYM(I),TTOI(I),RATIOS(1,I),TOTAL
IS(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
LSTM=LSTM-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=1,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
560 VALUE CONVERT PROCESSOR
   IF (LSTMT,GT,13) WRITE(U,910) (STMT(I),I=1,LSTMT)
   CALL CONVER (VALUES,CNT)
   GO TO 910
C
570 IF (NOTEQ(STMT,'PLUT',4)) GO TO 560
C
580 X=Y PLOT PROCESSOR
   LSTMT=LSTMT+4
   STAND=NOTEQ(STMT(5),'(K)',3)
   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
   DL 690 I=1,CNT
590 Y(I)=EVAL(VALUES(1,I))
   CALL PRNT (X,Y,SYM,CNT,STMT(K),STMT(J),LE1,LE2,STAND, PRT,TER,S 1CALE)
GO TO 910

610 IF (NOTEQ(STMT, 'TERNARY', 7)) GO TO 680

C TERNARY DIAGRAM PROCESSOR

LSTMT=STMT-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=STMT-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(O, EVAL(VALUE3(I,1)))
CALL PARSE (STMT(J), LE3, NAMES, NUM, ERR)
IF (ERR) GO TO 910
DO 660 I=1, CNT
Y1=Y(I)
X1=X(I)
SUM=X1+Y1+AMAX1(O, EVAL(VALUE3(I,1)))
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 TF, FORM, HEADG
PRINT 850, (STMT(I), I=8, LSTMT)
PRINT 850
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, PRINTER, SCALE)
GO TO 910

65
UNRECOGNIZED COMMAND

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 SCALE')
720 FORMAT (' ',A8,'DEFINITION AS ',80A1)
730 FORMAT (' ',A8,'DEFINITION AS ',80A1)
740 FORMAT (80A1)
750 FORMAT ('1NO MORE THAN ',I3, ' NORMS MAY BE STORED. THE LAST NORM WILL BE ERASED')
760 FORMAT ('1STATEMENT LENGTH(160) EXCEEDED ON THE FOLLOWING STATEMENT')
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 IGNORES 1 FOR NAME=1,A8)
800 FORMAT ('1TECTION OF ',80A1)
810 FORMAT ('10 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 ('1',A6,F11.3,3X,A6,F11.3,3X,A6,F11.3,3X,A6
11.3,3X,A6,F11.3)
830 FORMAT ('1PLOT COMMAND ERROR ON ',80A1)
840 FORMAT ('1TERNARY COMMAND ERROR ON ',80A1)
850 FORMAT ('1TERNARY RATIOS FOR ',80A1)
860 FORMAT ('1SAMPLE R1',6X,'R2',6X,'R3 SYMBOL/')
870 FORMAT ('1',A6,3F8.2,4X,A1)
880 FORMAT ('1UNRECOGNIZED COMMAND GIVEN AS ',80A1)
890 FORMAT ('1OXIDE COMMAND CONTAINS A NAME WHICH IS NOT IN THE LIST
1IF ACCEPTABLE OXIDES, ACCEPTABLE OXIDES ARE111',11A8'/1',10A8/120
2HE UNRECOGNIZED NAME IS ',A8)
900 FORMAT ('1THE FOLLOWING COMMAND CONTAINS EXCESSIVE CHARACTERS.1
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 COELL, D.G.M.R.
C
SUBROUTINE PREPPRO(BATCH)
C
BYTE FIL(32),BATCH
INTEGER UN,U
COMMON /COMMC/ U,UN,FIL
C
UN=5
U=5
C
10 FORMAT ('1 GNAP -- BATCH OR TERMINAL? ------ ',$)

66
ACCEPT 11, TCHAR
11 FORMAT(A4)
   IF (TCHAR.EQ.'BATC') GO TO 15
   TYPE 12
12 FORMAT(' OUTPUT TO TERMINAL OR PRINTER? ------ ',S)
   ACCEPT 11, TCHAR
   IF (TCHAR.EQ.'TERM') GO TO 25
   GO TO 28
25 CALL ASSIGN(6, 'TII')
   GO TO 28
15 TYPE 17
17 FORMAT(' NAME BATCH FILE: ------ ',S)
   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,NAMES=FIL,TYPE='OLD',READONLY)
   GO TO 27
29 OPEN (UNIT=UN,NAMES='GNAP,BAT',TYPE='OLD',READONLY)
27 BATCH=.TRUE.
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 SALTIC
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 APPEARANCE,
C
C DEC 1977 - ADAPTED BY LESTER NORTH, U.S.G.S. TO RUN ON THE D.G.M.R
PDP-11/45 COMPUTER UNDER RSX11-D WITH F4P COMPILER.

APR 1976 - MODIFICATIONS BY JOHN ODELL, D.G.M.R. TO D.G.M.R. AND U.S.G.S. REQUIREMENTS IN SAUDI ARABIA.

******************************************************************

SUBROUTINE NORM (CARD,OUTM,STORE,FORMAT,NCS,TTDI,RATIOS,TOTALS, 1 TYPES,NIGGLI,IDENT,NOX,ORDER,NONORM,ADJUST,SYM,NAMES,CNT)

IMPLICIT REAL(M,N)

PARAMETER IV=20

COMMON HEADG,TFORM,NORMAL,RASSYM

COMMON /COMM/ U,UN,FIL,BAT

THESE COMMON BLOCKS ALIGN SCALARS AND VECTORS TO AVOID PARENTHESES

COMMON /ALIGNED/ WC1,WC2,WC3,WC4,WC5,WC6,WC7,WC8,WC9,WC10,WC11,WC121,WC13,WC14,WC15,WC16,WC17,WC18,WC19,WC20,WC21


REAL K,MC(21),OUTM(100),WC(21),PM(33),NP(11),MM(36),PC(21),MG,VALUES(100),IV,RATIOS(2),TOTALS(2),TYPES(2),NIGGLI(12)

REAL TTDI(1),RM(21),WM(33),ADJUST(21)

REAL*8 NAMES(100)

LOGICAL CARD(80),SYM(1),IDENT(6),FORMAT(96),SAM,BUFFER(240), 1HEAEG(75),TFORM(16),RASSYM,STORE,NONORM,NORMAL,BAT,FIL(32)

INTEGER I,T,ORDER(21),NOX,NCS,UN,U,CNT

EQUIVALENCE (WC1,WC1), (MM1,MM1)

DATA RM /1,6543143E-2,9,807652E-3,6,6262047E-3,1,391858E-2,2,4806 188E-2,1,783186E-2,2,61345E-2,1,961533E-2,5,550825E-2,1,25183E-2,2,7 2.845002E-3,1,409694E-2,8,115645E-3,2,272213E-2,1,249629E-2,2,82063 36E-1,5,263501E-2,3,118762E-2,6,579372E-3,1,3.3852E-2,6,521481E-3/ 1DATA WM /60,0848,101,9612,183,3036,556,6734,524,449,278,2102,43 16,5238,284,1698,316,3342,116,8855,142,6412,105,989,462,0104,122,66 238,154,2882,116,1642,100,3962,131,9312,140,7976,203,7776,172,2436,3231,5383,223,8363,159,6922,151,7449,196,63,135,9782,79,3988,336,2 4084,78,9768,119,975,100,0894,84,32135/

IF (BAT) GO TO 4006

IF (NOTEQ(CARD(5),"BATCH",5)) GO TO 1001

IF (CARD(11),NE,"I") GO TO 4001

58
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)=
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(' ','AB',........> 'S)
3001 FORMAT(F6.2)
1004 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 1010 I=2,NOX
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.
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.
DO 60 I=1,11
60 NP(I)=0.
GO TO 80
C
-----------------------------------------------------------------------------------------
C RECALC (OUTM, IDENT, SAM, TTDI, 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

5 CHARACTER SAMPLE ID. 'SAM' WILL ALREADY CONTAIN THE PLOTTING
SYMBOL WHICH WILL BE PRINTED. THE OTHER PARAMETERS ARE COMMENTED
UPON AT THE BEGINNING OF NORM (INCLUDING 'NONORM' AND 'ADJUST').

ENTRY RECALC(OUTM,IDENT,SYM,TDI,RATIOS,TOTALS,TYPES,NIGGLI,
1 NONORM,ADJUST)
DO 70 I=1,21
70 PC(I)=OUTM(I)
80 RATI0=0,0
R1=0,0
IF (PC(3),NE.,0,0) RATI0=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(I)=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,(WC(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 MM18=0,5*WC16
MM11=MIN1(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-MM25
    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=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=WC5-0.0
    IF (WC5.GE.0.0) GO TO 200
    T=11
    GO TO 470
200 MM17=WC5
    MM33=WC14
    WC1=WC1-MM13
    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=WC6
    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
    MB=WC2
    WC6=DIFF
WC2=0.0
GO TO 250

240 MM6=WC6
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
WC6=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
WC4=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)=1
MM35
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=MM35-1
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=1.0*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
   DIFF=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=FC2+2.0*MM16
   IF (FAC1.LT.0.0.OR.FAC2.LT.0.0.OR.FAC3.LT.0.0) GO TO 400
   MM21=0.25*FC3
   MM36=MM36+0.25*FC2
   MM34=0.5*FC1
   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

73
410 MM9=0.5*DIFF2
    MM7=0.5*DIFF1
420 DIEN=MM34
    DIN2=DI
    DIEN=1*DI
    DIFF=2*DI
    HY=MM36
    HYEN=1*HY
    HYFS=2*HY
    OL=MM36
    OLFO=1*OL
    MM19=OLFO
    OLFA=2*OL
    MM20=OLFA
    MM16=MM16+DIWO
    MM17=DIEN+HYEN
    MM18=DIFF+HYFS
430 DO 440 I=1,33
    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=13,24),(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)=DIFF*WM(18)
    NP(5)=NP(2)+NP(3)+NP(4)
    NP(6)=HYEN*WM(17)
    NP(7)=HYFS*WM(18)
    NP(8)=NP(6)+NP(7)
    NP(9)=OLFO*WM(19)
    NP(10)=OLFA*WM(20)
    WOL=MM16+DIWO
    NP(11)=WOL*WM(16)
    NP(12)=NP(9)+NP(10)
    TOTALS(2)=DIFF1
    TYPES(1)=DIFF2
    TYPES(2)=Y
    TTDI1(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,MY,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
   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
   M=SUM+WC9
   NIGGLI(8)=M
   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=100.0+3.0*AL+ALK
   QZ=SI=SIP
   NIGGLI(9)=K
   NIGGLI(10)=MG
   NIGGLI(11)=SIP
   75
NISOLH(I2)=E2
IF (NONORM) GO TO 549
PRINT 700, AL,FM,C,ALK,8I,FE,TI,PM,K,NG,SED83
549 DO 550 I=1,33
550 OUTM(I+41)*PM(I)
DO 550 I=1,11
560 OUTM(I+44)*NP(I)
570 CONTINUE
RETURN
C 4910 CONTINUE
C=CONT-1
IF (BAT) CLOSE (UNIT=2)
BAT=.FALSE.
RETURN
C **************************************************************
C CONVER(VALUES,CNT) IS USED TO CONVERT VALUES OF OXIDES AND NORMATI
C MINERALS FOR THE 'CNT'損害 STORED IN 'VALUES' FROM PERCENTAGES
C TO MOLECULAR AMOUNTS.
C ***************************************************************
ENTRY CONVER(VALUES,CNT)
DO 599 I=1,CNT
DO 599 J=1,21
590 VALUES(J,I)=VALUES(J,I)*RM(J)
DO 599 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(56,I)/WM(16)
VALUES(58,I)=SUM
VALUES(55,I)=X1+Y1+SUM
X1=VALUES(59,I)/WM(17)
VALUES(60,I)=X1
Y1=VALUES(61,I)/WM(18)
VALUES(62,I)=Y1
VALUES(60,I)=X1
VALUES(63,I)=X1+Y1
VALUES(64,I)=VALUES(63,I)/WM(19)
VALUES(65,I)=VALUES(64,I)/WM(20)
VALUES(66,I)=VALUES(65,I)/WM(16)
RETURN
C 610 FORMAT (80A1)
620 FORMAT (1 ORIGINAL WT,PCT, OXIDES/0 8I02 AL2O3 FE2O3 FEO ,
1'MGO CAO NAO2 K2O H2O TI02 P2O5 MNO ZRO2 CO2 ',30
23 CL F 8 CR2O3 NIO BOO'/1 ',21F6,2/'0SUM OF ORIGINA
3L OXIDES=1',F6,2;'0CIPW NORM FOR SAMPLE NO. ',6A1,69X,'PLOTTING SYM
48O1 18 ',41)
**630 FORMAT ( '0CONSTITUENTS', SI02, AL203, FE203, FEO1, 6X, 'MGO
1 CAO', 6X, 'NA2O', K2O1, 6X, 'H2O1, 6X, 'TIO2', 5X, 'P2O5 AL2O3/SI02', 1 PERCENTAGES', 11F9.2, 10F3.3, ' MOL. AMTS.', 1, 11F9.4/0CONSTITUENTS
4 NIO BAO, 12X, 'FEO/FE203', 1 PERCENTAGES', 10F9.2, 10F3.3/0CONSTITUENTS
5 MOL. AMTS.', 1, 10F9.4)
6 TOTAL SALIC FEMIC', 1, 'MOL. AMTS.', 1, 9F9.4)
**650 FORMAT ( '0MINERALS', 12F9.3/10MINERALS', 9X, 'DI', DI= '401, 4X, 'DI1-EN', DI= 'FS', HY HY= '2A', HY= 'FS', OL OL= 'FO', 4X, 'OL= '2A
50 MOL. AMTS.', 1, 11F9.4/10PERCENTAGES', 11F9.3)
**660 FORMAT ( '0THORNTON + TUTTLE DIFFERENTIATION INDEX = ', 7F3.3)
**670 FORMAT ('0NORM NOT COMPUTABLE FOR SAMPLE NO. ', 6A1/0SEE ERROR CODE
2E1, 13/', OF PROGRAM WRITE-UP')
**690 FORMAT ( '0BARTHS CATIONS SI AL FE3+3 FE+2 MGI, 6X, 1CA NA1, 7X, 'K', 7X, 'H', 6X, 'TI', 7X, 'P', 6X, 'MN', 1, '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 'PRINTER' 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,PRINTER,
X SIZE)
C INTEGER CNT
BYTE SYM(CNT), E1(LE1), E2(LE2), E3(LE3), PAGE(101, 51), PRINTER
BYTE BLNK, STAR, ZERO, MODS, CHAR(4), HEADG(75), TFORM(16), LINK(8)
DIMENSION X(1), Y(1)
REAL IDENT(CNT), BAD(45), MORE
C COMMON HEADG, TFORM
C DATA BLNK, ZERO, STAR, MORE/ '1, '1, '1,'1, 'MORE/
DATA CHAR/4= '0630/
LINK(7)=1/1
C
IF (PRINTER) 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)
X1=9.05*0,12*LE3
CALL SYMBOL (X1,-0.3,0,21,E3,0,0,LE3)
CALL SIDE (120,0,9.05,0,0)
X1=4.54*0,12*LE1
CALL SYMBOL (X1,7.96,0,21,E1,0,0,LE1)
CALL SIDE (120,0,4.54,7.06)
C
PLOT THE POINTS
DD 50 I=1,CNT
Y1=7,8*4E-2*Y(I)
X1=9.05*E-2*X(I)+Y1/1.7321
IF (X1,NE,0,0,0,0,OR,Y1,NE,0,0) CALL SYMBOL (X1=0.05,Y1=0.07,
X=0.380/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((I)).
DD 100 I=2,50
LF=61-I
DD 60 J=1,LF
80 PAGE(J,I)=BLNK
LF=LF+1
PAGE(LF,I)=STAR
MODS=MOD(I=1,5),EQ,0
IF (MODS) 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 (MODS) 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=6
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 (X,Y)',PAGE(1,3),30)
M=3
DO 160 I=1,K
J=MOD(I-1,3)
IF (J.EQ.0) M=K+1
LF=10*J+1
160 CALL MOVE (BAD(I)),PAGE(LF,M),8)
170 PRINT TFORM,MEAOG
PRINT 1BP, (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 290, (E3(I),I=1,LE3)
RETURN
180 FORMAT (' ',55X,40A1)
190 FORMAT (' ',5X,101A1)
200 FORMAT ('+',105X,27A1)
END

C **********************************************************************
C PRINT(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
LENGTHS ARE 'LE1' AND 'LE2'. 'STAND' IS A LOGICAL VARIABLE WHICH TELLS PRNT THE POSITIVE DIRECTION OF THE X-AXIS. IF TRUE, X WILL INCREASE TO THE RIGHT. 'PAGE' IS AN AREA WHICH WILL BE USED IN THE PLOTTING. 'SYM' IS AN ARRAY OF CNT PLOTTING SYMBOLS, ONE FOR EACH POINT IN X AND Y. 'PRINTER' IS A LOGICAL VARIABLE GIVING THE PLOTTING DEVICE (.TRUE. IS LINEPRINTER). 'SIZE' IS USED TO FIX THE PLOTTING POINT SIZE AT 0.14".

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)

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 = 10.0 * XMAX + XMIN
DO 8 I = 1, CNT
X(I) = XMIN + XT - X(I)
GO TO 40
5 XMAX = X(I)
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
**PRPLOT** is a modified version of the routine by the same name written at Univ. of Michigan. See program documentation for details.

This version modified by John Odell D.G.M.R. for GNAP

**SUBROUTINE PRPLOT**

**IMPLICIT LOGICAL*1(W), LOGICAL*1(K)**

**COMMON HEADG, TFORM**

**COMMON /COMM/ U**

**DIMENSION ABNOS(11)**

**LOGICAL*1 NOS(10), IMAGE(5152), CHAR LABEL(1), XLAB(1)**

**LOGICAL*1 HC, NC, BL, HF, FOR1(19), FOR2(15), FOR3(19)**

**LOGICAL*1 CHAR(4), HEADG(75), TFORM(16), STAND**

**INTEGER U**

**EQUIVALENCE (NPT, CHAR(1))**

**DATA NOS /'0', '1', '2', '3', '4', '5', '6', '7', '8', '9'/**

**1 HC = '1', NC = '1', BL = '1', HF = '1', FOR1 = '1',**

**2 FOR2 = '(', '1', 'X', 'A', '1', '1', '1', '1', 'F', '1',**

**3 '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1',**

**4 FOR3 = '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1',**

**5 '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1',**

**DATA FOR1 /'(', '1', 'X', 'A', '1', '1', '1', '1', 'F', '1',**

**1 '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1',**

**1 '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1',**

**DATA KPLT1 / .FALSE., KPLT2 / .FALSE. /*

**110 NVP = NV + 1**

**NDH = 5 * NSH**

**NDHP = NDH + 1**

**NDV = NV + NSV**

**NDVP = NDV + 1**

**NIMG = (NDHP * NDVP)**

**FOR1(10) = NOS(4)**

**NA = MIN(4, NV) = 1**

**NB = 11 + MIND(NA, 120 - NDV)**

**11 = NB / 10**

**12 = NB - 11 * 10**

**FOR3(6) = NOS(11 + 1)**

**FOR3(7) = NOS(12 + 1)**

**FOR3(9) = NOS(NA + 1)**

**11 = NV / 10**

**12 = NV - 11 * 10**

**FOR3(11) = NOS(11 + 1)**

**FOR3(12) = NOS(12 + 1)**

**FOR3(13) = HF**

**11 = NSV / 100**

**13 = NSV - 11 * 10**

**12 = 13 / 10**
I3*I3-12+10
FOR3(14)=NO8(I1+1)
FOR3(15)=NO8(I2+1)
FOR3(16)=NO8(I3+1)
FOR3(17)=HP1
FOR3(18)=FOR3(9)
KPL0T1=TRUE.
GO TO 120
ENTRY PLOT2(IMAGE,XMAX,XMIN,YMAX,YMIN,WPRNT,SIZE)
IF (WPRNT) GO TO 120
*PLOT*.FALSE.
SCALE*SIZE
XMIN1=XMIN
YMIN1=YMIN
DV=XMAX
DM=XMAX
IF (ABS(DV-XMIN1).LT.1.E-6) DV=XMIN1+1.0
IF (ABS(DM-YMIN1).LT.1.E-6) DM=YMIN1+1.0
RETURN
120 KPL0T2=.TRUE.
*PLOT*.TRUE.
IF (KPL0T1) GO TO 130
NSH=10
NS=10
NSV=10
GO TO 110
130 CONTINUE
YMAX=YMAX
DM=(YMAX-YMIN)/(NDH)
DV=(XMAX-XMIN)/(NDV)
DO 140 I=1,NVP
ABN03(I)=(XMIN+((I-1)*NSV)*DV)
DO 150 I=1,NIMG
IMAGC(I)=BL
DO 185 J=I1,I2
KNH0R=MOD(I-1,NSH),NE,0
IF (KNH0R) GO TO 170
DO 160 J=11,I2
160 CONTINUE
DO 185 J=11,I2,NSV
IF (KNH0R) GO TO 185
IMAGE(J)=HC
170 CONTINUE
DC 185 J=11,I2,NSV
IF (KNH0R) GO TO 185
IMAGE(J)=NC
185 CONTINUE
XMIN1=XMIN=DV/2.
YMIN1=YMIN=DM/2.
RETURN
ENTRY PLOT3(CH,X,Y)
DUM1=(X-XMIN1)/DV
DUM2=(Y-YMIN1)/DV
IF (WPLT) 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,XLAH,STAND)
IF (WPLT) GO TO 200
CALL AXIS (0,0,0,0,LABEL,NL,7,90,0,YMIN1,DV)
C
THIS CODE IS NECESSARY TO REVERSE THE AXIS
C
YP=-0.1
CALL PLOT(YP,YP,3)
DO 300 I=1,10
XP=I-1
XP2=I
CALL PLOT(XP,YP,2)
CALL PLOT(XP2,YP,2)
CALL PLOT(XP2,YP,2)
300 CONTINUE
YP=-0.25
FP=XMIN1
DO 314 I=1,11
XP11=I-0.1
CALL NUMRFH(XP,YP,?1,FP,0.2)
314 CONTINUE
YP=0.5
XP5=0.5
CALL SYMBOL(XP,YP,0.14,XLAH,V,XNL)
GO TO 199
198 CALL AXIS (I,0,0,0,0,0,0,0,0,0,0,0,YMIN1,DV)
199 CALL PLOT (15,1,0,0,0,3)
RETURN
200 CONTINUE
PRINT TFMR,HEANG
DO 220 I=1,NDHP
NL=BL
IF (I.LE.NL) XL=LABEL(I)
12=I*NDVP
11=12-NDVP
IF (MOD(1-1,NSH).EQ.4) GO TO 210
```assembly
PRINT FOR2, ML, (IMAGE(J), J=11,12)
GO TO 220
210 CONTINUE
ORDNO=YM* (I=1)*DH)
PRINT FOR1, ML, ORDNO, (IMAGE(J), J=11,12)
220 CONTINUE
PRINT FOR3, (ABNO(J), J=1, NVP)
PRINT 240, (XLAB(J), J=1, NXL)
RETURN
C
240 FORMAT (1,50X,70A1//)
END
C
FUNCTION CONV (A,N1,N2,E,ERR) RETURNS THE NUMERIC VALUE OF A DECIMAL NUMBER
in character form in the string A. A(N1) is the leading digit and
A(N2) is the trailing digit. A is assumed to be at least N2 bytes
length. E is used as it is in the F(W,E) format code (ie. if no
decimal point appears it will be assumed in the E position, if pre
a decimal point overrides E), ERR is a logical variable indicating
whether or not the conversion was successful.
C
FUNCTION CONV (A,N1,N2,E,ERR)
LOGICAL *1 A(I), 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
C
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)
```

84
**Subroutine CLEAR (WIDTH, LENGTH, PAGE)**

INTEGER LINE, COLUMN, LENGTH, WIDTH, SIZE, IS
LOGICAL PAGE(131, 65), FORMAT(1), BLANK, STRING(1)

DATA BLANK/' '/

**This subroutine is a collection of entries designed to facilitate printing of lists in column-wise rather than row-wise fashion. 'WIDTH' and 'LENGTH' are integers which give the dimensions of the array 'PAGE' which is LENGTH*WIDTH BYTES in size, 'FORMAT' is an array which contains the object-time format of the item to be written on line 'LINE' starting in column 'COLUMN', 'SIZE' is an integer which gives the number of bytes specified in format.**

SUBROUTINE CLEAR(WIDTH, LENGTH, PAGE)
INTEGER LINE, COLUMN, LENGTH, WIDTH, SIZE, IS
LOGICAL PAGE(131, 65), FORMAT(1), BLANK, STRING(1)

DATA BLANK/' '/

**Blank out page**

DO 10 J=1, LENGTH
DO 10 I=1, WIDTH
10 PAGE(I, J)=BLANK
RETURN

**R(REAL) S(CALER) W(RITER)**

ENTRY RSW(LINE, COLUMN, PAGE, FORMAT, SIZE)
CALL CORE(PAGE(COLUMN, LINE), SIZE)
WRITE(99, FORMAT) RS
ENCODE(SIZE, FORMAT, PAGE(COLUMN, LINE)) RS
RETURN

**I(NTEGER) S(CALER) W(RITER)**

ENTRY ISW(LINE, COLUMN, PAGE, FORMAT, SIZE)
CALL CORE(PAGE(COLUMN, LINE), SIZE)
WRITE(99, FORMAT) IS
ENCODE(SIZE, FORMAT, PAGE(COLUMN, LINE)) IS
RETURN

**S(TRING) A(RRAY) W(RITER)**

ENTRY SAM(LINE, COLUMN, STRING, SIZE)
DO 20 I=1, SIZE
20 PAGE(COLUMN*I-1, LINE)=STRING(I)
RETURN
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 (1H1, 131A1)
END

******************************************************************
THIS SUBPROGRAM IS USED TO EVALUATE ARITHMETIC EXPRESSIONS.
These expressions are composed of variable names stored in NAMES,
the arithmetic operators (unary minus, /, *, =, and +), constants,
and the grouping symbols (). Evaluation is performed in the order
(given above) of the arithmetic operators in a left to right
fashion. This order can be overridden by the use of ()

FIRST A CALL TO PARSE(EXPR,L,NAMES,N,ERR) MUST BE MADE. EXPR
contains the character representation of the expression and is L
bytes (characters) in length. NAMES is a real*8 array containing
the n names which might appear in expr. ERR is logical*1 and is
set to .true. If the expression is syntactically incorrect or
contains a variable which is not in NAMES.

PARSE generates the reverse polish form of expr in the two arrays
POLISH and TYPE. POLISH contains the numeric values of any
constants in expr. TYPE contains three types of information.

IF TYPE(I) = 5 THEN THE I'TH POLISH FORM ITEM IS UNARY MINUS.

IF TYPE(I) = 4 THEN THE I'TH POLISH FORM ITEM IS THE OPERATOR /

IF TYPE(I) = 3 THEN THE I'TH POLISH FORM ITEM IS THE OPERATOR *

IF TYPE(I) = 2 THEN THE I'TH POLISH FORM ITEM IS THE OPERATOR =

IF TYPE(I) = 1 THEN THE I'TH POLISH FORM ITEM IS THE OPERATOR *

IF TYPE(I) = 0 THEN THE I'TH POLISH FORM ITEM IS A CONSTANT FOUND
IN POLISH(I).

IF TYPE(I) > 0 THEN THE I'TH POLISH FORM ITEM IS A VARIABLE WHOSE
VALUE IS IN VALUES(TYPE(I)).

A TRANSITION MATRIX TECHNIQUE IS USED FOR THE PARSE.

THE ENTRY EVAL(VALUES) MAY THEN BE USED TO RETURN THE NUMERIC
VALUE OF THE EXPRESSION BASED ON THE CURRENT VALUES IN VALUES
CORRESPONDING TO VARIABLES IN NAMES. A SIMPLE PUSH-DOWN STACK
TECHNIQUE IS USED.

******************************************************************
MODIFIED BY JOHN ODELL D.G.M.R. FOR GNEP FEBRUARY 1978

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)

INTEGER*2 INDEX, TYPE(41)
REAL NAMES*8(1), VALUES(1), POLISH(41), STACK(41), VARHLE*8
COMMON /CUMM/ U
EQUIVALENCE (NAME(1), VARHLE)

DATA TM /12, 32, 0, 81, 2*41, 101, 2*92, 93

86
1,21,22,23,113,32,33/  DO 340 J=1,L  
DATA CHAR/4*000/,BLANK/1 1/  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,L  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=INDEX + INDEX  
IF(INDEX .EQ. 0) GO TO 295  
VT=STACK(TOP)  TOP=TOP+1  
GO TO (310,320,330,320),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  SET OPERATOR CODES  
UNARY=-5.  
DIV=-4.  
PROD=-3.  
DIFF=-2.  
PLUS=-1.  
PAREN=0.  
C  CONSTRUCT POLISH FORM VIA TRANSITION MATRIX TM  

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
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
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 ('ERROR IN EXPRESSION ',A50)
490 FORMAT ('UNDEFINED NAME ',A8)
500 FORMAT ('DIVIDE BY ZERO ATTEMPTED, HENCE!')
END
C ******************************************************************************
C SUBROUTINE MOVE (A,B,N) MOVES 'N' BYTES (IE CHARACTERS) FROM THE ARRAY 'A' TO
C THE ARRAY 'B'.
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 FUNCTION 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 ******************************************************************************
FUNCTION INDEX (STRING,CHAR,N)
LOGICAL*1 STRING(N),CHAR,TEM(4) ,TEM(4)
EQUIVALENCE (TEM(1),NS), (TEM(1),NC)
   DATA TEM,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
NOTEQ(A,B,N) RETURNS A VALUE (TRUE OR FALSE) INDICATING WHETHER OR NOT THE TWO STRINGS A AND B ARE EQUAL IN THE FIRST N CHARACTERS.

LOGICAL FUNCTION NOTEQ(A,B,N)
LOGICAL A(I),B(I),TA(4),TB(4)
EQUIVALENCE (TA(I),NA), (TB(I),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