D A T A S A V E M A N U A L

by

James E. Quick

U. S. Geological Survey Open-File Report OF 88-213

CONTENTS

S E C T I O N 1

I N T R O D U C T I O N

Contents

OVERVIEW

Originally written as a small data management program to store electron microprobe data, DATASAVE has grown into a family of programs that collectively facilitate storage, editing and graphical presentation of numerical data on IBM PC and compatible computers. DATASAVE is specifically designed to manipulate element and oxide abundances, but it allows manipulation of any numerical information.

DATASAVE consists of six programs. The program, DATAEDIT, enables data storage, formating and editing. BLEND allows recombination of data. DATALIST prints contents of data files to the screen, disk or printer. DATAPLOT allows presentation of data in orthogonal, ternary and histogram form. Graphs may be created using either the CGA (color-graphics adapter, 640x200 pixel) or EGA (enhanced-graphics adapter, 640x350 pixel) standards. Graphs may be stored on disk and reproduced on a dot-matrix or laserjet printer. DATANORM calculates CIPW and Barth-Niggli norms for output to screen, printer, or disk. SPLICE is a programming utility package that may be merged to new application programs to manage disk input and output of data.

DATASAVE is designed to provide a high degree of flexibility. Specialized databases and graphical presentations may be created and modified. Users who are conversant with BASIC may extend the DATASAVE family of programs by writing new applications in IBM or Microsoft BASICA and utilizing SPLICE.

Before any of this is possible, however, the new user must master DATAEDIT and DATAPLOT. This entails (1) becoming familiar with the program conventions and menu options, (2) understanding the way in which DATASAVE handles data, and (3) learning to program graphics subroutines. To assist the new user, a brief description of DATASAVE files and conventions is presented in Section 2 and a short tutorial is presented in Section 3. Examples of data and graphics routines that are required to to perform the tutorial are provided on Disk 2 (SPLICE/DATA) of the accompanying disks.

SYSTEM REQUIREMENTS

DATASAVE programs will run on an IBM PC, XT, AT or compatible computer having at least 384 Kb (kilobytes) of memory and at least one disk drive. Two disk drives are recommended. A color-graphics adapter (CGA) or enhanced-graphics adapter (EGA) is required to run DATAPLOT. Graphs may be reproduced on Epson, IBM or Prowriter dot-matrix printers. A Mouse Systems or compatible mouse may be used to edit graphs.

All of the programs will run under IBM- or MS-DOS 2.0 or higher. DATASAVE programs with "EXE" extensions require the QUICKBASIC runtime module, BRUN30.EXE, which is provided on Disk 1. Operation of SPLICE requires IBM- or MS-BASICA.

Special care should be taken before running DATASAVE in conjunction with memory-resident programs such as SIDEKICK by Boreland. Such programs may assign "hot keys" that override the key assignments of DATASAVE.

INSTALLATION

Two disks accompany this manual. Disk 1 is titled "PROGRAMS" and contains DATASAVE programs. Disk 2 is titled "SPLICE/DATA" and contains SPLICE plus examples of data and graphics subroutines for use with the tutorial. The exact disk contents should be as follows.

File	Full name or function
Disk 1:	
BRUN30.EXE	QUICKBASIC runtime module.
MENU.EXE	This routine displays the DATASAVE Menu.
E.EXE	This is DATAEDIT.
L.EXE	This is DATALIST.
B.EXE	This is BLEND.
P.EXE	This is DATAPLOT.
N.EXE	This is DATANORM.
CODES	An example of a sample-code file.
Disk 2:	
SPLICE.BAS	This is SPLICE in IBM BASICA.
SPLDRV.BAS	This is SPLICE DRIVER.
CODES	An example of a sample-code file.
BASALT	A data file of basalt compositions.
KSI.SUB	Subroutine for a K2O-SiO2 plot.
KSI.SCN	CGA image file for a K2O-SiO2 plot.
KSI.PAR	Paramater file for a K2O-SiO2 plot.
AFM.SUB	Subroutine for an AFM ternary plot.
AFM.SCN	CGA image file for an AFM ternary plot.
AFM.PAR	Parameter file for an AFM ternary plot.
SI.SUB	Subroutine for an SiO2 histogram.
SI.SCN	CGA image file for an SiO2 histogram.
SI.PAR	Parameter file for an SiO2 histogram.
TIZR.SUB	Subroutine for Pearce-Cann Ti vs. Zr plot.
TIZR.SCN	CGA image file for Ti vs. Zr plot.
TIZR.PAR	Parameter file for Ti vs. Zr plot.
TIZRY.SUB	Subroutine for Pearce-Cann Ti-Zr-Y ternary plot.
TIZRY.SCN	CGA image file for Ti-Zr-Y ternary plot.
TIZRY.PAR	Parameter file for Ti-Zr-Y ternary plot.
REE.SUB	Subroutine for chondrite-normalized, rare-earth element plot.
REE.SCN	EGA image file for rare-earth element plot.
REE.PAR	Parameter file for rare-earth element plot.

DATASAVE and related programs are not copy protected and may be transferred to new disks using standard DOS commands. It is recommended that the original disks be copied as a precaution against disk failures.

Disk 1 is formatted with DOS 2.1. If you are operating under DOS 3.0 or higher, it is absolutely necessary to copy the contents of Disk 1 onto a disk that has been formatted with DOS 3.0 or higher.

Two Drive System

To create a bootable program disk:
(1) Place a blank disk in drive B.
(2) Place a DOS disk in Drive A.
(3) Type the following.

FORMAT B:/S

(4) Place on of the PROGRAMS disks in drive A.
Use 1A if you are running under DOS 2.
Use 1B if you are running under DOS 3.
(5) Type the following.

COPY A:*.* B:

To backup Disk 2:
(1) Place a blank disk in drive B.
(2) Place a DOS disk in Drive A.
(3) Type the following.

FORMAT B:

(4) Place the Disk 2 in drive A.
(5) Type the following.

COPY A:*.* B:

Hard Disk System

To install on a hard disk:
(1) Make the hard disk the default drive.
(2) Select the desired subdirectory on the hard disk.
(3) Place on of the PROGRAMS disks 1 in drive A. Use disk 1A if you are running DOS 2. Use disk 1B if you are running DOS 3. Type type the following.

COPY A:*.*

(4) Place Disk 2 in drive A and type the following.

Copy A:*.*

ACKNOWLEDGMENTS

IBM, IBM-DOS and IBM-BASICA are trademarks of International Business Machines Corporation. MS-DOS and MS-BASICA are trademarks of MicroSoft Incorporated. The BRUN30.EXE runtime module is the copywrited property of MicroSoft Incorporated; it may be distributed with copies of DATASAVE as long as its name and contents are not modified. This manual and compiled DATASAVE programs are in the public domain. The source code for these programs (excluding SPLICE.BAS and SPLDRIVE.BAS) remains the private property of James E. Quick.

DATASAVE has evolved in response to suggestions from users. D. Stoeser, R. Kamilli and particularly E. duBray have made significant contributions. This manual was improved by reviews by E. duBray and R. Taylor.

S E C T I O N 2

F I L E S , C O D E S A N D C O N V E N T I O N S

Contents

INTRODUCTION

This section presents terms, concepts and conventions that are crucial to operating DATASAVE programs. Although this section should be read first, learning to operate new programs is facilitated by hands-on experience. Therefore, it is recommended that the new user read this section quickly and proceed directly to the tutorial (Section 3). The user will probably benefit from a second reading of Section 2 after working though the tutorial.

FILES

DATASAVE uses five types of files. (1) Data files contain the user's data. (2) Sample code files are used to format and interpret the users data. (3) Subroutine files contain user-written programs for creating graphs. (4) Image files are used to store graphics generated by DATAPLOT. (5) Parameter files contain information required by DATAPLOT to use image files as templates for new plots.

Data files and sample-code files are created and maintained with DATAEDIT. Subroutine files, image files, and paramater files are created and maintained with DATAPLOT. DATAPLOT supports the creation of templates for rapid production of graphs that have standardized formats. A template consists of an image file, a parameter file and a subroutine.

DATA FILES

Data files are subdivided into discrete data sets that will be referred to as samples. A data file may contain up to 100 samples. A sample could be a whole-rock composition of a single rock or an single electron probe analysis of a pyroxene. A sample is stored as (1) a sample name, followed by (2) a sample code, followed by (3) up to 50 numerical items that will be referred to as entries.

Sample names may be up to 12 characters long. For example, a sample name could the the name of the rock upon which the analysis was made. A sample name may be written with upper- and lower-case letters, numbers and any of the symbols on the key board.

Sample codes are three character codes that are used to identify entries of the sample. When the sample code is encountered, DATASAVE programs look up the code in a sample-code file where entry labels are found. These entry labels are used to identify the following entries (see Sample-Code Files).

An entry could be an oxide abundance expressed as weight percent or an elemental abundance expressed in parts per million (ppm) or parts per billion (ppm). An entry could also be a quantity unrelated to concentration (elevation, temperature, etc.) expressed as a floating point number. All entries are stored in a compressed format to conserve disk space and accelerate file input/output. Because of the compressed format, attempts to inspect a data file from DOS will result in unintelligible character sequences.

During data entry or editing, the entire contents of a data file is held in memory where it is modified. Data files on disk are never modified until (1) the user chooses to back up the data in memory, or (2) data entry or editing is complete. Following data entry or editing, the entire contents of the data file is written to disk. Samples may not be read from or written to disk individually.

SAMPLE-CODE FILES

Sample-code files are used by DATASAVE programs to interpret the entries in each sample. A sample-code file contains up to 50 sample codes. A sample code is stored with a list of up to 50 entry labels that identify the entries of a sample during data entry or recall. For example, the sample codes, PYX and OL, might be created to define pyroxene and olivine analyses, respectively. These two sample codes could be stored with the following sequences of entry labels.

PYX:	OL:
NA2O	MGO
MGO	SIO2
AL2O3	CAO
SIO2	MNO
CAO	FEO
TIO2	NIO
CR2O3	
MNO	
FEO	

When DATAEDIT stores an analysis in a data file, it records a sample code followed by a string of numbers that are the concentrations arranged in the order specified by the entry labels of the sample code. Conversely, when an application program reads a data file, it looks up the each sample code in the sample-code file to determine what label is assigned to each concentration.

Sample codes are an essential mechanism to enable data input and storage. No DATASAVE program will run without a sample-code file containing the relevant sample codes.

SUBROUTINE FILES

A subroutine file contains one user-written program for constructing a graph. Subroutine files are always stored with the extension ".SUB".

IMAGE FILES

An image file contains all of the information necessary to recall a graphics image from disk and display it on the screen. Image files are always stored with the extension ".SCN".

PARAMETER FILES

A parameter file contains the information necessary to plot additional data on a graphics image that has been recalled from disk. This consists of the locations of the corners of the graph on the screen, the numerical values of the corners, and the name of the subroutine used to generate the graph. Parameter files are always stored with the extension ".PAR".

CONVENTIONS

DATASAVE programs utilize the following screen and keyboard conventions:

1. Programs are presumed to reside in the default drive.
2. All files names must follow DOS conventions.
3. Extensions to file names are restricted as follows.
    a. All data and sample code files may utilize 3 character extensions.
    b. Extensions for subroutine, image and parameter files are assigned by DATASAVE; the user should never assign extensions for these types of files.
4. Programs prompt the user with mixed lower/upper case statements.
5. Keyboard input should always produce upper case letters because many aspects of DATASAVE involve string recognition. DATASAVE sets the letter case at the beginning of each program; do not toggle the Caps Lock key.
6. Selections from any menu are made with single key strokes and do not require a carriage return.
7. Answers to Yes/No questions are always "Y" or "N" without a carriage return.
8. Error codes arising from user errors correspond to numbered error codes listed the the IBM or Microsoft BASIC manual.
9. At the query, "Drive (A or B)?", respond with any existing drive (A, B, C, etc.) followed by a carriage return. Do not input a colon after the drive letter.
10. In response to queries for file names or drive specifications, a carriage return will default to the most recently used file name and drive.

TECHNICAL CONSIDERATIONS

SAMPLE CODES

Above all, do not rearrange the entry labels of a sample code after data files have been created with that code! The contents of a data file are not automatically adjusted to compensate. They will continue to be stored in the original order and information will be scrambled during subsequent operations!

MAXIMUM PERFORMANCE

The speed of DATASAVE programs is enhanced by minimizing the number of different sample codes per data file and per sample-code file. The following suggestions will speed up program execution and help maintain sample code files.

1. Keep a master sample-code file named CODES on your program disk in which all sample codes are defined. Following a boot, all DATASAVE programs will automatically look for this file unless instructed otherwise.

2. Maintain small sample-code files on data disks that define only the sample codes present on each disk. Use these sample-code files to run application programs.

3. Adopt a standard data format whenever possible. Input data in whatever format is convenient. Later, convert it to a standardized format for final storage.

MEMORY

It is theoretically possible for DATASAVE programs to exceed available memory under conditions of extreme file sizes and long subroutines. The largest possible data file would consist of 100 samples with each sample containing 50 entries, and would require 31.2 kilobytes of string space in memory. A completely full sample-code file would contain 50 sample codes with each sample code having 50 entry labels, and would require 12.65 kilobytes of string space in memory. The largest possible graphics program, 300 steps, would require 10 kilobytes of string space in memory. Therefore, DATAPLOT could conceivably use 45.77 kilobytes of string space to support one resident data file, a sample code file and a graphics subroutine. DATASAVE programs require additional string space to support other variables but, unfortunately, 64 kilobytes is the maximum data space available to DATASAVE programs for strings. The 64-kilobyte wall has not been hit by any user to date. Nevertheless, users are cautioned to use small sample-code files when accessing extremely large data files or long subroutines. This will give an added benefit of greater speed of execution.

S E C T I O N 3

D A T A S A V E T E S T R I D E

Contents

INTRODUCTION

This tutorial is designed to give the user a brief "test ride" with each of the DATASAVE family of programs. To assist in this, a sample data file consisting of 25 Hawaiian basalt analyses and number of graphics subroutines and templates are provided on Disk 2. It is emphasized, however, that the new user will not emerge from the tutorial as a DATASAVE expert because a complete introduction to all of the capabilities of each program is not attempted here. That would require an prohibitively long tutorial. Instead, the tutorial consists of a list of directions that, if followed exactly, will demonstrate some of DATASAVE's potential. To achieve the maximum results with each program, the user is advised to consult the detailed program descriptions in Sections 4 through 11.

The tutorial is divided into 8 lessons. Lesson 1 covers the machinations of loading DATASAVE and getting the DATASAVE main menu on the screen. All subsequent lessons pick up at the end of Lesson 1. Playing with data is usually more interesting than entering it into computers. Therefore, the early lessons cover the use of DATAPLOT (Lessons 2 and 3), DATANORM (Lesson 4) and DATALIST (Lesson 5). Using DATAEDIT and BLEND to create and maintain sample-code and data files is deferred to Lessons 6 through 9.

CONVENTIONS

This tutorial is written with the following implicit assumptions.

1. DATASAVE has been been installed following the instructions in Section 1.

2. DATASAVE programs are resident on the default drive.

3. The default drive is "A". Users with hard disk drives will want to substitute the appropriate letter ("C", "D", etc.) for "A" when it appears in this tutorial as a drive designation.

4. Disk 2 (SPLICE/DATA) is in drive "B". If the contents of Disk 2 have been placed in a different drive, the user will need to substitute the letter of that drive for "B" when it appears in this tutorial as a drive designation.

The tutorial requires the user to type letters and words and strike keys. If an instruction reads 'Type "BASALT"', then type the letters between the quotation marks exactly as shown. If the instruction reads 'Strike "T"', then the user should type the indicated key once. RETURN is used to indicate the carriage return key and ESC is used to indicate the "Esc" key (normally in the upper left of the keyboard).

LESSON 1: GETTING STARTED

Perform the following steps to get DATASAVE running on the computer.

1. Turn on the computer.
2. Place the DATASAVE program disk in drive A.
3. Place Disk 2 (SPLICE/DATA) in drive B.
4. With the prompt "A>" on the screen, type "MENU".
5. A title screen (with a logo of an electron microprobe) will be displayed.
6. Strike any key to continue.
7. The screen should clear and then appear as follows.

```
*** D A T A S A V E   M E N U ***

          E     Edit
          P     Plot
          L     List
          N     Norm
          B     Blend
          T     Title
        ESC     Exit

          Choose one:
```

6. To select an option, strike the first letter of the option.

LESSON 2: INTRODUCTION TO DATAPLOT

This lesson introduces the new user to producing graphs with DATAPLOT templates. Using data and templates supplied on Disk 2, the user will create a K2O vs. SiO2 orthogonal plot and an AFM ternary diagram. (These templates were created for the CGA standard and will produce 640x200 pixel graphs on both CGA- and EGA-based systems.)

The first step is to load DATAPLOT.

1. Type "P" while the DATASAVE menu is displayed.

   The DATAPLOT menu will be displayed.

The next few steps select the sample-code file B:CODES for use in this run.

2. Strike "S".

   The program asks, "Sample-Code File?".

3. Type "CODES" followed by a RETURN.

   The program asks, "Drive (A or B)?".

4. Type "B" followed by a RETURN.

   The DATAPLOT menu will be displayed. Note that the sample-code file is now B:CODES. Also note that the plot type is orthogonal.

The first plot to be created will be a graph of K2O vs. SiO2 using a template and a data file stored on Disk 2.

5. Strike "T" to create a plot using a template.

   The program asks, "Template Name?".

6. Type "KSI" followed by a RETURN.

   The program asks, "Drive (A or B)?"

7. Type "B" followed by a RETURN.

   The subroutine B:KSI is recalled to create a graph of K2O vs. SiO2.

   The program asks, "Data File Name?".

8. Type "BASALTS" followed by a RETURN.

   The program asks, "Drive (A or B)?".

9. Type "B" followed by a RETURN.

   DATAPLOT recalls a data file titled B:BASALTS.

   A summary of the file contents is displayed.

   The program asks, "Use this file (Y/N)?".

10. Strike "Y". (Yes, use this file.)

    The program asks, "List values (Y/N)?".

11. Strike "N". (No, do not list values.)

    The program asks, "Plot all data (Y/N)?".

12. Strike "Y". (Yes, plot all the data.)

13. The word "Dot" appears in the lower left corner. Strike the "+" or "-" keys until "Square" appears in the lower left corner.

14. Strike RETURN. The data will be plotted as squares.

    Numbers will flash by on the screen as the plot is calculated.

    The program asks, "Recall more data (Y/N)?".

15. Strike "N". (No, do not recall more data.)

    A graph of K2O vs. SiO2 is displayed on the screen.

The next steps illustrate how DATAPLOT can identify data points and review data.

16. Strike the "+" key.

    A cross-shaped cursor appears in the upper left corner of the screen.

17. Press the cursor-control keys (the numeric key pad on most keyboards) or move the mouse until the cursor moves across the screen. Experiment with all of the cursor control keys. Strike "F" to speed up the movement of the cursor. Strike "S" to slow down the movement of the cursor.

18. Strike the SPACE BAR.

    The cursor becomes a dot.

19. Strike the SPACE BAR again.

    The cursor becomes a square.

20. Use the cursor-control keys to move the cursor (now a square) over one of the data points. The data point will be eclipsed.

21. Strike "X".

    The data point flashes and the sample name is displayed in the lower left corner.

22. Strike "V".

    The composition of the sample appears in the upper left corner.

23. Use the " ↑ ↓ " keys to scroll through the complete composition of the sample.

24. Strike ESC.

    The analysis of the sample disappears.

25. Strike ESC.

    The cursor and selected data point vanish (they'll be back).

26. Strike "N"

    The program asks, "Sample?".

27. Type "HAW10" followed by a RETURN.

28. The square corresponding to sample HAW10 flashes.

29. Strike any key to continue.

30. Strike "V".

    The composition of HAW10 appears in the upper left corner.

31. Strike ESC.

    The composition of HAW10 disappears.

32. Strike ESC twice.

33. Strike "Q".

    The program asks, "Clear screen (Y/N)?".

34. Strike "Y". (Yes, clear screen.)

    The DATAPLOT menu is displayed

The following steps will switch DATAPLOT into "ternary mode" and create an AFM diagram.

35. Strike "P" until the plot type is Ternary.

36. Strike "T" to create a plot using a template.

    The program asks, "Template Name?".

37. Type "AFM" followed by a RETURN.

    The program asks, "Drive (A or B)?"

38. Type "B" followed by a RETURN.

    The program asks, "Data file name?".

39. Strike RETURN twice to use B:BASALTS, which is already in memory.

    A summary of the file contents is displayed.

    The program asks, "Use this file (Y/N)?".

40. Strike "Y". (Yes, use this file.)

    The program asks, "List values (Y/N)?".

41. Strike "N". (No, do not list values.)

    The program asks, "Plot all data (Y/N)?".

42. Strike "Y". (Yes, plot all the data.)

    The current symbol appears in the lower left corner.

43. Select another symbol if desired. Strike RETURN to lock in the desired symbol.

    Numbers will flash by on the screen as the plot is calculated. The program asks, "Recall more data (Y/N)?".

44. Strike "N". (No, do not recall more data.)

    The screen should display an AFM diagram.

The following steps illustrate how a diagram may be labelled.

45. Strike "L".

    The program asks, "Label?".

46. Type "HAWAIIAN BASALTS" followed by RETURN.

47. Use the cursor-control (the numeric key pad on most systems) or the mouse to move the label around the screen. Strike "F" to speed up the motion. Strike "S" to slow down the movement for careful positioning. Note that the label may not be moved off of the screen.

48. Position the label to the left of the ternary and strike "E" to fix its position on the screen.

The graphics cursor is a powerful tool for modifying images. The following steps illustrate some of its potential.

49. Strike "+" to activate the graphics cursor.

    A cross-shaped cursor should appear in the upper left corner.

50. Using the cursor-control, "F" and "S" keys, position the cursor just above and to the left of the "H" in "HAWAIIAN".

51. Strike "B" to initiate a block function.

    The cursor disappears.

52. Press one of the cursor-control keys or move the mouse. The cursor will appear and move.

53. Position the cursor just below and to the right of the "N" in "HAWAIIAN".

54. Strike "M".

55. The cursor disappears and "HAWAIIAN" may be repositioned with the cursor-control keys or mouse.

56. Move "HAWAIIAN" above "BASALTS" and strike "E" to fix "HAWAIIAN" in its new position.

57. Move the cursor just below and left of "BASALTS".

58. Strike "B".

    The cursor disappears again.

59. Press one of the cursor-control keys. The cursor will appear. Move it just above and right of "BASALTS".

60. Strike "C". "BASALTS" disappears.

61. Use the cursor-control keys to make "BASALTS" appear and then move a copy of "BASALTS" to a new position.

62. Strike "E" to fix the position of the new "BASALTS" label.

63. Place the cursor just above and to the right of the new "BASALTS" label.

64. Strike "B". The cursor disappears.

65. Place the cursor just below and to the left of the new "BASALTS" label.

66. Strike "E". The second "BASALTS" label is erased.

67. Strike "L".

68. Move the cursor about 2 or 3 cm and strike "L" again.

    A line has just been drawn on the screen.

69. If a Mouse Systems mouse is attached, depress the left button and move the mouse in an arc.

    A curved line has just been drawn on the screen.

70. Strike ESC.

71. Strike "Q".

    The program asks, "Clear screen (Y/N)?".

72. Strike "Y".

    The DATAPLOT menu is displayed.

At this point the user may wish to examine more plots. Templates for Ti-Zr orthogonal plots (B:TIZR) and Ti-ZR-Y ternary plots (B:TIZRY) are provided on Disk 2. Be sure that the appropriate plot type is displayed on the DATAPLOT menu and strike "T" to create a plot with a template. Templates in enhanced graphics are supplied for a SiO2 histogram (B:SIO2HIST) and rare earth element plots (B:REEPLOT). To use these you must have an enhanced graphics adapter.

The final step in this lesson returns the user to the DATASAVE main menu.

72. Strike ESC.

    The DATASAVE main menu should be displayed.

LESSON 3: CREATING A GRAPH FROM SCRATCH

This lesson illustrates the creation of a graph "from scratch". The user will create a subroutine for a plot of TiO2 vs. FeO/MgO.

The first step is to load DATAPLOT.

1. Type "P" while the DATASAVE menu is displayed.

   The DATAPLOT menu will be displayed.

Examine the menu; if the current sample-code file is B:CODES proceed to step 2. If it is not, then perform steps 2, 3 and 4 from Lesson 2.

A subroutine must be created to tell the computer what to plot.

2. Strike "E" to edit a subroutine.

   The program displays a submenu.

3. Strike "N" to create a new subroutine.

   The Subroutine Editor is displayed.

4. Strike "D". A small "d" appears on line 1.
5. Type "TIO2" followed by a RETURN.

   The arrow advances one line.

6. Strike "O". A small "o" appears on line 2.
7. Type "=Y" followed by a RETURN. The arrow advances.
8. Strike "D". A small "d" appears on line 3.
9. Type "FEO" followed by a RETURN. The arrow advances.
10. Strike "D". A small "d" appears on line 4.
11. Type "MGO" followed by a RETURN. The arrow advances.
12. Strike "O". A small "o" appears on line 5.
13. Type "/" followed by a RETURN. The arrow advances.

14. Strike "O". A small "o" appears on line 6.

15. Type "=X" followed by a RETURN. The arrow advances.

If instructions were followed to the letter, the screen should look as follows.

```
SUBROUTINE EDITOR

     1   d TIO2
     2   o =Y
     3   d FEO
     4   d MGO
     5   o /
     6   o =X
     7  >
     8
     .
     .
```

If the screen does look like this, the user has just written a STACK subroutine (see Sections 5 and 6).

16. Strike ESC.

    The program asks, "Subroutine name?".

17. Type "TIFM" followed by a RETURN.

    The program asks, "Drive (A or B)?"

18. Type "B" followed by a RETURN.

    The program stores the subroutine as B:TIFM.SUB and presents a menu of options.

19. Strike ESC.

    The DATAPLOT menu is displayed.

The next steps will use the above subroutine to create a plot.

20. Strike "C".

    The program asks, "Subroutine name?".

21. The current subroutine (B:TIFM) will be used so strike RETURN twice.

    The program asks, "Data file name?".

22. Type "BASALTS" followed by a RETURN.

    The program asks, "Drive (A or B)?".

23. Type "B" followed by a RETURN.

    A summary of the file contents is displayed.

    The program asks, "Use this file (Y/N)?".

24. Strike "Y". (Yes, use this file.)

    The program asks, "List values (Y/N)?".

25. Strike "N". (No, do not list values.)

    The program asks, "Plot all data (Y/N)?".

26. Strike "Y". (Yes, plot all the data.)

    On CGA systems, the current symbol appears in the lower left corner.

    On EGA systems, the current color appears in the lower left corner.

27. Use the "+" and "-" keys to select the desired color and symbol and strike RETURN to lock them in.

    The program asks, "Recall more data (Y/N)?".

28. Strike "N". (No, do not recall more data.)

    The program displays upper and lower limits for each axis.

    The program asks, "Input lowest X Coordinate?".

29. Type "0" followed by a RETURN.

    The program asks, "highest X Coordinate?".

30. Type "2" followed by a RETURN.

    The program asks, "Input lowest Y Coordinate?".

31. Type "0" followed by a RETURN.

    The program asks, "highest Y Coordinate?".

32. Type "4" followed by a RETURN.

    The program asks, "Tick-mark spacing for X-axis?".

33. Type ".2" followed by a RETURN.

    The program asks, "Tick-mark spacing for Y-axis?".

34. Type ".2" followed by a RETURN.

    The graph is displayed.

DATAPLOT allows graph size, shape and position to be modified. The following steps illustrate this potential.

36. Strike "X". The program asks the following question.

    Change X-axis length by what percentage (+ or -)?

37. Type "-30" followed by a RETURN.

    The graph is redrawn.

38. Strike "Y". The program asks the following question.

    Change X-axis length by what percentage (+ or -)?

39. Type "-20" followed by a RETURN.

    The graph is redrawn.

40. Strike "M".

    The program asks, "Horizontal shift in pixels?".

41. Type "50" followed by a RETURN.

    The program asks, "Vertical shift in pixils?".

42. Type "-20" followed by a RETURN.

    The graph is redrawn in a new position.

43. Strike "L".

    The program asks, "Label?".

44. Type "TiO2 vs. FeO/MgO" followed by a RETURN. The lower case letters may be generated by depressing the "CAPS" key while striking the letters.

45. Position the Label in the upper left corner of the graph and strike "E".

The following steps will create a template using this graph.

46. Strike "S".

    The program asks, "File name?".

47. Type "TIFM" followed by a RETURN.

    The program asks, "Drive?".

48. Type "B" followed by a RETURN.

    The program displays the following choice.

    (1) Store Axes + Points (2) Store Axes Only

49. Strike "2".

    The data is stripped from the graph.

    The image of the empty graph is stored on disk. If enhanced graphics are being used, this could take up to 90 seconds.

    The program asks, "Store axis definitions (Y/N)?".

50. Strike "Y". (The answer must be "Y" to create a template.)

    The program returns to the DATAPLOT menu.

The user may now create a TiO2 vs. FeO/MgO plot using the new template.

51. Strike ESC.

    The DATASAVE main menu is displayed.

LESSON 4: CALCULATING NORMS

This lesson illustrates the use of DATANORM to perform and store the results of normative mineral calculations.

1. Strike "N" when the DATASAVE main menu is displayed.

   DATANORM is loaded and the C.S. Hutchison is referenced.

2. Strike any key.

   The following menu is displayed.

```
****************** MENU ****************

          S    Sample-Code File

          D    Directory Info

          I    Inkey Analysis

          R    Recall Data File

        Esc    Exit Program

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

     Current Sample-Code File: B:CODES
```

3. If the current sample-code file is not B:CODES, change it following steps 2 through 4 in Lesson 2.

The following steps will recall the file B:BASALTS and perform a norm calculation on a selected analysis.

4. Strike "R".

   The program asks, "Data file name?".

5. Type "BASALTS" followed by a RETURN.

   The program asks, "Drive (A or B)?".

6. Type "B" followed by a RETURN.

   The program recalls B:BASALTS, presents a summary and asks the following question.

   Use this file (Y/N)?

7. Strike "Y".

   The program asks, "Run all the data (Y/N)?".

8. Strike "N".

   A listing of the data is given by entry number in the file. "WR+" is the sample code. Sample names are in the last column.

9. Move the arrow to entry 10 (HAW10) using the cursor control keys " ↑ ↓ ".

10. When the arrow points at entry 10, Strike "X".

    The computer beeps.

11. Strike ESC.

    A summary of sample HAW10's composition is displayed.

    The program asks, "Modify input (Y/N)?".

12. Strike "Y".

    The program asks, "Element/Oxide?",

13. Type "CO2" followed by a RETURN.

    The program asks, "Weight Percent?".

14. Type "1.0" followed by a RETURN.

    The program asks "Element/Oxide?".

15. Strike RETURN.

    The modified analysis is displayed.

    The program asks, "Modify input (Y/N)?".

16. Strike "N".

    The program reports Fe++ / total Fe and asks,

    "Do you want to change it (Y/N)?".

17. Strike "N".

    The program asks, "Does the rock contain calcite (Y/N)?".

18. Strike "Y".

CIPW and NIGGLI norms are displayed.

19. Strike the cursor control keys " ↑ ↓ ".

    The norms are scrolled on the screen.

20. Turn on an attached printer and strike "P".

    The norm is reproduced on the printer.

21. Strike "ESC".

    The DATANORM menu is displayed.

DATANORM can batch process entire files and send the results to a printer and(or) disk. The following steps will create a new file named B:BASALTS.NRM that contains a CIPW norm for each sample in B:BASALTS. B:BASALTS.NRM will be used in the next Lesson 5.

22. Strike "R".

    The program will ask for the data file name and drive.

23. Strike RETURN twice to continue using B:BASALTS.

    The program recalls B:BASALTS, presents a summary and asks the following question.

    Use this file (Y/N)?

24. Strike "Y".

    The program asks, "Run all the data (Y/N)?".

25. Strike "Y".

    The program displays the number of analyses being batch processed and asks,

    "Do you want to fix Fe++/total Fe (Y/N)?".

26. Strike "N". (No, these analyses are OK as is.)

    The program asks, "Do you want normative calcite (Y/N)?".

27. Strike "N". (No, there's no analyzed CO2 anyway.)

    The program asks,

    "Do you want normative cancrinite (Y/N)?"

28. Strike "N".

    The program asks, "Do you want a print out (Y/N)?".

29. Strike "N". (No, not this time.)

    The program asks, "Do you want to store on disk (Y/N)?".

30. Strike "Y".

    The program asks, "Do you want to store CIPW (Y/N)?".

31 Strike "Y". (Yes. The only other choice is NIGGLI.)

    The norms are calculated.

    The program asks "Norm target file?".

32. Strike RETURN twice to use B:BASALTS.NRM.

    The program returns to the DATANORM menu.

33. Strike ESC.

    The DATASAVE main menu is displayed.

LESSON 5: LISTING A FILE

This lesson introduces the use of DATALIST to print a DATASAVE data file. The lessons assumes that the attached printer in an Epson FX, Epson Mx or IBM dot-matrix printer.

1. Strike "L" when the DATASAVE main menu is displayed.

   The DATALIST main menu is displayed.

2. Move the arrow with the ↑ ↓ keys so that it points at "Sample-Code File"

3. Strike RETURN.
   The program asks, "Sample-Code File?".

4. Type "CODES" followed by a RETURN.

   The program asks, "Drive (A or B)?".

5. Type "B" followed by a RETURN.

   The DATALIST main menu is displayed again.

6. Be sure that the printer is on. Move the arrow to to "Printer-Control Code" and strike RETURN twice.

7. Move the arrow to "Data File" and strike RETURN.

   The program asks, "Data file name?".

8. Type "BASALTS" followed by a RETURN.

   The program asks, "Drive (A or B)?".

9. Type "B" followed by a RETURN.

   The DATALIST menu is displayed.

10. Move the arrow to "PREVIEW" and strike RETURN.

    The file B:BASALTS is read and a summary is displayed.

11. Strike RETURN to continue.

    The program asks, "List standard deviations (Y/N)?".

12. Strike "N". (No, there aren't any for this file.)

    The program displays 2 lines on the screen.

13. Hold down the SPACE BAR and the data will be output to the screen. Release the SPACE BAR and the screen freezes.

14. After viewing a few analyses, strike ESC.

    The DATALIST main menu is displayed.

15. Move the arrow to "SEND TO PRINTER".

16. Strike RETURN.

    A summary of the file is displayed.

17. Strike RETURN to continue.

    The program asks, "List standard deviations (Y/N)?".

18. Strike "N". (No, there aren't any for this file.)

    The contents of the file are printed.

    When the file is printed, the DATALIST main menu is displayed.

DATALIST allows tables to be written to disk. Any commercial word-processing program that uses ASCII files can be used to edit these tables and merge them into documents.

The following steps store a table on disk using the data file B:BASALTS.

19. Move the arrow to "SEND TO DISK".

20. Strike RETURN.

    The program asks, "Target file?".

21. Type "BASALTS.TAB" followed by a RETURN.

    The program asks, "Drive (A or B)?".

22. Type "B" followed by a RETURN.

    A summary of the file is displayed.

23. Strike RETURN to write the table to disk.

    The program asks, "List standard deviations (Y/N)?".

24. Strike "N".

    The table is written to disk.

    The DATALIST main menu is displayed.

To exit DATALIST, perform the following steps.

25. Move the arrow to "EXIT PROGRAM".
26. Strike RETURN.

LESSON 6: SAMPLE-CODE FILES

It is essential that the prospective user learn to create and edit sample-code files. This lesson will introduce the user to creating a sample-code file from scratch.

1. With the DATASAVE MENU showing, type "E".

   DATAEDIT will be recalled off the default drive.

   The DATAEDIT main menu will be displayed.

To create a new sample-code file, simply select one for editing that does not already exist. This is performed by the following steps.

2. With the DATAEDIT menu displayed, type "S".

   The program asks "Sample-Code File?".

3. Type "TESTFILE" followed by a RETURN.

   The program asks "Drive (A or B)?".

4. Type "B" followed by a RETURN.

   The DATAEDIT menu is displayed on the screen.

   The current sample-code file is now B:TESTFILE.

5. Strike "M".

6. The program responds as follows.

```
B:TESTFILE was not found!

R    Return to Main Menu
C    Create new sample-code file
```

7. Strike "C". The program responds as follows.

```
There are no entries in the sample-code file!

Begin a new file by choosing option A!
```

8. Type "A" to select option A. The program responds as follows.

   Entry number 1

   Mineral Code?

As an example, the following steps will create a code for storing a record the grade and annual income by year for several, fictitious, poverty-stricken employees of the U.S. Geological Survey.

9. Type "INC" followed by a RETURN.

   A column of numbers and some instructions will appear on the screen.

   An arrow points to line 1.

10. Type "!GRD" (an abbreviation for grade) followed by a RETURN.

    The arrow advances to line 2.

11. Type "!PAY" followed by a RETURN.

    The arrow advances to line 3.

12. Strike ESC.

    A new code, "INC" has just been created.

13. Strike ESC.

    The program responds as follows.

    The sample-code file has been modified!

    Assign a new name (Y/N)?

14. Type "N".

    The new sample-code file is stored.

    The DATAEDIT main menu is displayed.

This code will be used in the next lesson to introduce data entry and editing.

15. Strike ESC.

    The DATASAVE main menu is displayed.

LESSON 7: ENTERING NEW DATA

The following steps will introduce the new user to entering new data and storing it on disk.

1. With the DATASAVE MENU showing, type "E".

   The DATAEDIT main menu will be displayed.

2. Check the current sample-code file. If it is B:NEWFILE, go on to step 3. Otherwise, change it to B:NEWFILE by striking "S" (see steps 2 to 4 in Lesson 2).

3. Strike "I".

   The program asks "Data File?".

4. Type "SALARIES" followed by a RETURN.

   The program asks "Drive (A or B)?".

5. Type "B" followed by a RETURN.

   A summary of B:SALARIES is reported.

6. Strike RETURN to continue.

   The program asks "Input standard deviations (Y/N)?".

7. Strike "N".

   The program responds with the following.

   Sample Number 1 Sample Name?

8. Type "JOHN SMITH" followed by a RETURN

   The program asks "Sample Code?".

9. Type "INC" followed by a RETURN.

   The program displays the following.

```
1.  GRD    val:>          <
2.  PAY    val:
```

10. Type "12" followed by a RETURN.

    The arrows advance to PAY.

11. Type "36850" followed by a RETURN.

    The computer beeps because all entries are now filled.

12. Strike ESC.

    The program asks, "Done with data entry (Y/N)?".

13. Strike "N".

    The program asks, "Back up data (Y/N)?"

14. Strike "N".

    The program responds with the following.

    Sample Number 2 Sample Name?

15. Type "JANE DOE" followed by a RETURN.

    The program asks, "Sample Code?".

16. The same code will be used so strike RETURN.

    The program displays the following.

```
1.  GRD    val:>          <
2.  PAY    val:
```

17. Type "13" followed by a RETURN.

18. Type "42100" but do not strike RETURN.

At this point, assume that 42100 should have been 47100.

19. Strike the <- key multiple times to erase the last four digits of 42100.

20. Type "7100" followed by a RETURN.

Actually Jane is a GS 14 not a 13, so that should be corrected as well.

21. Move the arrows, "> <" by striking PgUp or PgDn while the Ctrl key is depressed.

22. Position the "> <" around the "13".

23. Strike "D".

    The entry for GRD is deleted.

24. Type "14" followed by a RETURN.
25. Strike ESC.

    The program asks, "Done with data entry (Y/N)?".
26. Strike "Y".

    The file is stored.

    The DATAEDIT main menu is displayed.

LESSON 8: EDITING DATA

DATAEDIT provides many tools for editing individual samples and entire files. The following steps illustrate some of the editing potential by modifying one sample in the file B:BASALTS. A description of DATAEDIT's capabilities is presented in detail in Section 4.

1. With the DATASAVE main menu showing, type "E".

   The DATAEDIT main menu will be displayed.

2. Check the current sample-code file. If it is B:CODES, go on to step 3. Otherwise, change it to B:CODES by striking "S" (see steps 2 to 4 in Lesson 2).

3. Strike "E".

   The DATAEDIT editing menu is displayed.

4. Strike "E".

   The program asks, "Data File?".

5. Type "BASALTS" followed by a RETURN.

   The program asks, "Drive (A or B)?".

6. Type "B" followed by a RETURN.

   A summary of B:BASALTS is presented.

7. Strike RETURN to continue.

   Three columns of information are displayed.

   An arrow points to the top line.

The next steps will edit the data for sample HAW10.

8. Move the arrow with the cursor-control keys, " ↑ ↓ ".

9. Position the arrow so that it points at a line that reads

   "10 WR+ HAW10".

10. Strike "X".

    The data for HAW10 is displayed on the screen.

    The arrows, "> <" point to the entry for NA2O.

11. Move the arrows by depressing the Ctrl key and striking PgUP, PgDn, Home, End, <- or ->.

12. Position the "> <" at the entry for H2O.

13. Strike "D".

    The entry for H2O (which was 0) is deleted.

14. Type "1.0" followed by RETURN.

    The arrows advance to the right hand column (standard deviations).

15. Strike RETURN.

    The sum of the major elements is modified (see box).

16. Depress the Alt key and strike F1.

    The program asks, "New Name?".

17. Type "TEST" followed by a RETURN.

    TEST replaces HAW10 on the screen.

18. Strike ESC.

    The data for HAW10 (now TEST) disappears.

    At this point another analysis could be selected for editing.

19. Strike ESC.

    The program presents a series of options for storing or aborting the editing file.

20. Strike "A" this time. (Abort and do not save -- its better not to mess up a perfectly good file.)

    The DATAEDIT editing menu is displayed.

21. Strike ESC.

    The DATAEDIT main menu is displayed.

As a final exercise with DATAEDIT, the following steps will produce an average of the 25 analyses in the file B:BASALTS.

22. Strike "A".

    The program asks, "Data File?".

23. B:BASALTS is already in memory so strike RETURN twice.

    A summary of B:BASALTS is presented.

    The program asks, "Use this file (Y/N)?".

24. Strike "Y".

    The program asks, "Use all the data (Y/N)?".

25. Strike "Y".

    The program reports the number of sample that will be used in the calculation.

26. Strike any key to continue.

    An average is calculated. Standard deviations are expressed as a percentage of the abundance.

    The program asks, "Sample name?".

27. Type "BASALT AVE" followed by a RETURN.

    The program asks, "Print a hardcopy (Y/N)?".

28. Be sure the printer is on. Strike "Y".

    The program asks, "Store an disk (Y/N)?".

29. Strike "N". ("Y" appends BASALT AVE to the current file.)

    The program asks, "Do more calculations (Y/N)?".

30. Strike "N".

    The DATAEDIT main menu is displayed.

31. Strike ESC.

    The DATASAVE main menu is displayed.

LESSON 9: BLENDING DATA

The following steps will combine the oxide abundances from file B:BASALTS with the normative mineral abundances from B:BASALTS.NRM. This lesson presumes that Lesson 4 was performed successfully and that file B:BASALTS.NRM was created.

1. With the DATASAVE main menu showing, type "B".

   The BLEND menu is displayed.

2. If the current sample-code file is B:CODES go to step 3. Otherwise change the current sample-code file to B:CODES by performing steps 2 through 4 in Lesson 2.

3. Strike "B".

   The program asks, "Name of target file?".

4. Type "BTEST" followed by a RETURN.

   The program asks, "Drive (A or B)?".

5. Type "B" followed by a RETURN.

   The program asks, "Name of first source file?".

6. Type "BASALTS" followed by a RETURN.

   The program asks, "Drive (A or B)?".

7. Type "B" followed by a RETURN.

   The program asks, "Name of second source file?".

8. Type "BASALTS.NRM" followed by a RETURN.

   The program asks, "Drive (A or B)?".

9. Type "B" followed by a RETURN.

   The codes in each file are itemized.

   The program asks, "Target code?".

10. Type "WRC" followed by a RETURN.

    The program asks, "First source code?".

11. Type "WR+" followed by a RETURN.

    The program asks, "Second source code?".

12. Type "CPW" followed by a RETURN.

    A summary of the three codes is presented.

    An arrow points to the first line in the first column.

    The program asks, "OK (Y/N)?".

13. Strike "Y".

14. Strike "X".

    Small squares appear beside NA2O.

15. Strike "↓".

    The arrow on the left side of the screen moves down one line.

16. Alternate between striking "X" and "↓" until squares are displayed beside the oxides from Na2O to CO2.

17. Strike "↓" again.

    The arrow should be pointing at "!Q".

18. Strike "S".

    The source file changes to 2.

19. Alternate between striking "X" and "↓" until squares are displayed beside all of the entries in the first two columns.

20. Strike ESC.

    The program asks, "OK (Y/N)?".

21. Strike "Y".

    The program asks for the characters to be used in matching sample names.

    All of the samples in B:BASALTS have names like "HAW##" where ## is a 1 or 2 digit number.

    The program asks, "Source File 1: First character".

22. Strike "1" followed by a RETURN.

    The program asks, "Last character".

23. Strike "5" followed by a RETURN.

    The program asks, "Source File 2: First character".

24. Strike "1" followed by a RETURN.

    The program asks, "Last character".

25. Strike "5" followed by a RETURN.

    The program asks, "Setup satisfactory (Y/N)?".

26. Strike "Y".

    The data are BLENDed and the target file stored.

    The program returns to the BLEND menu.

27. Strike ESC.

    The program displays the DATASAVE main menu.

At this point the user may wish to inspect the target file B:BTEST using either DATAEDIT or DATALIST.

S E C T I O N 4

D A T A E D I T

Contents

INTRODUCTION

DATAEDIT is a collection of data- and file-handling options that allow a user to create and maintain a customized database. The available options are listed below and described in detail in subsequent sections.

DOS Options:

Directory Information	Display directory of any drive.

Data File Options:

Average data	Calculate and store averages and standard deviations for data.
Batch codes	Change the sample codes of an entire file.
Cull a file	Select samples from one or more files to be placed in a new file.
Edit a file	Edit entire contents of a file: delete samples and change sample names, codes, and entries.
Input new data	Enter new data from the keyboard.
Merge files	Combine two or more files into one file.

Sample-Code File Options:

Modify sample codes	Edit sample-code file: add new sample codes; edit or delete old sample codes; merge or cull sample-code files.
Select sample-code file	Change current sample-code file.

MENUS

DATAEDIT utilities are accessed from two menus. A "main menu" is displayed when DATAEDIT is loaded. Normal program operation will always return to the main menu. The main menu appears as follows.

```
*** DATAEDIT ***

 D    Directory Information

 I    Input new data

 E    Edit data

 S    Select sample code file

 M    Modify sample codes

 A    Average data

ESC   Exit program

Current Sample Code File: A:CODES
```

An "editing menu" is displayed when the user selects "Edit data" from the main menu by striking "E". The editing menu appears as follows.

```
* MENU *

 M     Merge files
 E     Edit a file
 C     Cull a file
 B     Batch codes
ESC    Main Menu
```

Both menus operate in the same manner. An option is selected by striking a single key. Do not follow the key stroke with a RETURN. Only the listed keys are live; other key strokes will produce no response.

DIRECTORY INFORMATION

DESCRIPTION

This option allows the user to inspect the directory of any drive.

OPERATION

1) Strike "D" while the main menu is displayed.

2) The screen will clear and the following question will appear at the bottom of the screen.

```
Directory Information          Drive (A or B)?
```

3) Key in the letter of the desired drive. Any existing drive (A, B, C, etc.) may be used. Do not use a colon.

4) Strike RETURN.

5) The program will display the requested directory.

6) Strike any key to return to the main menu.

AVERAGING DATA

DESCRIPTION

This option allows the user to calculate an average sample "composition". The calculation may be based on all samples in a data file or on individually selected samples. All the samples used in an average calculation must have the same sample code. The average calculation determines standard deviations expressed as percentages of the stated abundance. The results may be dumped to a printer and/or appended to the current data file for storage on disk.

OPERATION

1) Strike "A" while the main menu is displayed.

2) The screen will clear and the following question will appear.

```
Data File?
```

3) Type the name of the data file (e.g. DUMMY.ABC) and strike RETURN. Striking RETURN without typing a name will cause the program to use the name of the most recently used data file. The program will then ask for the drive location of the data file.

```
Data File? DUMMY.ABC     Drive (A or B)?
```

Note: type "DIR" for file name to display the directory of the drive indicated in the next step (step 4).

4) Type the letter of the drive (A, B, C, etc.) followed by a RETURN.

Typing RETURN without a letter will cause the program to use the drive of the most recently used data file. Typing two or more letters (e.g. "XX") followed by a RETURN will return the program to the main menu.

5) The program reads the file from disk.

6) The screen reports a summary of its contents. The display will look similar to the following.

B:DUMMY.ABC

There are 30 entries in the file.

Sample Code	# of entries
WRN	18
WR1	9
WRT	3

****

Use this file (Y/N)?

7) Type "N" to return to step 2 and select another file.
Type "Y" to proceed.

8) If all the samples have the same sample code, the program asks:

Use all the data (Y/N)?

9) A "Y" response uses all the data in the file and proceeds to step 17.
A "N" response proceeds to 10.

10) A summary of the file contents is presented in three columns.

The left column displays the entry number of each sample in the file. The middle column displays the sample code of each sample. The right column displays the name of each sample.

An arrow points to one of the samples.

11) To include a sample in the average, move the arrow to it using the cursor control keys, ↑ ↓ . These keys will scroll through the entire file once the arrow has reached the top or bottom of the screen.

12) Strike "X" when the arrow points to the desired sample.

13) The first 20 entries for the sample will be displayed on the right side of the screen.

14) Strike "X" to include the sample in the average.
Strike "P" to avoid including the indicated sample in the average.

15) Continue to select samples for the average calculation.

16) Strike ESC when selection is complete.

17) The screen clears and prints a message similar to:

Your are batch processing 10 samples.

Hit any key to continue!

18) Strike any key.

19) The average values and standard deviations are displayed on the screen.

20) The program asks:

Sample name?

21) Input any name from zero to twelve characters long and strike RETURN.

22) The program asks:

Print a hard copy (Y/N)?

23) A "Y" response prints the results on an attached printer. Be certain that the printer is on before striking "Y".
A "N" response proceeds without printing.

24) The program asks:

Store on disk (Y/N)?

25) A "N" response proceeds to step 26.
A "Y" response appends the average to the back of the current file and displays a screen similar to the following.

Source File: B:DUMMY.ABC

Same drive and name
I Input new drive and name
A Abort -- do not save

Your Choice:

The source file (e.g. B:DUMMY.ABC) will be the drive and name of the file recalled for averaging.

Striking RETURN will write the modified data file over the source file -- THE OLD FILE WILL BE DESTROYED!

Striking "I" will allow the user to type in a new name and drive for storing the edited data.

Striking "A" will return to the editing menu without storing the data.

26) The program asks:

Do more calculations (Y/N)?

27) A "N" response returns to the program to the main menu.
A "Y" response proceeds to step 28.

28) The program asks:

Use the same file (Y/N)?

29) A "N" response returns to step 2.
A "Y" response returns to step 6.

BATCH CHANGING CODES

DESCRIPTION

This option allows the user to load an existing data file from disk into an editing file in memory and change its sample codes. This results in reformatting the data.

The program prompts for the original code and the desired code to replace it. All samples with the indicated original code are modified. The original code is replaced and the order of the entries is changed to be compatible with the new code. The modified bile may be stored on disk.

Operation

1) Strike "E" while the main menu is displayed.

2) The screen will display the editing menu.

3) Strike "B" while the editing menu is displayed.

4) The screen will clear and the following question will appear.

```
Data File?
```

5) Type the name of the data file (e.g. DUMMY.ABC) and strike RETURN. Striking RETURN without typing a name will cause the program to use the name of the most recently used data file. The program will then ask for the drive location of the data file.

```
Data File? DUMMY.ABC     Drive (A or B)?
```

Note: type "DIR" for file name to display the directory of the drive indicated in the next step (step 6).

6) Type the letter of the drive (A, B, C, etc.) followed by a RETURN.

Typing RETURN without a letter will cause the program to use the drive of the most recently used data file. Typing two or more letters (e.g. "XX") followed by a RETURN will return the program to the main menu.

7) The program reads the file from disk.

8) The screen reports a summary of its contents. The display will look similar to the following.

```
                    B:DUMMY.ABC

              There are 30 entries in the file.

          Sample Code       # of entries

              WRN                18
              WR1                 9
              WRT                 3

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

                  Use this file (Y/N)?
```

9) Type "N" to return to step 4 and select another file. Type "Y" to proceed.

10) The program asks:

```
                    Original code?
```

11) Type one of the sample codes shown on the screen followed by a RETURN.

12) The program asks:

```
                      Change to?
```

13) Type the sample code that is desired for the reformatted data followed by a RETURN. The code must be one to three characters and present in the current sample-code file.

14) The screen will clear and show the number of the current sample as the program works through the data.

15) The following options will be displayed when the program has finished changing codes.

```
                Source File: B:DUMMY.ABC

                  Same drive and name
              I   Input new drive and name
              A   Abort -- do not save


                     Your Choice:
```

The source file (e.g. B:DUMMY.ABC) will be the drive and name of the file recalled for changing codes.

Striking RETURN will write the edited data over the source file -- THE OLD FILE WILL BE DESTROYED!

Striking "I" will allow the user to type in a new name and drive for storing the edited data.

Striking "A" will return to the editing menu without storing the data.

16) The program returns to the editing menu.

CULLING A FILE

DESCRIPTION

The cull option provides a mechanism for combining selected samples from more than one file into a single file. This option loads a data file from disk into an editing file in memory. Once in memory, the contents of the file may be examined, samples may be "culled" (selected) for entry into a separate target file in memory. More than one file may be "culled" to build up the target file. Samples may be culled on the basis of sample code, sample name or individual samples.

Culling by sample code copies all samples with the specified code from the editing file to the target file.

Culling by sample name copies all samples with the indicated name from the editing file to the target file. If a "name" that is n characters long is indicated, all samples are culled that have sample names beginning with the same n characters. For example, indicating a number of 123 would cull samples with sample numbers "123", "123ABC", and "123 201" because they begin with "123".

If the user opts to cull samples individually, the screen is cleared and a listing of the entry number, sample code, and sample name for each sample in the editing file is displayed on the left side of the screen. Only 24 samples are displayed at a time but the user may scroll through the entire file. Samples are culled by moving a cursor (arrow) to each sample and striking "X" to select the sample. Movement of the cursor and scrolling of the data is accomplished with the cursor-control keys as indicated in the menu at the bottom of the screen. The user may cull in two different modes. In the "Report on" mode, striking "X" causes the sample to be printed on the right side of the screen and the user may choose to cull the sample or "punt", which does not cull the sample. In the "Report off" mode, striking "X" automatically culls the indicated sample. The user may toggle between "Report on" and "Report off" with Alt/F10. The user may cease culling individual samples by striking the ESC key.

Each time the program completes a culling operation, it reports the total number of samples currently in the target file and asks the user if the more samples are to be culled from the current editing file. The user may return to the current file as many times as desired or call up as many different files as desired until the target file is full (100 samples) or the culling operation is complete.

Operation

1) Strike "E" while the main menu is displayed.

2) The editing menu will appear.

3) Strike "C" while the editing menu is displayed.

4) The screen will clear and the following question will appear.

```
Data File?
```

5) Type the name of the data file (e.g. DUMMY.ABC) and strike RETURN. Striking RETURN without typing a name will cause the program to use the name of the most recently used data file. The program will then ask for the drive location of the data file.

```
Data File? DUMMY.ABC     Drive (A or B)?
```

Note: type "DIR" for file name to display the directory of the drive indicated in the next step (step 6).

6) Type the letter of the drive (A, B, C, etc.) followed by a RETURN.

Typing RETURN without a letter will cause the program to use the drive of the most recently used data file. Typing two or more letters (e.g. "XX") followed by a RETURN will return the program to the main menu.

7) The program reads the file from disk.

8) The screen reports a summary of its contents. The display will look similar to the following.

```
                    B:DUMMY.ABC

              There are 30 entries in the file.

                Sample Code      # of entries

                    WRN               18
                    FD                 9
                    PX                 3


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

 S   Cull individual samples      C  Cull by sample code

ESC  Abort above                  N  Cull by sample name
```

9) To cull individual samples, proceed to step 10.
   To cull by sample code, skip to step 24.
   To cull by sample name, skip to step 28.

10) Strike "S" to cull individual samples.

11) A summary of the file contents is presented in three columns.

    The left column displays the entry number of the each sample in the file. The middle column displays the sample code of each sample. The right column displays name of each sample.

    An arrow points to one of the samples.

12) To include a sample in the average, move the arrow to it using the cursor control keys, ↑ ↓ . These keys will scroll through the entire file once the arrow has reached the top or bottom of the screen.

13) Strike "X" when the arrow points to the desired sample.

14) The first 20 entries for the sample will be displayed on the right side of the screen.

15) Strike "C" to cull the sample and add it to the target file. Strike "P" to abort the cull of the indicated sample.

16) Continue to select and cull samples.

17) When culling of individual samples is complete strike ESC.

18) The screen will clear and display a message similar to the following.

    Total samples culled = 19

    Cull more samples from this file (Y/N)?

19) A "Y" response returns to step 8. A "N" response proceeds to step 20.

20) The screen will display the following query.

    Cull another file (Y/N)?

21) A "Y" response returns to step 4. A "N" response proceeds to step 22.

22) The screen will clear and present the following options.

```
Source File: B:DUMMY.ABC

   Same drive and name
I  Input new drive and name
A  Abort -- do not save

      Your Choice:
```

The source file (e.g. B:DUMMY.ABC) will be the drive and name of the last file recalled for culling.

Striking RETURN will write the edited data over the source file -- THE OLD FILE WILL BE DESTROYED!

Striking "I" will allow the user to type in a new name and drive for storing the edited data.

Striking "A" will return to the editing menu without storing the data.

23) The program returns to the editing menu.

24) Strike "C" to cull by sample code.

25) The program clears the screen and asks:

Input sample code for selection?

26) Type the code (one to three characters) and strike RETURN.

27) The program culls the samples and returns to step 18.

28) Strike "N" to cull by sample name.

29) The program clears the screen and asks:

Input sample name for selection?

30) Type the sample name (one to 12 characters). A space counts as a character. Any sample that has a name beginning with the characters input at this point will be culled.

31) The program culls the samples and returns to step 18.

EDITING A FILE

DESCRIPTION

This option allows the user to edit the contents of existing data files. Data is recalled from disk and placed in memory where it may be modified. Editing begins by displaying the entry number, sample code, and name of each sample in the file. Only 24 samples are displayed at a time but the user may scroll through the entire file contents. Samples may be selected for editing with a cursor. Directions for cursor control, scrolling data, and editing functions are provided in a menu at the bottom of the screen. A sample may be deleted or displayed on the right side of the screen for editing. Sample names and codes may be changed. Individual entries in each sample may be modified. The edited data may be written over the original source file or stored as a new file.

OPERATION

1) Strike "E" while the main menu is displayed.

2) The editing menu will appear.

3) Strike "E" while the editing menu is displayed.

4) The screen will clear and the following question will appear.

```
Data File?
```

5) Type the name of the data file (e.g. DUMMY.ABC) and strike RETURN. Striking RETURN without typing a name will cause the program to use the name of the most recently used data file. The program will then ask for the drive location of the data file.

```
Data File? DUMMY.ABC     Drive (A or B)?
```

Note: type "DIR" for file name to display the directory of the drive indicated in the next step (step 6).

6) Type the letter of the drive (A, B, C, etc.) followed by a RETURN.

Typing RETURN without a letter will cause the program to use the drive of the most recently used data file. Typing two or more letters (e.g. "XX") followed by a RETURN will return the program to the main menu.

7) The program reads the file from disk.

8) The screen is cleared and a summary of the file contents is presented. The screen should look something like the following.

B:DUMMY.ABC

There are 30 entries in the file.

Sample Code	# of entries
WRN	18
FD	9
PX	3

****

ESC Continue
Abort

9) Strike RETURN to proceed.
Strike ESC to return to the editing menu.

10) A summary of the file contents is presented in three columns.

The left column displays the entry number of the each sample in the file. The middle column displays the sample code of each sample. The right column displays name of each sample.

An arrow points to one of the samples.

11) To include a sample in the average, move the arrow to it using the cursor control keys, ↑ ↓ . These keys will scroll through the entire file once the arrow has reached the top or bottom of the screen.

12) Strike "X" while the arrow points to the desired sample.

13) The entries for the sample will be displayed on the right side of the screen.

14) Edit data using options displayed at the bottom of the screen (see procedures described for editing during input Input New Data, Task 4: Editing Input).

15) When editing of the sample is complete, strike ESC.

16) Select another sample for editing by returning to step 2. Alternatively, proceed to step 17 to quit editing.

17) When editing the file is complete strike ESC to store or abandon data.

18) The screen will clear and present the following options.

```
Source File: B:DUMMY.ABC

  Same drive and name
I Input new drive and name
A Abort -- do not save

Your Choice:
```

The source file (e.g. B:DUMMY.ABC) will be the drive and name of the file called up for editing.

Striking RETURN will write the edited data over the old data -- THE OLD FILE WILL BE DESTROYED!

Striking "I" will allow the user to type in a new name and drive for storing the edited data.

Striking "A" will return to the editing menu without storing the data.

19) The program returns to the editing menu.

INPUT NEW DATA

DESCRIPTION

This option allows the user to enter data from the key board into a file containing up to 100 samples. The user may enter data with or without standard deviations. A screen editor facilitates data entry and allows correction of input errors. Data entry continues until the user has finished or the data file contains 100 samples.

OPERATION

Entering new data breaks down into four tasks, which are performed in the following order.

Task 1: File Selection--Select a file for data input.

Task 2: Sample Identification--Assign a name and sample code to a sample.

Task 3: Data Entry--Key in data entries (weight percents, etc.) for the sample.

Task 4: Editing Input--Correct the entries if necessary.

Task 1: File Selection

1) Strike "I" while the main menu is displayed.

2) The screen will clear and the following question will appear.

```
Data File?
```

3) Type the name of the data file (e.g. DUMMY.ABC) and strike RETURN. Striking RETURN without typing a name will cause the program to use the name of the most recently used data file. The program will then ask for the drive location of the data file.

```
Data File? DUMMY.ABC     Drive (A or B)?
```

Note: type "DIR" for file name to display the directory of the drive indicated in the next step (step 6).

6) Type the letter of the drive (A, B, C, etc.) followed by a RETURN.

Typing RETURN without a letter will cause the program to use the drive of the most recently used data file. Typing two or more letters (e.g. "XX") followed by a RETURN will return the program to the main menu.

7) The program reads the file from disk.

8) The program asks the user:

Input standard deviations (Y/N)?

9) Respond with "Y" to input standard deviations or "N" to ignore standard deviations. This response controls data input until the data entry option has been reentered from the main menu.

10) The program will proceed to sample identification.

Task 2: Sample Identification

1) Sample identification begins by reporting the entry number of the current sample in the file and requesting the name of the sample. The top line of the screen will appear similar to the following.

Entry Number 1 Sample Name?

2) Type the sample name and strike RETURN. The name may be 12 characters long and include numbers, spaces, upper- and lower-case letters, and any character on the keyboard except quotation marks. For example, type ONE.

3) The program requests the sample code. The screen will look similar to the following.

Entry Number 1 Sample Name? ONE Sample Code?

4) Type the sample code followed by a RETURN. The code must be three characters or less and must be present in the current sample-code file. For example, type WRN.

5) The program will display the entry labels assigned to the sample-code. The screen will look something like the following.

```
Entry  Number  1     Sample Name?  ONE      Sample  Code?WRN

                1.  NA2O   wt%:>        < +/-
                2.  MGO    wt%:          +/-
                3.  AL2O3  wt%:          +/-
                .
                .
                .
               20.  ND     ppm:          +/-
```

The exact appearance of the screen will depend on the sample code. The "+/-" will appear only if the user opted to input standard deviations. Only 20 entry labels are displayed at a time. However, the user may scroll through all the entry labels (see Data Editing).

The user may enter new data into sample ONE using the functions described under "Task 3: Data Entry".

Task 3: Data Entry

Data entry is accomplished by entering numbers into "cells". If the user opted to input standard deviations, the cells will be organized into two columns separated by "+/-". The left column of cells is intended for concentrations (weight percent, ppm or ppb) or floating point numbers. The right column is intended for standard deviations. If the user opted to omit standard deviations, only one column for concentrations will be available for data entry.

Data may be entered into the cell indicated by the arrows, "> <". If the cell is blank, it is considered "open" for data input and a number may be entered by typing the number on either the numeric key pad or the number keys at the top of the keyboard. When a number has been completed, striking RETURN will "close" the cell and move the arrows to the next cell in the sequence. If the user opted to input standard deviations, the program will always enter a concentration followed by its standard deviation before moving on to the next concentration. Data cannot be entered into a "closed" cell until it has been reopened

by deleting its contents (see Task 4: Editing Input). A RETURN without typing a number leaves the cell "open" but will be interpreted as a value of zero for the cell.

Note that the numeric key pad is automatically set at this time so that the number lock is on. Never use the Num Lock key -- DATASAVE turns it on and off as needed.

DATASAVE programs place some limits on numerical input from the keyboard. Numbers must be 11 characters or less in length including the decimal point. "Concentrations" must be input as integers or floating point numbers. Exponential notation is not allowed. Absolute values must be greater than or equal to 0.0000001. Integers must be less than or equal to 9999999 and floating point numbers must be less than or equal to 999999.9999. Standard deviations are interpreted by DATASAVE programs to be a percentage of the associated "concentration". Standard deviations must be positive numbers less than 300 (percent).

Data may be entered by the following steps.

1) The editor starts at the top entry and indicates a "cell" for data entry with "> <". Data will be entered from top to bottom.

2) Type an entry using either the numeric key pad or the keys at the top of the key board. Errors may be corrected by striking the "<-" key to erase the last key stroke.

3) Strike RETURN.

4) If the user is entering standard deviations, the arrows move to the right. If standard deviations are omitted, the arrows move down one entry.

5) Type the standard deviation or next entry as appropriate.

   Note: Standard deviations should be input as a percentage of the entry. DATAEDIT will not accept standard deviations larger than 300.

6) Continue typing numerical entries and striking RETURN.

7) When all entries for the sample are complete, press ESC.

8) The program responds with:

Done with data entry (Y/N)?

9) A "Y" response causes the program to store the data on disk and return to the main menu. Storage".
A "N" response causes the program to prompt the user with:

Back up data (Y/N)?

10) A "Y" response cause the entire data file to be backed up on disk before proceeding to the next sample.
A "N" response proceeds directly to the next sample.

Warning: data is not backed up automatically on disk. If data are typed into the memory and not backed up, a power failure, a boot or a "Ctrl/Break", will cause the data to be lost. Therefore, back up the data every 10 or 15 samples.

11) The screen clears and the program returns to Task 2: Sample Identification.

task 4: Editing Input

The data for a sample may be edited whenever it is displayed on the screen. Editing is performed with the following function keys.

Ctrl/Home	Move arrows to top, left cell
Ctrl/End	Move arrows to lower, left cell
Ctrl/PgUp	Move arrows up one cell
Ctrl/PgDn	Move arrows down one cell
Ctrl/->	Move arrows right one cell
Crtl/<-	Move arrows left one cell
D	Delete cell contents
<-	Delete last key stroke
ESC	Done with data entry

The Ctrl/PgUp and Ctrl/PgDn key combinations will scroll through all the entry cells if the arrows are at the top or bottom of the display, respectively.

The user can move the arrows to any cell with the function keys after the last cell has been closed with a RETURN. A cell can be reopened for data entry by moving to arrows to the cell and deleting the contents by striking "D". A new number may be entered into the cell as described in Task 3: Data Entry (steps 2-3).

A sample may be edited as many times as the user wishes. Striking ESC will store the sample in memory as described in Task 3: Data Entry (steps 7-11).

MERGE FILES

DESCRIPTION

This option allows the user to merge the contents of different files up to a total of 100 analyses. Merging recalls a file from disk and appends its entire contents to a target file in memory. The target file is initially empty but grows as files are merged to it. Following each addition to the target file, the program reports the total number of analyses in the target file and asks the user,

Merge another file (Y/N)?

If the answer is "N", the program allows the user to store or abandon the contents of the target file. If the answer is "Y", the program requests the name and drive of another source file to merge. This process can be continued until the target file contains 100 samples at which time the user is prompted to store or abort the target file.

OPERATION

1) Strike "E" while the main menu is displayed.

2) The editing menu will appear.

3) Strike "M" while the editing menu is displayed.

4) The screen displays the following message.

You may recall as many files as desired
up to a total of 100 samples!

Hit any key to continue!

5) Strike any key.

6) The screen will clear and the following question will appear.

Data File?

7) Type the name of the data file (e.g. DUMMY.ABC) and strike RETURN. Striking RETURN without typing a name will cause the program to use the name of the most recently used data file. The program will then ask for the drive location of the data file.

Data File? DUMMY.ABC Drive (A or B)?

Note: type "DIR" for file name to display the directory of the drive indicated in the next step (step 8).

8) Type the letter of the drive (A, B, C, etc.) followed by a RETURN.

   Typing RETURN without a letter will cause the program to use the drive of the most recently used data file. Typing two or more letters (e.g. "XX") followed by a RETURN will return the program to the main menu.

9) The program reads the file from disk.

10) The screen is cleared and a summary of the file contents is presented. The screen should look something like the following.

B:DUMMY.ABC

There are 30 entries in the file.

Sample Code	# of entries
WRN	18
FD	9
PX	3

****

ESC Continue
Abort

11) Type the letter of the drive (A, B, C, etc.) followed by a RETURN. Typing Return without a letter will cause the program to use the drive of the most recently used data file.

12) The program opens the file and reports a summary of its contents.

13) Strike RETURN to proceed. Strike ESC to reject the file and return to step 6.

14) The new data is appended to the target file in memory.

15) If the the target file contains less than 100 samples, the program clears the screen and asks:

Merge another file (Y/N)?

16) A "Y" response returns the program to step 4. A no response proceeds to step 17.

17) The screen will clear and present the following selections.

Source File: B:DUMMY.ABC

Same drive and name
I Input new drive and name
A Abort -- do not save

Your Choice:

The source file (e.g. B:DUMMY.ABC) will be the drive and name of the last file recalled for merging.

Striking RETURN will write the edited data over the source file -- THE OLD FILE WILL BE DESTROYED!

Striking "I" will allow the user to type in a new name and drive for storing the edited data.

Striking "A" will return to the editing menu without storing the data.

18) The program returns to the editing menu.

MODIFY SAMPLE CODES

DESCRIPTION

This option allows (1) a new sample-code file to be created, (2) existing sample codes to be edited, and (3) sample-code files to be culled or merged.

This option always operates on the current sample-code file as displayed at the bottom of the main menu.

OPERATION

Strike "M" while the main menu is displayed. The screen displays the sample codes that are defined in the current sample-code file. A menu of editing options are given at the bottom of the screen. The screen will look something like the following and the editing options are described in detail below.

SAMPLE CODE FILE: A:CODES

1. WRF	11.CD1	21.	31.	41.
2. NIG	12.	22.	32.	42.
3. WRT	13.	23.	33.	43.
4. CPW	14.	24.	34.	44.
5. P	15.	25.	35.	45.
6. OL	16.	26.	36.	46.
7. OP	17.	27.	37.	47.
8. GL	18.	28.	38.	48.
9. GA	19.	29.	39.	49.
10. GLU	20.	30.	40.	50.

P-print A-add E-edit D-delete M-merge C-cull F-file Esc-done

P-Print

A printer must be on and ready to receive data, otherwise the program will return an error message and return to the main menu. The print out is a complete listing of the sample codes and associated entry labels (e.g. MGO, AL203, etc.).

A-Add

This options allows the user to create a new sample code and add it to the current sample-code file. This option operates as follows.

1) Strike "A" when the option menu is displayed.

2) The program reports the next available entry number (e.g. Entry number 12) in the sample-code file and asks for the code.

```
Entry number  12

Mineral Code?
```

3) Type a code that is one to three characters long and that is not already used in the current file.

4) Strike RETURN.

5) The screen clears and then displays the Sample Code Editor, which should look as follows. Entry labels are created using instructions below.

```
 1                    S A M P L E   C O D E   E D I T E R
 2
 3
 4                              OPERATIONS:
 5
 6                         Del       Delete line
 7                         Ins       Insert line
 8                         Alt/F2    Change Code
 9
10                         ESC       Done
11
12
13                              VARIABLES:
14
15                       !   Floating point number
16                       *   Parts per million
17                       #   Parts per billion
18
19                    All others are weight percent
20
21
22                              Entry: 12
23                              Code: PX
24
```

The code being created or edited and its entry number in the file are displayed in the lower right corner of the screen. Numbers are shown on the left side of the screen to assist in organizing the entry labels. Note that only 24 numbers (and entry labels) can be viewed at a time, but it is possible to scroll through a maximum of 50 using the cursor control keys, "↑ ↓".

As many labels may be entered as desired up to a total of fifty. Enter labels as follows.

1. Move the arrow to the appropriate entry number using the cursor control keys, "↑ ↓".

2. Type the entry followed by a RETURN.

3. The arrow will advance to the next entry.

4. Type the next entry and strike RETURN.

5. Continue until the desired labels have been entered.

Labels must be one to five characters in length. Labels that begin with "!", "*" and "#" are treated as general floating point numbers, ppm concentrations and ppb concentrations, respectively. All other formats are treated as weight percent entries.

Correct an entry as follows. (1) Move the arrow to the erroneous label. (2) Delete it by striking the "Del" key. (3) Type the correct label followed by a RETURN.

Additional labels may be inserted into the list by moving the cursor to the entry number where insertion(s) is(are) desired and striking the "Ins" key as many times as desired.

The sample code may be changed by striking "Alt/F2" and typing in a new code.

Strike ESC to store the code and associated labels. The program will summarize the contents of the current sample-code file.

E-edit

The editing option allows all of the entries of a sample code to be modified. The editing option operates as follows.

1) Strike "E" when the option menu is displayed.

2) The program asks:

Code to change?

3) Type the code followed by a RETURN.

4) The program displays the entry labels using the Sample Code Editor. Operation of the Sample Code Editor is explained in the previous section (A-Add).

D-Delete

The program eliminates an indicated code and associated labels from memory. This option operates as follows.

1) Strike "D" with the option menu displayed.

2) The program asks:

Code to delete?

3) Type the code followed by a RETURN.

4) The code is deleted from the sample-code file in memory and the contents of the file is displayed.

M-merge

The merge option appends the contents of another sample-code file to the current sample-code file. If the total number of codes exceeds 50, the program truncates the codes at the 50th entry. The merge option operates as follows.

1) Strike "M" with the option menu displayed.

2) The screen will clear and the following question will appear.

```
Code file to merge?
```

3) Type the name of the sample-code file (e.g. DUMMY.ABC) and strike RETURN. Striking RETURN without typing a name will cause the program to use the name of the most recently used sample-code file. The program will then ask for the drive location of the sample-code file.

```
Code file to merge? DUMMY.ABC     Drive (A or B)?
```

Note: type "DIR" for the file name to display the directory of the drive indicated in the next step (step 4).

4) Type the letter of the drive (A, B, C, etc.) followed by a RETURN.

Typing RETURN without a letter will cause the program to use the drive of the most recently used sample-code file. Typing two or more letters (e.g. "XX") followed by a RETURN will return the program to the main menu.

5) The program reads the file from disk and appends its contents to the sample-code file in memory.

6) The program displays a summary of the current sample-code file and a menu of options.

C-cull

The cull option allows the user to select (cull) codes from the current file. The user may cull as many codes as desired and even cull the same code more than once. When culling is complete, codes that were not culled are deleted from memory. The user may then store the modified file in memory on disk as a sample-code file or continue to modify the file in memory. The cull option operates as follows.

1) Strike "C" with the option menu displayed.

2) The program asks:

Code to cull?

3) Type the code (three characters or less) followed by a RETURN.

4) The program asks:

Cull another (Y/N)?

5) A "Y" response returns the program to step 2.
A "N" response proceeds to step 6.

6) All codes that were not culled are deleted from the file in memory.

7) The program displays a summary of the current sample-code file and a menu of options.

The cull option may be used to facilitate creating new sample codes from existing codes. Consider a file containing the following code and associated labels.

Code	Labels							
PX	NA2O	MGO	AL2O3	SIO2	CAO	TIO2	MNO	FEO

The user wants to add the code CPX, which is similar to PX but has labels for K2O and trace elements abundances as follows.

Code	Labels							
CPX	NA2O	MGO	AL2O3	SIO2	K2O	CAO	TIO2	MNO	FEO
	*LA	*CE	*PR	*ND	*SM				

The user could type the code in from scratch using the "A-add" option. Using cull, however, the same effect can be achieved with the following steps.

1) Cull PX -- this will eliminate all codes except PX from memory.

2) Edit the PX code:

   a. use Alt/F2 to change the code to CPX
   b. use "Ins" to create a space between SIO2 and CAO
   c. type in K2O
   d. move to the end of the labels and enter *La, *Ce, *Nd and *Sm
   e. strike ESC.

3) Merge the original file (still present on disk) to the file in memory

4) Store the new file

The new file will begin with CPX and contain the entire contents of the original sample code file.

F-file

This option changes the name of the current sample-code file. This option operates as follows.

1) Strike "F" when the option menu is displayed.

2) The screen will clear and the following question will appear.

   Sample-Code File?

3) Type the name of the sample-code file (e.g. DUMMY.ABC) and strike RETURN. Striking RETURN without typing a name will cause the program to use the name of the most recently used sample-code file. The program will then ask for the drive location of the sample-code file.

   Sample-Code File? DUMMY.ABC Drive (A or B)?

   Note: type "DIR" for file name to display the directory of the drive indicated in the next step (step 4).

4) Type the letter of the drive (A, B, C, etc.) followed by a RETURN.

   Typing RETURN without a letter will cause the program to use the drive of the most recently used sample-code file. Typing two or more letters (e.g. "XX") followed by a RETURN will return the program to the main menu.

5) The displays a summary of the current sample-code file and a menu of options. Note: the file is not stored until the user strikes ESC when the option menu is displayed.

ESC-Done

This option stores the file in memory using the current file name. Operation is as follows.

1) Strike ESC when the option menu is displayed.

2) If the sample-code file has not been modified, the program returns to the main menu.

3) If the sample-code file has been modified then the following message is displayed.

   The sample-code file has been modified!

   Assign a new name (Y/N)?

4) A "N" response stores the sample-code file on disk using the current name and the program returns to the main menu.
   A "Y" response proceeds to step 5.

5) The screen will clear and the following question will appear.

   Sample-Code File?

6) Type the name of the sample-code file (e.g. DUMMY.ABC) and strike RETURN. Striking RETURN without typing a name will cause the program to use the name of the most recently used sample-code file. The program will then ask for the drive location of the sample-code file.

   Sample-Code File? DUMMY.ABC Drive (A or B)?

   Note: type "DIR" for file name to display the directory of the drive indicated in the next step (step 4).

7) Type the letter of the drive (A, B, C, etc.) followed by a RETURN.

   Typing RETURN without a letter will cause the program to use the drive of the most recently used sample-code file. Typing two or more letters (e.g. "XX") followed by a RETURN will return the program to the main menu.

8) The program returns to the main menu.

SELECT SAMPLE-CODE FILE

DESCRIPTION

This option allows the current sample-code file to be changed. This is necessary if the user wants to use a specialized sample-code file on another disk.

The name and drive of the sample-code file are stored in the DOS work space and are retained in memory unless the computer is rebooted. The user may move from one DATASAVE program to another without reentering the sample-code file name or drive.

OPERATION

1) Strike "S" while the main menu is displayed.

2) The screen will clear and the following question will appear.

   Sample-Code File?

3) Type the name of the sample-code file (e.g. DUMMY.ABC) and strike RETURN. Striking RETURN without typing a name will cause the program to use the name of the most recently used sample-code file. The program will then ask for the drive location of the sample-code file.

   Sample-Code File? DUMMY.ABC Drive (A or B)?

   Note: type "DIR" for file name will display the directory of the drive indicated in the next step (step 6).

4) Type the letter of the drive (A, B, C, etc.) followed by a RETURN.

   Typing RETURN without a letter will cause the program to use the drive of the most recently used sample-code file. Typing two or more letters (e.g. "XX") followed by a RETURN will return the program to the main menu.

5) The program returns to the main menu. Note that the name of the current sample-code file has changed.

S E C T I O N 5

I N T R O D U C T I O N T O S T A C K

Contents

INTRODUCTION

STACK is a programming language specifically developed for writing graphics subroutines for DATAPLOT. It is the cornerstone of DATAPLOT and must be mastered before new graphs may be created.

STACK allows the user to define the axes of plots by writing small, reverse-Polish programs. Programs up to 300 steps long may be written in STACK using the editor provided in DATAPLOT. Fifty memories are available for storing numbers. Math functions include all arithmetic functions, sine, cosine, arctangent, logarithmic and exponential functions, square roots, absolute value and negation. Program flow may be controlled by "if-then" and "goto" commands. A "trace on" command allows subroutine operation to be traced for debugging. STACK program statements are provided to (1) select symbols and (for EGA systems) colors for data points and (2) connect data points with lines. STACK programs may be stored on disk, recalled, edited and listed.

None of the above capabilities can be exploited until the user understands the principles of reverse Polish notation. The following section presents an introduction to reverse Polish. A manual for any programmable Hewlett-Packard calculator is recommended for additional instruction.

REVERSE-POLISH NOTATION

Conventional algebraic notation places functions between the relevant numbers in mathematical expressions. Thus the sum of 3 and 4 is represented by the following.

```
3 + 4
```

Polish notation places functions before the relevant numbers, so that the above sum would be represented by the following.

```
+ 3 4 .
```

Computer programs operate more efficiently if the functions follow the numbers. That way, the numbers are already available when the computer gets an instruction to do something with them. FORTH utilizes this principal and is, therefore, one of the fastest executing languages. This approach to calculations may be expressed with a notation that is known as reverse Polish. The sum of 3 and 4 in reverse Polish notation is as follows.

```
3 4 +
```

The reason that programming in reverse Polish is efficient is seen when evaluating a more complex expression. Consider the following expression written in algebraic notation (where * indicates multiplication).

```
8 * ( 3 + 4 )
```

Evaluation of this expression requires the computer to read through it and determine which step, 3 + 4 , is to be performed first. The computer must scan the expression again to determine which step is performed next. In contrast, the same expression translates to reverse Polish as the following.

```
3 4 + 8 *   or  8 3 4 + *
```

The computer does not have to interpret the expression before beginning the calculation; instead, each function is performed as it is encountered from left to right. The longer the expression, the more efficient reverse Polish notation is relative to algebraic notation.

Computers evaluate reverse Polish notation by creating stacks of numbers. Hence, the basis for naming this language STACK. The concept is that numbers may be placed in order in the computer's memory with each number "stacked" on top of the previous number.

An analogy is often made to stacking dishes. As each dish is washed, it is placed on the top of a stack. When a clean dish

is required, it is removed from the top of this stack. Therefore, the last dish placed on the stack is the first used. Similarly, a computer starts from the top of a "stack" of numbers and works its way toward the bottom during execution.

Consider the last expression in reverse polish. The numbers would be entered or "stacked" into the computer's memory in the order, 8 followed by 3 followed by 4. This could be represented graphically by the following "stack".

```
4
3
8
```

STACK functions typically operate by removing the top one or two numbers from the stack and placing the result of the calculation back on the top of the stack. Consider again the following calculation.

```
8 3 4 + * .
```

This results in a stack being created and then modified with each step in the calculation. This may be represented in the following manner.

```
8          3          4          7          56
                        +          *
  ------>  8  ------>  3  ------>  8  ------>
                       8
```

The number 8 is placed on the stack, followed by 3, which is followed by 4. The operation, "+", removes 4 and 3 from the top of the stack and put their sum, 7, on the top of the stack. The operation, "*" removes 7 and 8 from the top of the stack and places their product on the stack. The same calculation may also be performed with the following notation.

```
4  3 + 8 *
```

This may be represented as follows.

```
4          3          7          8          56
                      +                     *
  ------> 4 ------>          ------> 7 ------>
```

The same number of steps are involved but the stack is never as deep. First, 4 is placed on the stack followed by 3. The operation "+" removes 3 and 4 and puts their sum, 7, on the stack. The number, 8, is added to the stack. The operation, "*", removes 8 and 7 from the stack and places their product, 56, on the stack.

PROGRAMMING STATEMENTS

STACK has three basic types of statements. Data statements instruct the computer to place a specified datum on the stack. Constant statements instruct the computer to place a specified numerical constant on the stack. Operation statements instruct the computer to perform the indicated numerical or logical function (e.g. multiplication). Statements have the following syntactcal forms.

Data statement:

###.	d	ABCDE	where ABCDE is a left-justified, five-character, alpha-numeric field.

Constant statement:

###.	c	1234567890	where 1234567890 is a left-justified, floating point, numerical field. Exponential notation is not allowed.

Operation statement:

###.	o	XYZ	where XYZ must obey syntax described under "Operations".

Statements are identified by a line number, "###.", which is automatically supplied by STACK. The "d", "c" and "o" identify the statement as a data statement, constant statement or operation statement, respectively.

The following examples illustrate the three types of statements and explain their specific functions.

Example 1: Statement 1 of a hypothetical subroutine. This is an example of a data statement that places the SIO2 concentration on the stack.

```
1  d   SIO2
```

Example 2: Statement 5 of a hypothetical subroutine. This is an example of a constant statement that places the value of pi on the stack.

```
5  c   3.141592
```

Example 3: Statement 108 of a hypothetical subroutine. This is an example of an operation statement that removes the top two numbers from the stack and places their product on the stack.

```
108  o   *
```

SUBROUTINE EDITOR

INTRODUCTION

STACK subroutines are created and edited with a subroutine editor. The operation of the editor is described in this section. Gaining access to the editor is described under PROGRAM OPERATION. The syntax and function of individual STACK statements are described in detail under STACK STATEMENTS.

APPEARANCE

The subroutine editor will appear as follows if a new subroutine is being created from scratch. The numbers identify program steps. Twenty steps are displayed at a time but up to 300 steps may be viewed by scrolling the screen. An arrow points to the current step where a STACK statement may be written.

```
SUBROUTINE EDITOR

 1 >
 2
 3
 4
 5
 6
 7
 8
 9
10
11
12
13
14
15
16
17
18
19
20

          Ctrl/F1--List          ESC--Done
```

KEYBOARD CONTROL

The subroutine editor controls the "active keys" on the keyboard and the case of the letters. Do not strike "Num Lock" or "Caps Lock" while the editor is displayed as this will interfere with its operation.

EDITING KEYS

Writing and editing a subroutine is performed with the following editing keys.

KEY	FUNCTION
Home	Moves cursor to top step on screen.
End	Moves cursor to bottom step on screen.
↑	Moves cursor up one step.
↓	Moves cursor down one step.
+	Advances the display 20 steps.
-	Retreats the display 20 steps.
Ins	Inserts one blank step at the cursor.
Del	Deletes one step at the cursor.
Ctrl/F1	Lists the subroutine on a printer.
Esc	Terminates editing / stores subroutine.
d	Initializes a data statement.
c	Initializes a constant statement.
o	Initializes an operation statement.

WRITING A STACK STATEMENT

A STACK statement may be written on any blank line using the following steps.

1) Place place the arrow on a blank line.

2) "Initialize" the statement by striking "d", "c" or "o".

   This temporarily turns off the editing keys. A lower-case "d", "c" or "o" will appear on the screen beside the arrow.

3) Write a STACK statement using alpha-numeric entries from the keyboard.

   The statement should be written without using the space bar. Note, STACK automatically places spaces between the statement and the preceding "d", "c" or "o".

   Errors may be corrected by striking "<-" (backarrow) which will erase one character to the left. Alternatively, strike the "Del" key to erase the entire statement.

4) Strike RETURN to enter the statement into memory.

5) The program activates the editing keys and advances the arrow to the next subroutine step.

EDITING A STACK STATEMENT

An existing statement can be modified only by rewriting it. The following steps may be used to rewrite a statement.

1) Move the arrow to the statement that will be modified.

2) Strike the "Del" key.

3) Strike the "Ins" key.

4) Write the new statement as described in the previous section (Writing a STACK Statement).

EXITING THE EDITOR

Strike ESC to exit the subroutine editor. Remember that the ESC key is not responsive when a STACK statement is being written. The last statement must be finished by striking RETURN before the ESC key will respond.

STACK scans a program before it is stored and scanning stops at the first empty line. Delete any empty lines in the middle of the subroutine before exiting the editor. All statements that follow the first empty line will be lost.

Exiting the subroutine editor involves the following steps.

1) Strike the ESC key.

2) The program will ask:

   Subroutine name?

3) Type the name that the subroutine will be stored under on disk (e.g. AFM) followed by a RETURN.

   This is the only acceptable input to this question. DO NOT ENTER "DIR" OR A RETURN WITHOUT A NAME.

   The name should be eight characters or less in length and obey DOS conventions. DO NOT INCLUDE AN EXTENSION AS DATAPLOT WILL AUTOMATICALLY APPEND ".SUB" TO THE NAME.

4) The program will ask for a drive to store the program.

   Subroutine Name? AFM Drive?

5) Type the letter of the drive (A, B, C, etc.) followed by a RETURN.

   Type two or more letters (e.g. "XX") followed by a return to return the program to the main menu without storing the subroutine.

SUBROUTINE EXECUTION

DATAPLOT runs the current STACK subroutine once for each sample during normal execution. At the end of each iteration through the subroutine, the program looks at the last values assigned to X, Y, A, B, or C and includes them in an array that is latter used to plot points on the screen. At this juncture, the program looks to see if another sample needs to be plotted. If so, the program clears the "STACK stack" and returns to step 1 in the STACK subroutine. If not, the program leaves STACK and creates a graph on the screen.

Exceptions to this normal execution are introduced by the use of STACK commands, "CHAIN" and "HOLD" (see Section 6).

PROGRAM TIPS

The user should remember that DATAPLOT takes care of data input and looping through the STACK subroutine. With this in mind, write a STACK subroutine as if a calculation were going to be performed using only one sample (or the appropriate number of samples if the "CHAIN" statement is used).

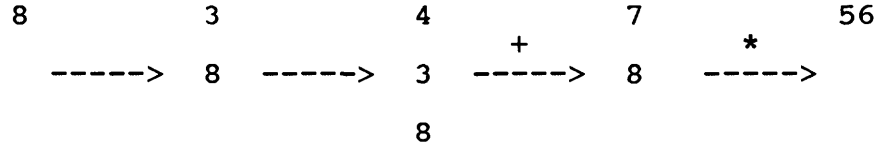
A program written for a Hewlett-Packard programmable calculator can be translated with little effort and only minor modifications in logic.

SUBROUTINE EXAMPLES

Example 1: The calculation 8 3 4 + * in reverse Polish notation, is performed by the following STACK subroutine.

```
1.  c  8
2.  c  3
3.  c  4
4.  o  +
5.  o  *
```

The appearance of the stack during the execution of this subroutine has already been discussed in the introduction to reverse Polish and may be represented as follows.



Example 2: The following subroutine produces a K2O verse SiO2 plot.

Program:			Description:
1.	d	K2O	Recall K2O and place on stack.
2.	o	=Y	Set the top of stack (K2O) = Y-axis.
3.	d	SIO2	Recall SIO2 and place on stack.
4.	o	=X	Set the top of the stack (SIO2) = X-axis.

Example 3: The following subroutine calculates an AFM diagram.

Program:			Description:
1.	d	NA2O	Recall NA2O and place on stack.
2.	d	K2O	Recall K2O and place on stack.
3.	o	+	Compute NA2O+K2O and place on stack.
4.	o	=A	Set top of stack (NA2O+K2O) = A-apex.
5.	d	FE2O3	Recall Fe2O3 and place on stack.
6.	c	.9	Place constant (.9) on stack.
7.	o	*	Compute .9FE2O3 and place on stack.
8.	d	FEO	Recall FEO and place on stack.
10.	o	+	Place FEO + .9FE2O3 on stack.
11.	o	=B	Set top of stack (FEO + .9FE2O3) = B-apex.
12.	d	MGO	Recall MGO and place on stack.
13.	o	=C	Set top of stack (MGO) = C-apex.

S E C T I O N 6

S T A C K S T A T E M E N T S

Contents

INTRODUCTION

This section describes the function and syntax of STACK statements. The STACK editor that is used to write programs is described in Section 7. Statements are provided to (1) perform math, (2) control data input and output (I/O), (3) control program flow by branching, (4) store, recall and manipulate numbers in 50 memories, (5) control plotting of symbols, colors and lines, and (6) debug subroutines. Forty-five statements are presently available and are organized as follows.

MATH STATEMENTS

INPUT/OUTPUT STATEMENTS

BRANCHING STATEMENTS

MEMORY-CONTROL STATEMENTS

PLOT-CONTROL STATEMENTS

DEBUGGING STATEMENTS

ADDITION, SUBTRACTION

+ ADDITION

Removes the top two numbers from the stack and places their sum on the stack.

```
5   +    11
6 -----> 8
8
```

Example: The following subroutine adds K2O and NA2O and sets them equal to the Y-axis. SIO2 is set equal to the X-axis.

```
1. d  K2O
2. d  NA2O
3. o  +
4. o  =Y
5. d  SIO2
6. o  =X
```

- SUBTRACTION

Removes the top two numbers from the stack, subtracts the upper number from the lower number and places the difference on the stack.

```
8   -    -2
6 -----> 1
1
```

Example: The following subroutine subtracts the background of an Cu x-ray peak (!BKG) from the peak intensity (!PK) and plots the result again the Cu concentration.

```
1. d  !PK
2. d  !BKG
3. o  -
4. o  =Y
5. d  *CU
6. o  =X
```

MULTIPLICATION, DIVISION

* MULTIPLICATION

Removes the top two numbers from the stack and places their product on the stack.

```
3   *    12
4 -----> 9
9
```

Example: The following subroutine reads CR2O3 and AL2O3 weight percents, calculates cation abundances and plots a histogram of the Cr/Cr+Al values.

```
 1. d  CR2O3          11. o  *
 2. c  151.96         12. o  RM01
 3. o  /              13. o  +
 4. c  2              14. o  RM01
 5. o  *              15. o  SWAP
 6. o  SM01           16. o  /
 7. d  AL2O3          17. o  =X
 8. c  101.96
 9. o  /
10. c  2
```

/ DIVISION

Removes the top two numbers from the stack, divides the lower number by the upper number, and places the dividend on the stack.

```
6    /    5
30 -----> 1.5
1.5
```

Example: The following subroutine plots a histogram of MGO/FEO.

```
1. d  MGO
2. d  FEO
3. o  /
4. o  =X
```

SQUARE ROOT, ABSOLUTE VALUE

SQR SQUARE ROOT

Removes the top number from the stack and replaces it with it's square root. The number must be positive or an error will result.

```
16    SQR     4
 1   ------>  1
83           83
```

Example: the following program lines calculate the square root of the !CTS (which could represent the total counts in an experiment).

```
1. d  !CTS
2. o  SQR
```

ABS ABSOLUTE VALUE

Removes the top number from the stack and replaces it with it's absolute value.

```
-2    ABS     2
 3   ------>  3
 0            0
```

Example: the following program lines calculate the absolute value of !X1 minus !X2.

```
1. d  !X1
2. d  !X2
3. o  -
4. o  ABS
```

NEGATION, POWER

NEG NEGATION

Removes the top number from the stack and replaces it with the negative of that number.

```
 2    NEG    -2
 3   ------>  3
 0            0


-2    NEG     2
 3   ------>  3
 0            0
```

Example: the following program lines calculate the negation of !X1.

```
1. d  !X1
2. o  NEG
```

^ POWER

Removes the top two numbers from the stack, raises the lower number to the power of the upper number, and places the result on the stack.

```
 5    ^     32
 2  ------>  4
 4
```

Example: the following program lines calculate the square of !DIST (which could represent the distance from a radioactive source).

```
1. d  !DIST
2. c   2
3. o   ^
```

EXPONENTIAL, LOG (BASE E), LOG (BASE 10)

EXP EXPONENTIAL

Removes the top number from the stack and replaces it with it's exponential.

```
1    EXP   2.71828
5   -----> 5
14         14
```

Example: the following program lines calculate the exponential of !X1.

```
1. d  !X1
2. o  EXP
```

LOG LOGARITHM (NATURAL)

Removes the top number from the stack and replaces it with it's natural log.

```
1    LOG   .693147
5   -----> 5
14         14
```

Example: the following program lines calculate the natural log of !X1.

```
1. d  !X1
2. o  LOG
```

LOG10 LOGARITHM (BASE 10)

Removes the top number from the stack and replaces it with it's base 10 log.

```
2    LOG   .301030
5   -----> 5
14         14
```

Example: the following program lines calculate the log (base 10) of !X1.

```
1. d  !X1
2. o  LOG10
```

SINE, COSINE

SIN SINE

Removes the top number from the stack (which is presumed to be expressed in degrees) and replaces it with it's sine.

```
 57    SIN     0.43616
  1   ------>   1
 18             18
```

Example: the following program lines calculate the sine of !X1.

```
1. d  !X1
2. o  SIN
```

COS COSINE

Removes the top number from the stack (which is presumed to be expressed in degrees) and replaces it with it's cosine.

```
 12    COS     0.84385
 56   -----> 56
  3             3
```

Example: the following program lines calculate the cosine of !X1.

```
1. d  !X1
2. o  COS
```

TANGENT, ARCTANGENT

TAN TANGENT

Removes the top number from the stack (which is presumed to be expressed in degrees) and replaces it with it's tangent.

```
12    TAN     0.212557
56   -----> 56
 3            3
```

Example: the following program lines calculate the tangent of !X1.

```
1. d  !X1
2. o  TAN
```

ATN ARCTANGENT

Removes the top number from the stack and replaces it with it's arctangent expressed in degrees.

```
.5    ATN     26.5651
56   -----> 56
 3            3
```

Example: the following program lines calculate the arctangent of !X1.

```
1. d  !X1
2. o  ATN
```

SWAP

SWAP STACK MANIPULATION

Exchanges the positions of the top two numbers in the stack.

```
 6    SWAP   14
14   ------>  6
 1             1
```

Example: the following program lines calculate MGO/MGO+FEO.

```
1. d  MGO
2. d  FEO
3. o  +
4. d  MGO
5. o  SWAP
6. /
```

ASSIGN VALUE TO X-AXIS, Y-AXIS

=X EQUALS X AXIS

Sets the value at the top of the stack equal to "X", i.e. the ordinate.

This operation is valid for creating orthogonal plots and histograms only.

Example: The following subroutine will plot a histogram of SiO2 abundances.

```
1.  d  SIO2
2.  o  =X
```

(Note: if used with the orthogonal plot option, this subroutine will plot a series of SiO2 concentrations along the X-axis of the plot because no Y values are defined.)

=Y EQUALS Y AXIS

Sets the value at the top of the stack equal to "Y", i.e. the coordinate.

This operation is valid for creating orthogonal plots only.

Example: The following program will generate a K2O vs. SiO2 plot.

```
1.  d  SIO2
2.  o  =X
3.  d  K2O
4.  o  =Y
```

ASSIGN TERNARY APICES

=A	EQUALS A APEX
=B	EQUALS B APEX
=C	EQUALS C APEX

These operations set the top of the stack equal to either the A, the B or the C apex of a ternary diagram for plotting.

These operations are valid for constructing ternary plots only.

A, B, C may be used in the program in any order, and DATAPLOT normalizes the values for A, B and C automatically.

The triangular plot will always be generated with the apices arranged as follows.

B

A C

Example: The following program generates an AFM diagram.

```
 1.  d  NA2O
 2.  d  K2O
 3.  o  +
 4.  o  =A
 5.  d  FEO
 6.  d  FE2O3
 7.  c  .9
 8.  o  *
 9.  o  +
10.  o  =B
11.  d  MGO
12.  o  =C
```

CHAIN

CHAIN NEXT DATA SET, DO NOT PLOT

CHAIN commands the program to call up the next analysis in the data set without plotting a point.

(Normally, a point is plotted for each analysis in a given data set.)

CHAIN is useful when two analyses are required to generate a single point on a plot.

Example: the following subroutine will generate a plot of TIO2 in spinel verses TIO2 in clinopyroxene if the data file begins with a spinel analysis and alternates every-other-one between spinel and clinopyroxene.

```
1.  d  TIO2      gets TIO2 from spinel
2.  o  =Y        sets spinel TIO2 = Y
3.  o  CHAIN     calls up next cpx
4.  d  TIO2      gets TIO2 from cpx
5.  o  =X        sets cpx TIO2 = X
```

At the conclusion of the subroutine, the program will establish a point and continue to the next analysis in the data file, which should be another spinel.

CAUTION: identification of points on the screen using the cursor is inhibited by CHAIN.

HOLD

HOLD PLOT, REMAIN IN SAME DATA SET

HOLD commands the program to plot a point using the defined values (=X, =Y, etc.) but to abort the normal transition to the next analysis in the data file.

Example: the following subroutine will plot the chondrite normalized values for 2 rare earth elements for each sample in a data file.

```
 1.  o  LON    connects points with line
 2.  d  LA     recalls LA abundance
 3.  c  .33    chondritic abundance for LA
 4.  o  /      normalizes LA to chondritic
 5.  o  =Y     set value from step 4 = Y
 6.  c  1      La is first REE
 7.  o  =X     sets LA ordinate = 1
 8.  o  HOLD   causes LA point to be plotted
 9.  d  ND     recalls ND abundance
10.  c  .6     chondritic abundance for ND
11.  o  /      normalizes ND to chondritic
12.  o  =Y     set value from step 12 = Y
13.  c  4      ND is fourth REE
14.  o  =X     sets ND ordinate = 4
15.  o  LBRK   breaks line between analyses
```

The subroutine will plot LA and ND for each analysis in the data file. The program could easily be extended to plot complete REE patterns for each analysis.

CAUTION: identification of data points on the screen using the cursor is inhibited by HOLD.

OMIT

OMIT OMIT CURRENT DATA SET FROM THE CALCULATION

OMIT commands the program to omit the current sample from the plot, skip ahead to the next sample, and restart the STACK program at line 1.

The OMIT statement is useful for limiting a plot to analyses that fall within a specified compositional range.

An OMIT statement may occur at any point in the program.

Example: the following subroutine will create a K2O vs. SiO2 plot using only those samples that have less than or equal to 12 weight percent CaO.

```
1. d  CAO
2. o  >12
3. o  OMIT
4. d  SIO2
5. o  =X
6. d  K2O
7. o  =Y
```

Line 1 recalls the weight percent of CAO in the current sample. If CAO is greater than 12 weight percent, the current sample is omitted from the calculation. Otherwise, the weight percent of SIO2 is assigned to the X-axis and the weight percent of K2O is assigned to the Y-axis.

GOTO, IF EQUAL THEN

GT### GOTO

Directs the program to go to line ###, skipping the intervening lines. There is no space between "GT" and "###". ### may be an integer from 1 to 3 characters long.

Example:

```
1.  d  FEO
2.  d  FEO
3.  d  MGO
4.  o  +
5.  o  =0
6.  o  GT9
7.  o  /
8.  o  GT10
9.  c  0
10. o  =X
```

Line 1 loads the FEO concentration onto the stack. Line 2 loads the FEO concentration onto the stack. Line 3 loads the MGO concentration onto the stack. Line 4 adds MGO and FEO. Line 5 tests the sum; if the sum equals 0 then the program is directed to line 9 to avoid a division by zero error. The program then continues to line 10 where the X value is set to 0 for plotting. If the sum is not zero then the value, FEO/(FEO+MGO), is determined in line 7 and line 8 directs the program to line 10 where this is set equal to the X value for plotting.

IF EQUAL THEN

=### IF EQUAL THEN

If the value at the top of the stack equals the indicate value to the right of the "=", the following line is executed. Otherwise, the following line is skipped.

may be a floating point number up to 6 characters long.

Example:

```
10. o  =3.1416
11. o  GT20
12. ....
```

If the top of the stack equals 3.1416 (pi) then line 11 is executed and the program jumps to line 20 with a goto statement.

IF GREATER THEN, IF LESS THEN

>### IF GREATER THEN

If the value at the top of the stack is greater than the indicate value to the right of the ">", the following line is executed. Otherwise, the following line is skipped.

may be a floating point number up to 6 characters long.

Example:
```
10. o  >0
11. o  GT20
12. ....
```

If the top of the stack exceeds 0, then line 11 is executed and the program jumps to line 20 with a goto statement.

<### IF LESS THEN

If the value at the top of the stack is less than the indicate value to the right of the "<", the following line is executed. Otherwise, the following line is skipped.

may be a floating point number up to 6 characters long.

Example:
```
10. d CAO
11. o <10
12. o GT30
```

Line 10 places the concentration of CAO at the top of the stack. If CAO is less than 10, then line 11 is executed and the program jumps to line 30 with a goto statement.

IF "NOT ANALYZED" THEN,
IF "BELOW DETECTION LIMIT" THEN

=NA IF "NOT ANALYZED" THEN

If the value at the top of the stack was stored as "not analyzed" (see DATAEDIT), the following line is executed. Otherwise, the following line is skipped.

Example:
```
1. d  !TEMP
2. o  =NA
3. o  OMIT
4. o  =X
```

This program could be used to plot histograms of fluid inclusion temperatures. Data would not be used in the histogram unless the temperature (!TEMP) was actually measured. Line 1 recalls the temperature (!TEMP) from the current sample. If the temperature was not analyzed then "OMIT" is executed and the program recalls the next sample and starts again at line 1. Otherwise, program execution continues at line 4. (See OMIT.)

=BDL IF "BELOW DETECTION LIMIT" THEN

If the value at the top of the stack was stored as "below detection limit" (see DATAEDIT), the following line is executed. Otherwise, the following line is skipped.

Example:
```
1. d  BAO
2. o  =NA
3. o  OMIT
4. ......
```

The program recall the value for BAO. If BAO was not analyzed then "OMIT" is executed and the program recalls the next sample and starts again at line 1. Otherwise, program execution continues at line 4. (See OMIT.)

IF "BELOW DETECTION LIMIT" OR
"NOT ANALYZED" OR EQUAL TO ZERO THEN

=ZIP IF "BELOW DETECTION LIMIT" OR "NOT ANALYZED" THEN

If the value at the top of the stack was stored as "below detection limit" or "not analyzed" (see DATAEDIT), the following line is executed. Otherwise, the following line is skipped.

Example:

```
1. d  *LA
2. o  =ZIP
3. o  OMIT
4. ......
```

The program recalls the value for LA (in ppm). If LA was not analyzed then "OMIT" is executed. (See OMIT.) If the current value for LA equals zero or was below the detection limit then "OMIT" is executed. Otherwise, program execution continues at line 4. (See OMIT.)

SAVE IN MEMORY

SM##$ SAVE IN MEMORY

Reads a value from the top of the stack and stores it in the specified memory, ##. ## may be any integer that ranges from 1 to 50.

$ may be used to specify arithmetic operations in addition to simple storage. $ may be left blank or may be "+", "-", "*" or "/" with the following results.

Blank -- top of stack is stored in memory ## in place of existing memory contents

\+ -- top of stack is added to contents of memory ## and the sum is placed in memory ##.

\- -- top of stack is subtracted from contents of memory ## and the difference is placed in memory ##.

* -- top of stack is multiplied by the contents of memory ## and the product is placed in memory ##.

/ -- top of stack is divided into the contents of memory ## and the quotient is placed in memory ##.

Note: the stack is not affected by the SM##$ function. Storing a number at the top of the stack in a memory, leaves that number on the stack.

Example:

```
1. d  FEO
2. o  SM01
3. d  MGO
4. o  SM01+
5. o  RM01/
6. o  =X
```

This subroutine calculates MGO/(MGO+FEO) using memory 01. Lines 1 and 2 recall FEO and store it in memory 01. Lines 3 and 4 recall MGO and add it to the contents of memory 01. MGO is the last entry on the stack. Line 5 divides the last entry on the stack (MGO) by the contents of memory 01 and places the quotient on the stack. Line 6 sets the top of the stack equal to the X-axis.

READ FROM MEMORY

RM##$ READ FROM MEMORY

Reads a number from memory ## and places it on the top of the stack.

$ may be used to specify arithmetic operations in addition to placing a number on the stack. $ may be left blank or may be "+", "-", "*" or "/" with the following results.

Blank -- the contents of memory ## is simply placed on the top of the stack.

\+ -- the contents of memory ## is added to the top of the stack and the top of the stack is replaced by the sum.

\- -- the contents of memory ## is subtracted from the top of the stack and the top of the stack is replaced by the difference.

* -- the contents of memory ## is multiplied by the top of the stack and the top of the stack is replaced by the product.

/ -- the contents of memory ## is divided into the top of stack and the top of the stack is replaced by the quotient.

Note: the RM##$ function does not alter the contents of memory ##.

Example: See SAVE IN MEMORY

DUMP DATA DIRECTLY FROM AN ANALYSIS INTO MEMORIES

D###%% DUMP

This operation loads data directly from an analysis sequentially into memories beginning from memory 1.

specifies the number of the first element/oxide of an analysis to "dump" into memory 1 and %% specifies the last element/oxide of the same analysis to "dump"; all intervening element/oxides are stored sequentially in memories/

Example: Consider the following analysis.

SIO2	55.0	
TIO2	1.0	
AL2O3	17.0	---> memory 1
FE2O3	.5	---> memory 2
FEO	5.0	---> memory 3
MGO	9.0	---> memory 4
CAO	9.0	
NA2O	3.0	
K2O	.2	

The line,

1. o D0306

will place the third through sixth oxides, i.e. Al2O3 through MGO, into memories 1 to 4, sequentially.

Example: See NORMALIZE MEMORIES

NORMALIZE MEMORIES

NM### NORMALIZE MEMORIES

Normalizes the contents of memories 1-20 to the value specified by ###. ### may be a positive or negative, floating point number, up to three characters long.

NM0 is a special case that sets memories 1-20 to zero.

This operation is useful for normalizing cation data, etc.

Example: Consider the following analysis.

SIO2	55.0
TIO2	1.0
AL2O3	17.0
FE2O3	.5
FEO	5.0
MGO	9.0
CAO	9.0
NA2O	3.0
K2O	.2

The following program will enter the contents of the analysis into memories 1 to 9, normalize the memories to 100 percent and plot the normalized value for SIO2 in a histogram.

1.	o	D0109	Dumps data into mem 1-9
2.	o	NM100	Normalizes mem 1-20 to 100
3.	o	RM1	reads normalized SIO2
4.	o	=X	sets normalized SIO2=X

LINE ON, LINE OFF, LINE BREAK

LON LINE ON

Turns the line drawing function on. All points plotted after this command will be connected by lines until a LOFF (line off) or LBRK (line break) is encountered. The line drawing function is off until activated by LON.

Example: the following subroutine will plot the chondrite normalized values for 2 rare earth elements for each analysis in a data file. The points for each analysis will be connected by a line.

```
 1.  o  LON     Turn line function on
 2.  d  LA      recalls LA abundance
 3.  c  .33     chondritic abundance for LA
 4.  o  /       normalizes LA to chondritic
 5.  o  =Y      set value from step 4 = Y
 6.  c  1       La is first REE
 7.  o  =X      sets LA ordinate = 1
 8.  o  CHAIN   causes LA point to be plotted
 9.  d  ND      recalls ND abundance
10.  c  .6      chondritic abundance for ND
11.  o  /       normalizes ND to chondritic
12.  o  =Y      set value from step 11 = Y
13.  c  4       ND is fourth REE
14.  o  =X      sets ND ordinate = 4
15.  o  LBRK    Breaks the line so that it is
                not drawn to the next point
                of the next analysis.
```

LOFF LINE OFF

Turns the line drawing function off. The last point being calculated when LOFF is encountered will not be connected to the previous point by a line. Subsequent points are not connected by lines until LON is encountered.

LBRK LINE BREAK

Turns the line drawing function off. The last point being calculated when LBRK is encountered is connected to the previous point by a line. Subsequent points are not connected by lines until LON is encountered. See LON for example.

PLOT POINT, DOT, SQUARE, TRIANGLE

PNT PLOT A POINT (one pixel)

Causes all subsequent points to be plotted as points (one pixel in size) until another plotting control function is encountered. This command overrides keyboard selection of symbols during plotting.

DOT PLOT A DOT

Causes all subsequent points to be plotted as dots until another plotting control function is encountered. This command overrides keyboard selection of symbols during plotting.

SQU PLOT A SQUARE

Causes all subsequent points to be plotted as squares until another plotting control function is encountered. This command overrides keyboard selection of symbols during plotting.

TRI PLOT A TRIANGLE

Causes all subsequent points to be plotted as triangles until another plotting control function is encountered. This command overrides keyboard selection of symbols during plotting.

Example: The next subroutine plots an AFM diagram using symbols defined as follows.

```
triangle        MGO/MGO+FEO > .6
square          .5 < MGO/MGO+FEO <= .6
dot             .4 < MGO/MGO+FEO <= .5
point           MGO/MGO+FEO <=.4

 1. d  MGO       15. o  PNT
 2. d  MGO       16. d  NA2O
 3. d  FEO       17. d  K2O
 4. o  +         18. o  +
 5. o  /         19. o  =A
 6. o  TRI       20. d  FE2O3
 7. o  >.6       21. c  .9
 8. o  GT16      22. o  *
 9. o  SQU       23. d  FEO
10. o  >.5       24. o  +
11. o  GT16      25. o  =B
12. o  DOT       26. d  MGO
13. o  >.4       27. o  =C
14. o  GT16
```

SELECT COLOR

COLR# SET COLOR TO #

Sets the symbol color for plotting with an EGA monitor. All symbols will be plotted in the indicated color until another color statement is encountered.

COLR# does not work in creating histograms.

The COLR# overrides keyboard selection of colors during plotting.

may range from 1 to 7. These values correspond to the following colors:

Number	Color
1	blue
2	green
3	cyan
4	red
5	magenta
6	brown
7	white

Example: the following subroutine will create a K2O vs. SIO2 plot. Analyses reflecting strong carbonate replacement (CAO >12) will be plotted in red (color 4); all others will be plotted in green (color 2).

```
1.  o  COLR2     set default color to green
2.  d  CAO
3.  o  >12
4.  o  COLR4     if CAO > 12 use red instead
5.  d  SIO2
6.  o  =X
7.  d  K2O
8.  o  =Y
```

TRACE ON, TRACE OFF

TRON TRACE ON

Turns on the trace function, which causes the program to list on the printer the line number, the contents of the top of the stack and the operation at each step of the subroutine during program operation.

TRON may be embedded at any place in a subroutine. Used in conjuction with TROFF, blocks of the subroutine may be isolated for debugging.

Example:

```
1. d  FEO
2. o  SM01
3. o  TRON
4. d  MGO
5. o  SM01+
6. o  TROFF
7. o  RM01/
8. o  =X
```

Line 3 turns the trace function on. Line 6 turns the trace function off.

CAUTION: TRON requires an attached printer. Failure to turn the printer on will result in a fatal error.

TROFF TRACE OFF

Turns the trace function off. See TRON

S E C T I O N 7

D A T A P L O T

Contents

INTRODUCTION

DATAPLOT allows the user to create customized orthogonal (X vs. Y), ternary and histogram plots. Graphs are based on subroutines that may be created and edited by the user. Data may be plotted using combinations of four symbols (dots, squares, triangles, or single pixels). Systems that have EGAs (enhanced-color adapters) may also utilize seven colors. Graphs are displayed on the monitor where they may be modified with CAD (computer-assisted design) functions. Users who have a Mouse System mouse may use the mouse to place labels and draw. Finalized graphs may be stored on disk and used as templates for future plots. Graphs may be reproduced on IBM or Epson dot-matrix printers.

This section describes the operation of DATAPLOT. Programming in STACK, the language required to create new graphs with DATAPLOT, is described in Sections 5 and 6.

MAIN MENU

DATAPLOT is run from a main menu of options. This menu is displayed following loading of the program. The menu will appear similar to the following.

*** DATAPLOT ***

D	Directory	E	Edit Subroutine
S	Sample Code File	C	Construct Plot
M	Monitor Type	T	Recall Template
P	Plot Type	R	Recall Image

ESC Exit Program

Monitor: CGA
Plot Type: Orthogonal
Sample-Code File: A:CODES
Current Data File:
Current Subroutine:

The bottom of the screen displays (1) the type of monitor that will be used for creating graphs, (2) the type of graph that will be created (orthogonal, ternary or histogram), (3) the location and name of the current sample-code file, (4) the location and name of the current data file, and (5) the location and name of the current STACK subroutine.

The top of the menu displays nine program options that are selected with single key strokes. The options listed below and described in detail in subsequent sections.

C	Construct Plot	Allows new graph to be created.
D	Directory	Displays directory of any drive.
E	Edit Subroutine	Allows editing of old subroutine or creation of new subroutine with the STACK subroutine editor.
ESC	Exit Program	Returns to DOS.
M	Monitor Type	Switches from CGA, EGA and monochrome modes.
P	Plot Type	Switches between orthogonal, ternary and histogram plotting modes.
R	Recall Image	Allows a stored graph to be recalled for display on the screen.
S	Sample-Code File	Allows current sample-code file to be changed.
T	Recall Template	Allows graph to be created using a previously created template.

D DIRECTORY

DESCRIPTION

This option allows the user to inspect the directory of any drive.

OPERATION

1) Strike "D" while the main menu is displayed.
2) The screen will clear and the following question will appear at the bottom of the screen.

   Directory Information Drive (A or B)?

3) Key in the letter of the desired drive. Any existing drive (A, B, C, etc.) may be used. Do not use a colon.
4) Strike RETURN.
5) The program will display the requested directory.
6) Strike any key to return to the main menu.

S SAMPLE-CODE FILE

DESCRIPTION

This option allows the current sample-code file to be changed. This is necessary if the user wants to use a specialized sample-code file on another disk.

OPERATION

1) Strike "S" while the main menu is displayed.
2) The screen will clear and the following question will appear.

   Sample Code File?

3) Type the name of the sample-code file (e.g. DUMMY.ABC) and strike RETURN. Striking RETURN without typing a name will cause the program to use the name of the most recently used sample-code file. The program will then ask for the drive location of the sample-code file.

```
Sample Code File? DUMMY.ABC          Drive (A or B)?
```

Type "DIR" for the file name to display the directory of the drive indicated in the next step.

4) Type the letter of the drive (A, B, C, etc.) followed by a RETURN.

Typing RETURN without a letter will cause the program to use the drive of the most recently used sample-code file. Typing two or more letters (e.g. "XX") followed by a RETURN will return the program to the main menu.

5) The program returns to the main menu. Note that the name of the current sample-code file has changed.

E EDIT SUBROUTINE

DESCRIPTION

This options allows the user to create a new STACK subroutine or to recall an existing subroutine for editing. The operation of the STACK subroutine editor is described in Section 5. STACK statements are described in Section 6.

OPERATION

1) Strike "E" while the main menu is displayed.

2) The screen will clear and then present the following options.

*** STACK EDITOR ***

Your choices are as follows:

N	Make a NEW subroutine
E	Edit an old subroutine
ESC	Exit to dataplot

3) Strike "N" to write a new subroutine from scratch. Proceed to step 8.
Strike "E" to edit an existing subroutine. Proceed to step 4.
Strike ESC to return to the main menu.

4) The following question will appear.

Subroutine name?

5) Type the name of the subroutine (e.g. AFM) and strike RETURN.

   Striking RETURN without typing a name will cause the program to use the name of the most recently used subroutine.

   DO NOT PUT AN EXTENSION ON THE NAME. DATAPLOT WILL AUTOMATICALLY ADD THE EXTENSION ".SUB".

   Type "DIR" for the subroutine name to display the directory of the drive indicated in the next step.

6) The program asks for the drive location of the subroutine.

   Subroutine name? AFM Drive (A or B)?

7) Type the letter of the drive (A, B, C, etc.) followed by a RETURN.

   Typing RETURN without a letter will cause the program to use the drive of the most recently used subroutine. Typing two or more letters (e.g. "XX") followed by a RETURN will return the program to the main menu.

8) The screen will clear and then display the subroutine editor. The editor will be a "blank slate" if the user selected "N" in step 3. The editor will contain the specified subroutine if the user selected "E" in step 3. The subroutine may be modified and stored under the same name or under a new name. (see SUBROUTINE EDITOR).

10) The program returns to the main menu on exiting the subroutine editor.

C CONSTRUCT PLOT

DESCRIPTION

This option allows a graph to be constructed from scratch on the monitor. The type of graph (orthogonal, ternary or histogram) is determined by the plot type indicated on the main menu. The user may edit the image on the screen to create a customized graph. Editing options include: (1) changing aspect ratio of the axes; (2) changing graph boundaries and tick marks; (3) moving the graph on the screen; (4) placing labels; (5) drawing lines or pixels; and (6) block erasing, copying and moving. The graph may be stored on disk with or without data points. A graph that is stored without data points may be used later as a template for creating new graphs. The graph may be reproduced on an attached dot-matrix printer.

OPERATION

1) While the main menu is displayed, select the plot type by striking "P" then strike "C".

2) The screen clears and the following question appears.

   Subroutine name?

3) Type the name of the subroutine (e.g. AFM) and strike RETURN.

   Striking RETURN without typing a name will cause the program to use the name of the current subroutine.

   DO NOT PUT AN EXTENsION ON THE NAME. DATAPLOT WILL AUTOMATICALLY ADD THE EXTENSION ".SUB".

   Type "DIR" for the subroutine name to display the directory of the drive indicated in the next step.

4) The program asks for the drive location of the subroutine.

    Subroutine name? AFM Drive (A or B)?

5) Type the letter of the drive (A, B, C, etc.) followed by a RETURN.

   Typing RETURN without a letter will cause the program to use the drive of the current subroutine. Typing two or more letters (e.g. "XX") followed by a RETURN will return the program to the main menu.

6) The following question appears.

    Data file name?

7) Type the name of the data file (e.g. MORB) and strike RETURN.

   Striking RETURN without typing a name will cause the program to use the name of the current data file.

   Type "DIR" for the data file name to display the directory of the drive indicated in the next step.

8) The program asks for the drive location of the data file.

    Data file name? MORB Drive (A or B)?

9) Type the letter of the drive (A, B, C, etc.) followed by a RETURN.

   Typing RETURN without a letter will cause the program to use the drive of the current data file. Typing two or more letters (e.g. "XX") followed by a RETURN will return the program to the main menu.

10) The screen clears and displays a summary of the data file. The screen will look similar to the following.

    There are 26 samples in the file.

Sample Code	# if samples
WRN	18
WR+	8

    Use this file (Y/N)?

11) Strike "Y" to proceed to step 12.
Strike "N" to return to step 6.

12) The program asks:

List values (Y/N)?

13) Strike "Y" to list results of the STACK calculations on an attached printer. The printer must be on or an error will result.
Strike "N" to avoid listing results.

14) The program asks:

Plot all data (Y/N)?

15) Strike "Y" to plot all data. The program skips to step 28.
Strike "N" to proceed to step 16.

16) The program asks:

Plot all of one code (Y/N)?

17) Strike "Y" to skip to step 26.
Strike "N" to proceed to step 18.

18) The current plotting color is indicated in the lower left corner for systems having EGA capability. Select the desired color with the "+" and "-" keys. Strike RETURN to lock that color in.

19) The current plotting symbol is indicated in the lower left corner. Select the desired symbol with the "+" and "-" keys. Strike RETURN to lock that symbol in.

20) The screen clears and a summary of the file contents is presented in three columns.

The left column displays the entry number of each sample in the file. The middle column displays the sample code of each sample. The right column displays the name of each sample.

21) To include a sample in the plot, first move the arrow to it using the cursor control keys, ↑ ↓ . These keys will scroll through the entire file once the arrow has reached the top or bottom of the screen.

22) Strike "X" to include the sample in the plot.

23) Continue selecting samples until done.

24) Strike ESC to quit selecting samples from the current file.

25) Skip to step 31.

26) The program asks:

Sample Code?

27) Type one of the sample codes displayed on the screen and strike RETURN..

28) The current plotting color is indicated in the lower left corner for systems having EGA capability. Select the desired color with the "+" and "-" keys. Strike RETURN to lock that color in.

29) The current plotting symbol is indicated in the lower left corner. Select the desired symbol with the "+" and "-" keys. Strike RETURN to lock that symbol in.

30) The program works through the indicated samples.

The number of the sample being used in the calculation is displayed on the screen.

31) The screen clears and asks:

Recall more data (Y/N)?

32) Strike "Y" to return to step 6.
Strike "N" to continue to step 33.

33) The program presents different questions to the user depending on whether an orthogonal, ternary or histogram plot is being constructed. These are discussed separately below.

Ternary Plot:

The program proceeds directly to graphics and constructs a ternary plot.

Orthogonal Plot:

The user must establish the range of the plot by identifying the lowest and highest X and Y values to be plotted. These values will be used to draw a box on the screen. Data that plot outside the box will not appear on the screen.

The user must establish tick-mark spacing for both axes. The tick-mark spacing must always be greater than zero. The best results are obtained with tick-mark spacings that result in 5 to 20 tick marks on each axis.

The following steps establish graph limits and tick-mark spacing.

a) The program displays a screen similar to the following. The minimum and maximum values of X and Y calculated by the STACK subroutine are reported for reference.

```
Minimum X value is 48.29
Maximum X value is 76.12

Minimum Y value is .0485
Maximum Y value is 57

Input lowest X Coordinate?
```

b) Type the desired lower limit for the X-axis followed by a RETURN. This entry establishes the left side of the graph.

c) The program asks:

```
highest X Coordinate?
```

d) Type the desired upper limit for the X-axis followed by a RETURN. This entry establishes the right side of the graph.

e) The program asks:

```
Input lowest Y Coordinate?
```

f) Type the desired lower limit for the Y-axis followed by a RETURN. This entry establishes the bottom of the graph.

g) The program asks:

```
highest Y Coordinate?
```

h) Type the desired upper limit for the Y-axis followed by a RETURN. This entry establishes the top of the graph.

i) The program asks:

Tick-mark spacing for X-axis?

j) Type the tick-mark spacing followed by a RETURN.

k) The program asks:

Tick-mark spacing for Y-axis?

l) Type the tick-mark spacing followed by a RETURN.

m) The program draws the graph.

Histogram Plot:

The user must establish the range of the plot by identifying the lowest and highest X values to be plotted and the maximum column height of the histogram. These values will be used to draw a box on the screen. Data that plot with X-values outside the box will not appear on the screen. Columns that exceed the maximum column height will be truncated.

The user must establish the column width for calculating the histogram. The tick-mark spacing of the X-axis will be equal to this value.

The user must establish the the tick-mark spacing for the Y axis. The tick-mark spacing should be a positive integer. The best results are obtained with tick-mark spacings that result in 5 to 20 tick marks.

The following steps establish graph limits, column width and tick-mark spacing.

a) The program displays a screen similar to the following. The minimum and maximum values of X calculated by the STACK subroutine are reported for reference.

Minimum X value is 48.29
Maximum X value is 70.78

Input lowest X Coordinate?

b) Type the desired lower limit for the X-axis followed by a RETURN. This entry establishes the left side of the graph.

c) The program asks:

highest X Coordinate?

d) Type the desired upper limit for the X-axis followed by a RETURN. This entry establishes the right side of the graph.

e) The program asks:

Column width for X-axis?

f) Type the width desired for the histogram columns followed by a RETURN. The program expects the column width to be expressed in the same units as the upper and lower coordinates.

g) The program displays something similar to the following.

The highest column is 8

Input column height?

h) Type a maximum column height followed by a RETURN. This number should be larger than the highest column. This entry will be used to scale the Y-axis.

i) The program asks:

Tick-mark spacing for Y-axis?

j) Type a positive integer and strike RETURN.

k) The program displays the histogram.

T RECALL TEMPLATE

DESCRIPTION

This option allows the user to recall a preconstructed template for graphing data. Templates are useful for rapid creation of plots that have standard formats. Templates may be created by following the steps outlined under the S-Save option of Editing Graphs; a "Y" response is required for step 10 to create a template.

A template consists of three files. (1) A image file, which has an extension ".SCR", contains the image to be displayed on the screen. (2) A parameter file, which has an extension ".PAR", contains crucial information such as minimum and maximum coordinate values, their pixel locations on the screen, and the drive and name of the relevant STACK subroutine. (3) A subroutine file contains the STACK subroutine that will be used to plot more points. These files must have been created previously (see Creating a Template) and they must be available on disk or the template option will not work.

OPERATION

1) While the main menu is displayed, select the plot type by striking "P" then strike "T".

2) The screen clears and the following question appears.

   Template name?

3) Type the name of the template (e.g. AFM) and strike RETURN.

   Striking RETURN without typing a name will cause the program to use the name of the current subroutine.

   DO NOT PUT AN EXTENSION ON THE NAME. DATAPLOT WILL AUTOMATICALLY ADD EXTENSIONS.

   Type "DIR" for the template name to display the directory of the drive indicated in the next step.

4) The program asks for the drive location of the template.

   Template name? AFM Drive (A or B)?

5) Type the letter of the drive (A, B, C, etc.) followed by a RETURN.

   Typing RETURN without a letter will cause the program to use the drive of the current subroutine. Typing ESC followed by a RETURN will return the program to the main menu.

6) The following question appears.

   Data file name?

7) Type the name of the data file (e.g. MORB) and strike RETURN.

   Striking RETURN without typing a name will cause the program to use the name of the current data file.

   Type "DIR" for the data file name to display the directory of the drive indicated in the next step.

8) The program asks for the drive location of the data file.

   Data file name? MORB Drive (A or B)?

9) Type the letter of the drive (A, B, C, etc.) followed by a RETURN.

   Typing RETURN without a letter will cause the program to use the drive of the current data file. Typing "ESC" followed by a RETURN will return the program to the main menu.

10) The screen clears and displays a summary of the data file. The screen will look similar to the following.

    There are 18 samples in the file.

    Sample Code # if samples

    WRN 18

    Use this file (Y/N)?

11) Strike "Y" to proceed to step 12.
    Strike "N" to return to step 6.

12) The program asks:

    List values (Y/N)?

13) Strike "Y" to list results of the STACK calculations on an attached printer. The printer must be on or an error will result.
Strike "N" to avoid listing results.

14) The program asks:

Plot all data (Y/N)?

15) Strike "Y" to plot all data. The program skips to step 28.
Strike "N" to proceed to step 16.

16) The program asks:

Plot all of one code (Y/N)?

17) Strike "Y" to skip to step 26.
Strike "N" to proceed to step 18.

18) The current plotting color is indicated in the lower left corner for systems having EGA capability. Select the desired color with the "+" and "-" keys. Strike RETURN to lock that color in.

19) The current plotting symbol is indicated in the lower left corner. Select the desired symbol with the "+" and "-" keys. Strike RETURN to lock that symbol in.

20) The screen clears and a summary of the file contents is presented in three columns.

The left column displays the entry number of each sample in the file. The middle column displays the sample code of each sample. The right column displays the name of each sample.

21) To include a sample in the plot, first move the arrow to it using the cursor-control keys, ↑ ↓ . These keys will scroll through the entire file once the arrow has reached the top or bottom of the screen.

22) Strike "X" to include the sample in the plot.

23) Continue selecting samples until done.

24) Strike ESC to quit selecting samples from the current file.

25) Skip to step 31.

26) The program asks:

Sample Code?

27) Type one of the sample codes displayed on the screen and strike RETURN..

28) The current plotting color is indicated in the lower left corner for systems having EGA capability. Select the desired color with the "+" and "-" keys. Strike RETURN to lock that color in.

29) The current plotting symbol is indicated in the lower left corner. Select the desired symbol with the "+" and "-" keys. Strike RETURN to lock that symbol in.

30) The program works through the indicated samples.

The number of the sample being used in the calculation is displayed on the screen.

31) The screen clears and asks:

Recall more data (Y/N)?

32) Strike "Y" to return to step 6.
Strike "N" to continue to step 33.

33) The program presents the graph on the screen.

R RECALL IMAGE

DESCRIPTION

This option allows the user to recall a previously stored graph and display it on the monitor. This is useful for (1) inspection of a graph, (2) editing a graph, (3) photographing a graph from the monitor, or (4) reproducing a graph on a printer.

OPERATION

A graph may be recalled from disk by the following steps.

1) Strike "R" when the main menu is displayed.

2) The screen clears and the following question appears.

   Image file?

3) Type the name of the image file (e.g. AFM) and strike RETURN.

   DO NOT PUT AN EXTENSION ON THE NAME. DATAPLOT WILL AUTOMATICALLY ADD THE EXTENSION ".SCR".

   Type "DIR" for the template name to display the directory of the drive indicated in the next step.

4) The program asks for the drive location of the image file.

   Image file? AFM Drive (A or B)?

5) Type the letter of the drive (A, B, C, etc.) followed by a RETURN.

   Typing two or more letters (e.g. "XX") followed by a RETURN will return the program to the main menu.

The following options are presented at the bottom of the screen after the graph has been displayed. An option is selected with a single key stroke.

P--Print	Striking "P" erases the menu line. The image on the screen is reproduced on an attached dot-matrix printer. Strike any key to display the menu line.
R--Camera Ready	Striking "R" erases the menu line. The image on the screen may be photographed. Strike any key to display the menu line.
L--Label	Striking "L" allows addition of labels to the graph. See EDITING GRAPHS.
+--Cursor	Striking "+" displays the graphics cursor, which may be used to modify the graph. See EDITING GRAPHS.
S--Save	Striking "S" causes the program to store the graph as it appears on the screen. This option begins by requesting the file name and drive for storing the graph.
ESC--Menu	Returns the program to the main menu.

ESC EXIT PROGRAM

DESCRIPTION

This option returns the computer to DOS and calls the program MENU.

OPERATION

Strike the ESC key while the main DATAPLOT menu is displayed.

EDITING GRAPHS

DATAPLOT allows manipulation of graphics on the screen. The available options are presented in a menu line at the bottom of the graphics screen. The options are selected with single key strokes.

The screen-editing options are described below. Some of these options are slightly different for orthogonal, ternary and histogram plots and some options may be operated with a Mouse Systems or compatible mouse.

P-PRINT

Plots: orthogonal, ternary, histgram.

Description:

This option erases the menu line and reproduces the image on an attached dot-matrix printer. WARNING: if the printer is off, an error occurs, the program returns to the main menu and the user could loose the graph.

Operation:

1) Make certain that the printer is on.
2) Strike "P".
3) The program presents printer options.
4) Select the appropriate printer.
5) The menu line will be displayed when printing is complete.

S-SAVE

Plots: orthogonal, ternary, histogram

Description:

This option allows the user to store the image displayed on the screen to a disk. The stored image may be recalled at a later time for editing, photography or printing. A stored image may also be used as a template for plotting additional data.

Operation:

1) Strike "S"

2) The program asks:

File name?

3) Type the file name (e.g. TEST) to be used to store the image. Strike RETURN.

DO NOT USE AN EXTENSION!! DATAPLOT WILL AUTOMATICALLY APPEND THE EXTENSION ".SCR".

4) The program asks:

Drive?

5) Type the drive letter (A, B, C, etc.) followed by a RETURN.

6) If the graph was created using "T Recall Template" or "R Recall Image" options then skip to step 11.

7) The program presents the following options.

(1) Store Axes + Points (2) Store Axes Only

8) Strike "1" to store the graph exactly as it appears on the screen. Strike "2" to strip data points off the graph before storing it.

9) The program asks:

Store axis definitions (Y/N)?

10) A "Y" response stores a parameter file that is necessary to create a template.
A "N" response does not store a parameter file.

X/Y-ASPECT

Plots: orthogonal, histogram

Description:

Striking either "X" or "Y" allows the user to adjust the length of the X-axis or Y-axis, respectively. The program requests the percentage change in length on the screen. Negative numbers cause the relevant axis to

shrink and positive numbers cause it to expand. This option clears the screen before redrawing the modified graph. As a result, labels (see below) will be lost. The graph cannot be expanded beyond the limits of the screen; to "zoom" part of the graph, use the "T-Redraw" option.

Operation:

To change the length of the X axis:

1) Strike "X"

2) The program asks:

Change X-axis length by what percentage (+ or -)?

3) Type the percentage with a minus sign if applicable (e.g. -30) and strike RETURN. A plus sign is optional.

4) The program redraws the graph and displays the menu line.

To change the length of the Y axis, follow the above steps but strike "Y" in step 1.

M-MOVE

Plot: orthogonal, ternary, histogram

Description:

This option allows the user to reposition the graph on the screen. The program requests the number of pixels to move in the X (horizontal) direction and then the number of pixils to move in the Y (vertical) direction. A carriage return is the equivalent of zero or no movement along that direction. Numbers should be integers. Negative input moves the graph to the left or down. Positive input moves the graph up or right.

The program prevents the user from moving any part of the graph off of the screen.

Operation:

1) Strike "M"

2) The program asks:

   Horizontal shift in pixels?

3) Type the number of pixels that the graph will be moved horizontally. The horizontal axis of the screen is 640 pixels wide.

4) The program asks:

   Vertical shift in pixels?

5) Type the number of pixels that the graph will be moved vertically. The vertical axis is 200 pixels in CGA mode and 350 pixels in EGA mode.

6) The program redraws the graph and displays the menu line.

L-LABEL

Plot: othogonal, ternary, histogram

Description:

This option allows the user to write alpha-numeric labels for placement on the screen. Labels may be written using all keyboard characters. Labels may be moved to any place on the screen. Labels may be moved across existing graph, data and labels without affecting them.

Operation:

To put a label on the screen.

1) Strike "L" when the menu line is displayed.

2) If the monitor is a normal color graphics monitor, skip to step 4.

   If the monitor is using enhanced graphics, the current color will be displayed in the lower left corner.

3) Select a new color, if desired, with the "+" and "-" keys and lock the color in by striking RETURN.

4) The program asks:

Label?

5) Type the label followed by a RETURN.

The length of the label should not exceed 30 characters.

6) Position the label using the cursor control keys (keys arrows with arrows).

Strike "F" to move the label 10 pixels at a time.

Strike "S" to move the label 1 pixel at a time.

A label may be positioned using a mouse.

7) Strike "E" to emplace on the screen when the location is satisfactory.

8) The program displays the menu line.

To delete an existing label:

1) Strike "L" when the menu line is displayed.

2) If the monitor is a normal color graphics monitor, skip to step 4.
If the monitor is using enhanced graphics, the current color will be displayed in the lower left corner.

3) Select the exact color of the label to delete with the "+" and "-" keys and lock the color in by striking RETURN.

4) The program asks:

Label?

5) Type the label to be deleted exactly as it appears on the screen and strike RETURN.

The length of the label should not exceed 30 characters.

6) Position the movable label using the cursor control keys (keys arrows with arrows) and(or) Mouse Systems mouse so that the movable label eclipses the label to be deleted.

7) Strike "D" to delete both the movable and original labels from the screen.

8) The program displays the menu line.

(For an alternative method, see the Block erase option.)

T-REDRAW

Plot: orthogonal, histogram
Not functional for graphs created with templates.

Description:

This option allows the user to alter the range of the plot and modify the tick marks. The screen clears the program requests (1) new coordinate limits and (2) new tick-mark spacings. WARNING: all labels will be erased when this option is selected.

Operation:

1) Strike "T" when the menu line is displayed.

2) The screen clears. Follow the steps described under C Construct Plot beginning with step 33.

Q-QUIT

Plot: orthogonal, ternary, histogram

Description: This option returns to the main menu.

Operation:

1) Strike "Q" when the menu line is displayed.

2) The menu line is erased and the program asks:

Clear screen (Y/N)?

3) A "N" response continues to display the graph.
A "Y" response returns the program to the main menu.

A-ADD

Plot: orthogonal, ternary, histogram

Description:

This option allows the user to select additional data files so that more data points can be added to the graph. WARNING: this option clears the screen and causes all labels to be erased.

Operation:

1) Strike "A" when the menu line is displayed.

2) The screen clears.

If the graph was created with the C Construct Plot option, follow the steps described under that option beginning with step 6.

If the graph was created with the T Template option, follow the steps described under that option beginning with step 3.

+-CURSOR

Plot: orthogonal, ternary, histogram

Description:

Striking "+" causes a cursor that is shaped like a cross to be displayed in the upper left corner of the screen. The cursor may be moved with the cursor-control keys and(or) mouse. Cursor speed is controlled with the "F" and "S" keys.

The cursor form and function may be changed by striking the space bar. With each toggle of the space bar the cursor alternates between a cross, a dot, a square, and a triangle. The cursor is in drafting mode when the cross is displayed. The cursor is in point identification mode when the other symbols are displayed.

Drafting mode:

In the drafting mode, the following menu is displayed at the bottom of the screen.

B-Block +/- pixel L-line Esc-done C-Color

The C-Color option will not be displayed for systems operating in the CGA mode. Single key strokes using these characters produces the following results.

B-Block: This option defines one corner of a rectangle at the position of the cursor. Cursor may be moved to select the opposite corner of the rectangle. At this point, striking "E" will erase the contents of the rectangle. Striking "M" will allow the contents of the rectangle to be moved using the same keys as the "L-Label" option. Striking "C" will allow the contents of the rectangle to be moved but will leave the rectangle unchanged. There is no limit to the size of the block that may be used with "E", but "M" and "C" are memory limited. If a block is to big to be moved or copied, the user may perform the task with several smaller blocks. Striking ESC terminates the block option.

+/-: Striking a "+" will place a pixel on the screen at the center of the cursor. Striking a "-" will remove a pixel on the screen at the center of the cursor.

L-Line: Striking "L" defines the starting point of a line. The curser is then moved to the end point of the line and "L" is struck again to draw the line.

Esc-done: Striking ESC returns the program to the graphics menu.

C-Color: Striking "C" displays the current color for drafting in the lower left hand corner of the screen. Select the desired color using the "+" and "-" keys and lock the color in by striking RETURN. Lines and pixels will be drawn with this color until a new color is selected.

Mouse: If a mouse is attached, moving the mouse with the left button depressed will draw a continuous curve.

Point-identification mode:

In the point-identification mode, the following menu is displayed at the bottom of the screen.

F/S-speed X-select N-name Esc-done

The "F" and "S" keys control the speed of the cursor. The cursor movement may be controlled with the cursor-control keys and(or) mouse. The shape of the cursor may be changed by toggling the space bar.

To identify a point, move the cursor so that it eclipses the point of interest and type "X" to select the point. The program prints the sample number in the lower left corner and gives the user the option of deleting the point, viewing the corresponding analysis or continuing. If the user elects to view the point, a window is opened in the upper left corner of the screen and the analysis may be viewed three entries at a time. The entire sample contents may be viewed by scrolling with the "|" and "|" cursor-control keys.

Striking "N" causes the program to query for a sample number and search for the corresponding point. If the point falls on the screen, the program makes the point flash and gives the user the opportunity to delete the point, view the corresponding analysis, or continue.

Only the contents of the most recently recalled data file may be viewed. Attempts to view other data will return the message:

Function unavailable.

Mouse: If a mouse is attached, depressing any button on the mouse provides greater precision for placing the cursor.

D-DIMEN

Plot: ternary
Does not work for graphs created from templates.

Description:

This option allows the user to change the size of a triangular plot. WARNING: this option clears the screen and labels will be lost.

Operation:

1) Strike "D" when the menu line is displayed.

2) The screen asks:

Change limbs by what percent?

3) Enter a number and strike RETURN. A positive number will increase the size of the triangle. A negative number will decrease the size of the triangle.

4) The triangle is redrawn and the menu line is displayed.

R-ASPECT

Plot: ternary
Does not work for graphs created from templates.

Description:

This option allows the user to change the aspect ratio of the triangle. This is particularly useful because different screens and printers have different horizontal and vertical pixel spacing. A triangle that appears equilateral on the screen may not be reproduced as equilateral on a printer, and visa versa. Some experimentation will be necessary to determine appropriate aspect values for each screen and printer. WARNING: this option clears the screen and labels will be lost.

Operation:

1) Strike "R" when the menu line is displayed.

2) The program reports the current aspect ratio (e.g. .349) and asks for a new ratio as follows.

Current aspect ratio: 0.349 New aspect ratio?

3) Type the new aspect ratio and strike RETURN.

   The aspect ratio must be greater than zero. Increasing the aspect ratio makes the triangle taller and thinner.

4) The program redraws the triangle and displays the menu line.

OPERATION TIPS

1. Do not label a graph until the position, size, aspect ratio, etc. have been finalized.

2. DATAPLOT can plot 500 data points at one time. However, more than 500 point may be plotted by storing a template with data points and then using that template to plot additional data.

3. The image file of a template may be modified at any time without interfering with the operation of the "template" option. This means that the image file may be edited with the "recall image" option, stored with the same name, and used in modified form to construct plots.

4. Any CAD program that reads and writes "BLOADED" screens may be used to edit the CGA image files.

5. Storing an image without data points will leave holes in labels or lines created with the cursor if those features overlap data points. To avoid this, data-free plots may be created by plotting the contents of a nonexistent or empty data file.

6. Sometimes, pixels are offset slightly during printing of a graph. This tends to happen in the direction of printer-head motion when a complex array of pixels is encountered. Some letters are particularly troublesome, such as "V" or "M" or "N". The offset is normally restricted to one line unless another complex pixel array is encountered. This may be corrected in most cases by moving the offending pixel array a few pixels on the screen and reprinting. This is the best way to correct labels. If the offending array cannot be shifted (for reasons of accuracy, etc.), the problem can often be corrected by adding or deleting a key pixel on the screen with the cursor and reprinting.

S E C T I O N 8

D A T A L I S T

Contents

INTRODUCTION

DATALIST allows the user to print the contents of files in tabular form to (1) dot matrix or letter quality printers, (2) ASCII files on disk, or (3) the computer display. The program prints the entire contents of the specified data file. The output is organized so that numbers are printed in columns, and all entries are segregated by code. The output begins each page with the file name and current code being printed.

DATALIST may be run in two modes. In the default mode, the only input required is (1) the sample-code, (2) the data file and (3) whether or not to print standard deviations. Alternatively, the user may customize the results by changing the default options in two menus.

On initialization, DATALIST displays the following screen.

```
                    *** DATALIST ***

        >>  Sample Code File:          A:DATABASE

            Printer Control Code:      CHR$(15)

            Format:                    Default

            Data File:

            PREVIEW

            SEND TO PRINTER

            SEND TO DISK

            EXIT PROGRAM
```

A cursor is located left of the options and may be moved up or down using the cursor control keys. An option may be selected by striking RETURN when the cursor is pointing at the desired option. Operation in the default mode is described below followed by a summary of each of the options.

DEFAULT MODE

Default mode assumes that the computer is communicating to an 80 column Epson or IBM dot matrix printer, or a dot matrix printer that uses the same control codes.

Four steps are required to print the contents of a file. (1) Check the current sample code file to see that it is appropriate. If it is not, strike RETURN with the cursor adjacent to "Sample Code File:" and respond to the queries with the appropriate file name and drive. (2) Move the cursor to "Printer Control Code:" and strike RETURN twice. (3) Move the cursor to "Data File:", strike return, and answer the queries to identify a data file. (4) Move the cursor to "SEND TO PRINTER" and strike RETURN.

When the file is printed, the program will prompt the user to list another file or exit the program.

OPTION DESCRIPTIONS

SAMPLE-CODE FILE

This option clears the screen and prompts the user to enter the sample-code file and then the drive. Enter the drive letter without a colon. Entering "DIR" for the sample code file will print a directory on the screen for the indicated drive and cause the program to prompt for the sample-code file and drive again.

PRINTER-CONTROL CODE

This option allows the user to input a printer-control code to be sent to the printer. When this option is selected, the program asks for the code to be sent. The user responds by typing the desired code exactly as it is listed in the printer manual and striking carriage return. The program immediately sends that code to the printer. For example, "CHR$(15)", the default code, is the Epson code for compressed type. This option may be used as many times as required prior to printing a file so that control of the printer can be relatively flexible.

The following rules apply to typing in control codes.

1. ESC and CHR$(27) are equivalent.
2. Do not exceed the available space for input.
3. Do not end the control code with a comma or semicolon

FORMAT

The format option clears the screen and displays a submenu that appears as follows.

```
>>  SUBTITLES
    Form Feed:                CHR$(12)
    Line Width:               132
    Print Sequence:           WT%/PPM/PPB/ALT
    Wt% Format:                  ###.##
    PPM Format:                 #######
    PPB Format:                 #######
    Alt Format:                ####.###
    bdl Message:                    bdl
    na  Message:
    STORE FORMAT
    RECALL FORMAT
```

Subtitles

This option allows the user to create subtitles for each of the four types of concentrations and/or floating point numbers. For example, numbers stored as weight percents could be subtitled "Weight Percent", "Major Elements", or "OXIDE ABUNDANCES (WT %)". Information stored as alternative floating point numbers could be subtitled "Normative Minerals" or whatever is appropriate.

Form Feed

Selecting this option allows the user to enter the appropriate printer control code to cause a form feed. The default, CHR$(12), works on most printers, and so normally this option can be ignored.

Line Width

This option allows the user to enter the desired number of characters per line in the printout. Permissible values depend on the characteristics of the printer. Note that the line width for many dot matrix printers with 80 column-wide carriages can be set to 120 or more if the printing is done in compressed mode (use the Printer Control Code option).

Print Sequence

This option allows the user to define the order in which data is printed. The program requests the user to input a new print sequence and the user responds with a string like one of the following:

WT%/PPM/ALT
PPM/PPB/WT%
ALT

where WT%, PPM, PPB and ALT correspond to weight percent concentrations, parts per million concentrations, parts per billion concentrations, and alternative floating point numbers, respectively. Concentration types that are not included will not be printed. The input must obey the following rules.

1. Use only capital letters.
2. Use only WT%, PPM, PPB or ALT.
3. Separate each concentration type with a "/".
4. Type the sequence without spaces.

An input error can be corrected by selecting the Print Sequence option again.

Wt. % Format

This option allows the user to specify the number of significant figures and position of the decimal point for the print out. The program requests the user to type in a new format that must be within two indicated field boundaries. The user may respond by typing the following keys.

space bar -- valid at the beginning of the field only

-- indicates a position in the number field

. -- indicates the location of the decimal point

carriage return -- enters the format into memory.

Errors may be corrected by selecting the Wt% Format option again.

PPM Format

This option allows the user to select a printing format for parts per million concentrations. The operation is identical to the Wt% Format option.

PPB Format

This option allows the user to select a printing format for parts per billion concentrations. The operation is identical to the Wt% Format option.

ALT Format

This option allows the user to select a printing format for alternative floating point numbers. The operation is identical to the Wt% Format option.

bdl Message

This option allows the user to select a message to be printed whenever a concentration is below detection limit (see Section 4). The default message is "bdl". To input a new message:

1. Move the cursor arrow to "bdl Message".
2. Strike RETURN.
3. The program asks:

   New bdl message?

4. Type the new message and strike RETURN.

na Message

This option allows the user to select a message to be printed whenever a concentration is stored as "not analyzed" (see Section 4). The default message is " " (i.e. empty space). To input a new message:

1. Move the cursor arrow to "na Message".
2. Strike RETURN.
3. The program asks:

   New na message?

4. Type the new message and strike RETURN.

Store format

This option allows the user to specify a file name followed by a target drive for storing all of the format information. Use of an extension in the file name is prohibited because DATALIST adds ".FMT" to the user's entry.

Recall format

This option allows the user to recall a printing format that has been stored previously.

DATA FILE

This option allows the user to recall a data file for printing. The program prompts the user for the file name and the source drive. Enter the file name using standard DOS conventions followed by a carriage return. Enter the drive letter followed by a carriage return; do not use a colon after the drive letter. Entering "DIR" for the file name will print the directory of the specified source drive.

PREVIEW

This option prints the contents of the specified file on the screen. The line width is necessarily set to 80 characters for this operation only. The program will print continuously to the screen as long as the space bar is depressed. The screen will "freeze" for inspection if the space bar is released. Striking ESC will interrupt printing and return the program to the menu.

SEND TO PRINTER

This option prints on a dot-matrix or letter-quality printer the content of the specified data file using the current format.

SEND TO DISK

This option writes the data table to an ASCII file on disk. The resulting file may be modified and merged into documents with commercial word-processing programs (such as WORDSTAR) that utilize ASCII files.

EXIT PROGRAM

This option causes the program to exit to DOS.

S E C T I O N 9

B L E N D

Contents

INTRODUCTION

BLEND is program that combines selected entries from two samples with different sample codes.

Consider the situation in which a data set consisting of major elements analyses was determined and stored with DATASAVE. At a later time, the same samples were analyzed for trace elements. It would be advantageous if the computer could combine these two data sets into a single file in which each sample contained a complete record of major and trace elements. BLEND was written expressly to meet such a need.

INITIALIZATION

The following steps will activate BLEND and display its menu.

1. With the DATASAVE menu displayed on the screen, strike "B" followed by striking RETURN.
2. The computer will display a screen similar to the following.

```
                    ***B L E N D ***

                D   Directory Information

                S   Select Sample Code File

                B   Blend Calculation

              ESC   Exit Program

          Current Sample Code File: CODES
```

DIRECTORY INFORMATION

The following steps will display the directory of any drive.

1. Strike "D" when the BLEND main menu is displayed.
2. The screen will clear and display the following.

   Directory Information Drive (A or B)?

3. Enter the desired drive as a single letter (A, B, C, etc.) and strike RETURN.
4. The directory will be displayed.
5. Hit any key to return to the main menu.

SELECT SAMPLE-CODE FILE

The following steps will allow the user to change the designated sample-code file. The directory of any drive may be inspected to assist in selecting a new sample-code file.

1. Strike "S" when the BLEND main menu is displayed.
2. The screen will clear and display the following.

   Sample-Code File?

3. Type the name of the new sample-code file followed by a RETURN. To display the directory of a drive, type "DIR" followed by a RETURN. Striking RETURN without typing a name or "DIR" will cause the program to use the current file name.
4. The Screen will display the following question.

   Drive (A or B)?

5. Type the letter of the drive (A, B, C, etc.) followed by a RETURN.

   Striking RETURN without typing a letter will cause the program to use the drive of the current sample-code file. Typing two or more letters (e.g. "XX") followed by a RETURN will return the program to the main menu without changing the sample-code file.

6. The program returns to the main menu.

BLEND CALCULATION

The following steps will blend the selected contents of two data files and store the result on disk.

1. Strike "B" when the BLEND main menu is displayed.
2. The screen will clear and display the following.

   IDENTIFY TARGET AND SOURCE FILES

   Name of target file?

3. Type the name of the file that will be used to store the blended data and strike RETURN. CAUTION: this process will overwrite a preexisting file with the same name.
4. The Screen will display the following question.

   Drive (A or B)?

5. Type the letter of the drive (A, B, C, etc.) followed by a RETURN.
6. The Screen will display the following question.

   Name of first source file?

7. Type the name of the first file to be used in the blend calculation followed by a RETURN. Type "DIR" in place of the file name to view the directory of any drive.
8. The Screen will display the following question.

   Drive (A or B)?

9. Type the letter of the drive (A, B, C, etc.) followed by a RETURN.

   Typing two or more letters (e.g. "XX") followed by a RETURN will return the program to the main menu.

10. The computer reads the first source file.
11. The Screen will display the following question.

    Name of second source file?

12. Type the name of the second file to be used in the blend calculation followed by a RETURN. Type "DIR" in place of the file name to view the directory of any drive.

13. The Screen will display the following question.

    Drive (A or B)?

14. Type the letter of the drive (A, B, C, etc.) followed by a RETURN.

    Typing two or more letters (e.g. "XX") followed by a RETURN will return the program to the main menu.

15. The computer reads the second source file.

16. The program displays a screen similar to the following.

    SELECT CODES

    Sample-Code File: A:CODES
    Codes: WRT NIG CPW AM PX OL FD OP WR+

    First Source File: B:FILEONE
    Codes: WRN

    Second Source File: B:FILETWO
    Codes: CPW

    The information allows the user to check the file names and also to see the sample codes used in each file.

17. The following question is displayed at the bottom of the screen.

    Target code?

18. Type a code (1-3 characters) followed by a RETURN. This entry must match one of the codes in the sample-code file.

19. The following question is displayed at the bottom of the screen.

    First source code?

20. Type a code (1-3 characters) followed by a RETURN. This entry must match a code present in both the sample-code file and the first source file.

21. The following question is displayed at the bottom of the screen.

    Second source code?

22. Type a code (1-3 characters) followed by a RETURN. This entry must match a code present in both the sample-code file and the second source file.

23. The screen displays three columns separated by vertical lines. Each column contains a file name in reverse video and the entry labels of the code to be used in the calculation. The first column represents the target file, the second column represents the first source file and the third column represents the second source file.

24. The following question is displayed at the right side of the screen.

OK (Y/N)?

25. Strike "Y" if the information in the columns is correct; the program will continue. Strike "N" if the information in the columns is incorrect; the program will return to step 16.

26. The follow menu of options appears on the right side of the screen.

Match	Entries
↑↓	Cursor
X	Select
S	Source
Esc	Done

SOURCE FILE
1

A cursor points to entry labels for the target file. This cursor may be moved with the arrow keys, "↑ ↓".

The source file indicated on the right side of the screen indicates the file currently active for matching entry labels. Striking "S" toggles the active source file between 1 and 2.

27. Move the cursor to a desired entry label in the target file.

28. Select the active source file, if necessary, by striking "S".

29. Strike "X". A small square will appear to the right of the entry label in the target file. Another small square will appear to the left of the matching entry label in the active source file. The computer beeps if a match cannot be made.

    Steps 27 through 29 inform the program which entries will be taken from source file 1 and which entries will be taken from source file 2.

    It is not necessary to match all of the entry labels in the target file. Those that are not matched will be assigned values of "not analyzed", which is effectively a zero value.

30. Repeat steps 27 through 29 to match as many entry labels in the target file as desired. Those that are not matched will be set equal to zero when the new file is created.

31. Strike the "ESC" key to continue.

32. The following question is displayed at the right side of the screen.

    OK (Y/N)?

33. Strike "Y" if the information in the columns is correct; the program will continue. Strike "N" if the information in the columns is incorrect; the program will return to step 16.

34. A screen similar to the following will be displayed. For a detailed discussion of the following operation, reference the section on sample number correlation.

Warning: Sample numbers in Source File 1 will be duplicated in the Target File!

A sample number has up to 12 characters: 1 2 3 4 5 6 7 8 9 A B C

Source File 1: First character

35. Type one of the characters (1 through C) that corresponds to the first character to be used in the match.

36. The program then prompts:

    Last character

37. The program displays in flashing characters that part of the sample name to be used in the match.

38. The program then displays the following

    Source File 2: First character

39. Type one of the characters (1 through C) that corresponds to the first character to be used in the match.

40. The program then prompts:

    Last character

41. The program displays in flashing characters that part of the sample name to be used in the match.

42. The following question is displayed at the bottom of the screen.

    Setup satisfactory (Y/N)?

43. Strike "Y" to proceed with the BLEND calculation. Strike "N" to return to step 34.

44. The BLEND calculation is performed, the data is stored in the target file, and the program returns to the main menu.

SAMPLE NUMBER CORRELATION

In order to blend data, the program must be able to recognize the corresponding analyses from the two source files. It is presumed that the same character strings will appear somewhere in corresponding analysis names in the source files. However, users commonly assign different prefixes or suffixes to the sample names so that simple matching of analysis names is not always possible. Consider the following example. Source file 1 might contain analyses with names in the format "240###MAJ" where ### range from 001 to 100. Similarly, source file 2 might contain analyses with names in the format "240###TRACE" where ### range from 001 to 100. The objective would be to combine the data from analyses that have matching characters for ###.

A sample name is composed of up to 12 characters which may be represented by "1 2 3 4 5 6 7 8 9 A B C". During sample number correlation, the user is asked to identify what part of each sample name is to be used in the correlation by selecting the first and last characters from this string. For the above example, a match could be made solely on the basis of the 4 through sixth characters. In this case, the first character would be "4" and the last character would be "6" for both source files. Alternatively, the match could have been made on the basis of the first through sixth characters. In this case, the first character would be a "1" and the last character would be a "6" for both source files. Using the seventh through twelfth characters in this case would prevent a match as the analysis names in the two files have different suffixes.

The same number of characters must be identified for both the source files or a match will not be made.

S E C T I O N 1 0

D A T A N O R M

Contents

INTRODUCTION

DATANORM is based on the CIPW and Niggli norm calculations presented in the Laboratory Handbook of Petrographic Techniques by C.S. Hutchison. The calculations are performed on anhydrous compositions normalized to 100 weight percent. Formula weights and mineral abbreviations are those of Hutchison. Mineral abbreviations are:

Q	Quartz	Ks	K-metasilicate	Tn	Titanite
C	Corundum	Di	Diopside	Pf	Perovskite
Or	Orthoclase	Wo	Wollastinite	Ru	Rutile
Ab	Albite	Hy	Hypersthene	Ap	Apatite
An	Anorthite	Ol	Olivine	Zr	Zircon
Lc	Leucite	Cs	Ca-orthosilicate	Fr	Fluorite
Ne	Nepheline	Mt	Magnetite	Pr	Pyrite
Kp	Kaliophite	Cm	Chromite	Cc	Calcite
Ac	Acmite	Il	Ilmenite	Hl	Halite
Ns	Na-metasilicate	He	Hematite	Th	Thernardite
				Nc	Cancrinite

MENU

When DATANORM is started, a menu of options is displayed on the screen that will be similar to the following.

```
****************** Menu ***************

        S   Sample Code File

        D   Directory Info

        I   Inkey Analysis

        R   Recall Data File

      Esc   Exit Program

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

  Current Sample Code File:  C:CODES
```

An option is selected by striking a single key. Do not follow the key stroke with a carriage return. Striking ESC will exit DATANORM and load the DATASAVE menu. The other options are described below.

S SAMPLE-CODE FILE

This option allows the current Sample-Code File to be changed. The screen is cleared and the user is prompted for the new file name and drive as follows.

Sample-Code File? XBASE Drive (A or B)? A

The Sample Code File (XBASE for example) is input using DOS conventions followed by a carriage return. The Drive is then input (A for example) using only a single letter followed by a carriage return. The name and drive are stored in the DOS work space and are not volatile unless the computer is rebooted.

D DIRECTORY INFO

This option allows the user to inspect the directory of any drive. The screen is cleared and the following question appears at the bottom of the screen.

Directory Information Drive (A or B)?

The program will display the requested directory until another key stroke (any key) returns operation to the main menu.

I INKEY ANALYSIS

This option allows the user to input a single analysis from the key board for calculation of normative minerals. The analysis may be edited before the calculation to correct input errors and edited after the norm calculation to study the effects of compositional changes.

DATA ENTRY

Abundances of elements/oxides are requested in the following order: SiO2, TiO2, Al2O3, Cr2O3, Fe2O3, FeO, MnO, MgO, NiO, CaO, Na2O, K2O, BSrO, SO3, S, P2O5, ZrO2, F, Cl, CO2, H2O. Data is entered by typing the abundance of the indicated element/oxide as a floating point number followed by a carriage return. A carriage return without numerical input is interpreted as zero.

DATA EDITING

Following data entry, the program prompts the user with "Modify input (Y/N)?". A "Y" response allows the user to correct as many element/oxide abundances as desired. The user must identify the element to modify by entering it from the keyboard. A carriage return without an entry terminates the editing routine.

FE++ AND CO2

Following data editing, the program reports Fe++/(total Fe) and the user has an opportunity to modify the value. If the bulk composition contains CO2, the user may choose to express the CO2 in terms of calcite, cancrinite or neither. In the latter case, CO2-bearing phases are not calculated.

RESULTS

The results of the CIPW and Niggli normative calculations are displayed in columnar format along with the original oxide weight percents, and the anhydrous formula percents. The results are too long to display on a single screen but the user may scroll through the norms using the "up" and "down" cursor control keys. Striking "F" and "S" control the the scroll speed. The user may print the results by striking "P". Striking "E" returns the program to the editing mode so that the bulk composition may be modified and a new norm calculation performed. Striking ESC returns the program to the main menu.

R RECALL DATA FILE

This option allows the user to recall DATASAVE files for calculation of normative minerals. Data recall begins by clearing the screen and requesting the file name and drive as follows.

Data File? DUMMY Drive (A or B)? B

The Data file (DUMMY in the example) is input using normal DOS conventions followed by a carriage return. The target drive is keyed in followed by a carriage return. Typing "DIR" in place of the file name will display the directory of the indicated drive.

A summary of the file contents is reported and the user is asked the following questions.

Use this file (Y/N)?

Run all the data (Y/N)?

If the user does not elect to use the current file, the program returns to the data recall subroutine and requests the name of a new file. If the user does not choose to run all the data, the program allows selection of individual analyses for calculations. If only one analysis is selected the program allows the user to edit the data prior to calculating a norm. If more than one analysis are selected or if the user elects to run all the data, the program asks the following series of questions prior to batch procession the data.

Do you want to fix Fe++/total Fe (Y/N)?

Do you want normative calcite (Y/N)?

Do you want normative cancrinite (Y/N)?

Do you want a print out (Y/N)?

Do you want to store on disk (Y/N)?

The user may store either CIPW or Niggli results on a disk. If the user opts to store the norm results on disk, the program then asks "Do you want to store CIPW (Y/N)?". A "Y" response causes the calculations to begin. A "N" response cause the program to ask "Do you want to store Niggli (Y/N)?". A "N" response at this point aborts disk storage. Following complete batch processing the programs prompts the user for the name of the target file for storing the norm results. The default target file has the same root as the data file but uses the extension ".NRM".

Data is written to disk using the sample code CPW (for CIPW norms) or NIG (for Niggli norms). Those codes must be present in the user's sample-code file if the data is to be recalled by a DATASAVE program at a later date. Both codes are defined in the sample-code file titles CODES and supplied on the program disk (disk 1). Both CPW and NIG have the following entry labels.

1. !Q	11. !KS	21. !TN	31. !NC
2. !C	12. !DI	22. !PF	32. !MG#
3. !OR	13. !WO	23. !RU	
4. !AB	14. !HY	24. !AP	
5. !AN	15. !OL	25. !ZR	
6. !LC	16. !CS	26. !FR	
7. !NE	17. !MT	27. !PR	
8. !KP	18. !CM	28. !CC	
9. !AC	19. !IL	29. !HL	
10. !NS	20. !HE	30. !TH	

The "!" prefix identifies the entries as general floating point numbers not to be confused with oxide weight percents. The last entry, "!MG#", is the Mg/(Mg+Fe) ratio of the mafic silicate minerals.

S E C T I O N 1 1

S P L I C E

Contents

INTRODUCTION

SPLICE is a program "shell" written in IBM BASICA that may be merged with a new application subroutine to access data from DATASAVE files. Geochemical programs commonly deal with a series of chemical analyses by performing a single repetitive calculation on each analysis in turn. SPLICE is designed to facilitate this type of programming, by (1) recalling DATASAVE data and sample-code files, (2) translating file contents into usable variables, (3) allowing selection of individual analyses for the calculation, and (4) controlling repetitive calculations. DATAPLOT, DATATERN, DATAHIST, and DATANORM were all written with SPLICE.

The user may utilize SPLICE by (1) writing a subroutine in IBM BASICA or a compatible version of BASIC, (2) storing the subroutine, and (3) running SPLDRIVE (re: splice drive) from BASICA. SPLDRIVE renumbers the subroutine, merges SPLICE with the user-generated subroutine, and stores the result as a program that may be run with little or no modification from BASICA. The resulting program may also be compiled with a BASIC compiler.

CREATING A PROGRAM WITH SPLICE

A new program may be written for the DATASAVE family by the following procedure.

1. Write a subroutine in BASICA to perform the desired calculations.

2. Store the subroutine in ASCII format.

3. Load BASICA (if it is not already loaded).

4. Be sure that the subroutine, SPLICE and SPLDRIVE are on the default drive. Type the following statement.

   Run "SPLDRIVE"

5. Follow the instructions on the screen. SPLDRIVE will renumber the user-generated subroutine, merge it to SPLICE and store the result. A new program has been created.

6. Modify the new program, if necessary, to correct for user-specific data formats (see Final Touches).

7. Compile the program if a BASIC compiler is available.

WRITING THE SUBROUTINE

Line Numbers

The subroutine should be written using line numbers and IBM BASICA sytax. Although versions of BASIC are available that do not require line numbers, SPLDRIVE uses a renumbering command that expects each program line to be numbered.

Loops

The subroutine should be written as if the calculation is going to be performed on a single analysis. SPLICE will handle the "Do-Loop" if multiple analyses are batch processed.

Variables

Any variables may be used in writing a subroutine except the following.

AP	EN
BATCH	EP$
BT	JJ
CC	KONTER
CTR	NB
DBPTR	NE
DB$	SC$
DF$	SD$
DT	SN$
DT$	ST$
	SQ

Using these variables will compromise data handling by SPLICE and result in severe errors at run time.

MERGING SPLICE AND THE SUBROUTINE

SPLICE may be merged with the user's subroutine by running SPLDRIVE from BASICA. SPLDRIVE will ask the user for the name of the subroutine and the intended name for the final program. SPLDRIVE then asks the user to invoke a series of commands with the "Enter Key" that will renumber the user's subroutine, merge it to SPLICE and store the result. Note: the final program should be stored in ascii format if it is going to be compiled.

FINAL TOUCHES

Data is assigned to variables in lines 530 to 800 of SPLICE. These default lines cover the most common oxides used in geochemical data. Additional elements, oxides or variables may be added after a subroutine has been merged to SPLICE by creating new program lines between line 800 and line 900. The syntax of the new program lines must correspond to the following example.

```
590 IF EP$(N)="MNO  " THEN MNO=CC(N):GOTO 900
```

In the above example, "MNO " is a five-character, left-justified field that is present in the sample code definition(s) that will be used at run time. The string (e.g. "MNO ") must appear exactly as it does in the sample code definition and there must be exactly 5 characters and spaces between the quotation marks. The expression, MNO=CC(N), sets the value of the variable, MNO. MNO must correspond exactly to the variable used in the user's subroutine. However, it is not required that MNO appears in the preceding character string; for example, XYZ or dummy(2) could have been used in place of MNO. Additional examples of properly written lines may be seen by listing lines 530-800 of SPLICE.

COMPILING

Execution of uncompiled, "SPLICE-based" programs is particularly slow during data recall. It is, therefore, recommended that programs be run in compiled BASIC.